

Package: multipointR (via r-universe)

September 25, 2026

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

Version 0.99.6

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 inhomogeneous Gibbs'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)

URL <https://github.com/mjemons/multipointR>

BugReports <https://github.com/mjemons/multipointR/issues>

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, rlang

biocViews Software, Spatial, Transcriptomics, SingleCell

Suggests knitr, BiocStyle, patchwork, SpatialFeatureExperiment, rmarkdown, SpatialDatasets, sosta, glmnet, 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://bioc.r-universe.dev
Date/Publication 2026-08-26 14:28:29 UTC
RemoteUrl https://github.com/bioc/multipointR
RemoteRef HEAD
RemoteSha 2af89138ec317c161d5d0df885016bf63dd2287b

Contents

.dfToppp	2
.fitSharedModelAcrossImages	3
.fitSingelModelsPerImage	4
.speToDf	5
defineInteractionModel	6
deparseFormula	7
fitModel	8
fitModelAcrossImages	9
get_fun_call	11
get_innermost_call	12
mdlToDf	12
plot.multipointRppm	13
speToPPP	14
Index	15

.dfToppp	<i>Convert SpatialExperiment object to ppp object</i>
----------	---

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

```
.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

<code>spe</code>	SpatialExperiment; object subset to a single image
<code>model</code>	character; the spatstat.model function to use for computation. Typically one of ppm, kppm or dppm.
<code>imageId</code>	character; column in the colData of the SpatialExperiment object specifying the image ID
<code>imageLs</code>	list; a list specifying the subset of all images. If NULL then all images are considered
<code>marks</code>	character; the column with the labels e.g. cell types
<code>formula</code>	formula; the formula to pass to the ppm function
<code>family</code>	detpointprocfamily; Family to use in the point process model. One of dppGauss, dppMatern, dppCauchy, dppBessel or dppPowerExp
<code>threshold</code>	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 inspatstat.model‘

.fitSingelModelsPerImage

fit \$n\$ univariate Models for all \$n\$ images

Description

fit \$n\$ univariate Models for all \$n\$ images

Usage

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

Arguments

<code>spe</code>	SpatialExperiment; object subset to a single image
<code>model</code>	character; the <code>spatstat.model</code> function to use for computation. Typically one of <code>ppm</code> , <code>kppm</code> or <code>dppm</code> .
<code>imageId</code>	character; column in the <code>colData</code> of the SpatialExperiment object specifying the image ID
<code>imageIds</code>	list; a list specifying the subset of all images. If NULL then all images are considered
<code>marks</code>	character; the column with the labels e.g. cell types
<code>formula</code>	formula; the formula to pass to the <code>ppm</code> function
<code>family</code>	<code>detpointprocfamily</code> ; Family to use in the point process model. One of <code>dppGauss</code> , <code>dppMatern</code> , <code>dppCauchy</code> , <code>dppBessel</code> or <code>dppPowerExp</code>
<code>threshold</code>	numeric; a threshold to apply on the minimum number of points a point pattern needs to have to fit a <code>ppm</code> model to it.
<code>interaction</code>	character; Formula specifying whether to fit a Hardcore, Strauss, StraussHard or Fiksel process to the data
<code>cellspacing</code>	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
<code>lambda</code>	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
<code>ncores</code>	numeric; the number of cores to used for parallel processing
<code>verbose</code>	logical; whether to print informations on the fitting
<code>...</code>	other parameters passed on to <code>dppm</code> model from <code>spatstat.model</code>

Value

`mppm` object of the shared fit across all images

<code>.speToDf</code>	<i>Transform a SpatialExperiment into a dataframe</i>
-----------------------	---

Description

Transform a SpatialExperiment into a dataframe

Usage

```
.speToDf(spe)
```

Arguments

<code>spe</code>	SpatialExperiment; object subset to a single image
------------------	--

Value

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

```
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 $\sqrt{n(n+1)}$ as done in <code>spatstat.model::Hardcore</code> . 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 ~ spatstat.geom::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	<i>deparse the Formula for spatstat and extract the relevant data</i>
----------------	---

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 ~ spatstat.geom::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,
  improve.type = NULL,
  relaxed = FALSE,
  ...
)
```

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 $\sqrt{n(n+1)}$ as done in <code>spatstat.model::Hardcore</code> . In the case of StraussHard, only the Strauss interaction radius can be user-provided, the Hardcore interaction is always estimated from the data.
improve.type	character the improve.type argument from <code>spatstat.model::ppm.ppp</code> , passed on directly
relaxed	logical whether or not to perform a relaxed fit with only the non zero coefficients from glmnet improvement
...	other parameters passed on to ppm model from <code>spatstat.model</code>

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 ~ spatstat.geom::distfun(CD8_T_cell)"
  )
)
```

fitModelAcrossImages	<i>Fit point process models across all images in the SpatialExperiment object</i>
----------------------	---

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 = FALSE,
  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",
  imageIs = list("1", "2"),
  marks = "cellType",
  formula = as.formula(
    "Keratin_Tumour ~ spatstat.geom::distfun(CD8_T_cell)"
  ),
  threshold = 10
)
```

get_fun_call	<i>Title</i>
--------------	--------------

Description

Title

Usage

```
get_fun_call(expr, var)
```

Arguments

expr part of a formula object
var variable to extract

Value

the extracted formula call of the bespoke variable

get_innermost_call	<i>Function to obtain the innermost call of a formula expression</i>
--------------------	--

Description

Function to obtain the innermost call of a formula expression

Usage

```
get_innermost_call(term)
```

Arguments

term	the term of the function to de-nest
------	-------------------------------------

Value

the innermost call of a nested function

mdlToDf	<i>Convert the list of spatial models to a dataframe</i>
---------	--

Description

Convert the list of spatial models to a dataframe

Usage

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

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
fine	logical Logical flag passed to summary.ppm controlling the coarse or finer estimate of the covariance matrix

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 ~ spatstat.geom::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 ~ spatstat.geom::distfun(CD8_T_cell)"
  )
)
plot(mdl)
```

speToPPP	<i>convert a SpatialExperiment object into a point pattern</i>
----------	--

Description

convert a SpatialExperiment object into a point pattern

Usage

```
speToPPP(spe, marks, continuous = FALSE, window = NULL)
```

Arguments

spe	SpatialExperiment; object subset to a single image
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

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

`.dfToppp`, [2](#)
`.fitSharedModelAcrossImages`, [3](#)
`.fitSingelModelsPerImage`, [4](#)
`.speToDf`, [5](#)

`defineInteractionModel`, [6](#)
`deparseFormula`, [7](#)

`fitModel`, [8](#)
`fitModelAcrossImages`, [9](#)

`get_fun_call`, [11](#)
`get_innermost_call`, [12](#)

`mdlToDf`, [12](#)

`plot.multipointRppm`, [13](#)

`speToPPP`, [14](#)