Package: multipointR (via r-universe)

November 25, 2026

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

Version 1.0.0

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)

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), terra

Config/testthat/edition 3

Depends R (>= 4.1.0)

VignetteBuilder knitr

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

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:56:35 UTC

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

RemoteRef RELEASE_3_24

RemoteSha d9c9eed38f987535b303474c590a984e4b6053f5

Contents

.dfToppp 2
.fitSharedModelAcrossImages 3
.fitSingelModelsPerImage 4
.speToDf 6
defineInteractionModel 6
deparseFormula 7
fitModel 8
fitModelAcrossImages 10
get_fun_call 12
get_innermost_call 13
mdlToDf 13
plot.multipointRppm 14
speToPPP 15
summary.multipointRppm 16

Index **17**

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

Description

Convert SpatialExperiment object to ppp object

Usage

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

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. The default is to estimate the window as a convex hull around the available points as estimated by the function spatstat.geom::ripras. If users provide instead window = NULL a rectanglular window surrounding min and max is estimated instead. A custom window can be supplied too.

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

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
window	owin; An observation window of the point pattern of class. The default is to estimate the window as a convex hull around the available points as estimated by the function spatstat.geom::ripras. If users provide instead window = NULL a rectanglular window surrounding min and max is estimated instead. A custom window can be supplied too.
...	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 = "kppm",
  imageId,
  imageLs = NULL,
  marks,
  formula,
```

```
  family = spatstat.model::dppGauss(),
  threshold = NULL,
  interaction = "StraussHard",
  cellspacing = NA,
  lambda = NULL,
  ncores = 1,
  verbose = TRUE,
  window,
  ...
)
```

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
window	owin; An observation window of the point pattern of class. The default is to estimate the window as a convex hull around the available points as estimated by the function spatstat.geom::ripras. If users provide instead window = NULL a rectanglular window surrounding min and max is estimated instead. A custom window can be supplied too.
...	other parameters passed on to dppm model from spatstat.model

Value

mppm object of the shared fit across all images

.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

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 ~ 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,
    window = NULL
)

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, window)
```

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.
window	owin; An observation window of the point pattern of class.

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,
    window = NULL
)
```

---

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 = "kppm",
  marks,
  formula,
  clusters = "LGCP",
```

```
  threshold = NULL,
  interaction = "Hardcore",
  lambda = NULL,
  cellspacing = NA,
  improve.type = NULL,
  method.kppm = "palm",
  relaxed = FALSE,
  penalised = TRUE,
  standardize = FALSE,
  selectionExclude = NULL,
  ctrlKppm = list(),
  covmodel = list(model = "exponential"),
  window = "ripras",
  ...
)
```

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
clusters	character; Cox Process or clustering variant to use with kppm. One of either "Thomas", "MatClust", "Cauchy", "VarGamma", "LGCP"
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.
improve.type	character the improve.type argument to ppm or kppm`` from spatstat.model::ppm.ppp, passed on for more details
method.kppm	character the method to use for kppm
relaxed	logical whether or not to perform a relaxed fit with only the non zero coefficients from glmnet improvement
penalised	logical whether or not to penalise against extreme values in the clustering in the objective function

standardize — logical whether or not to standardise all distances to to a 0,1 interval to improve spline and polynomial fits

selectionExclude — character; Formula term specified to be removed from the selection of the Lasso. This is necessary with improve.type = "enet", because else the p-values will be not valid due to obvious double dipping. The inference then becomes an LRT with and without the variable indicated in selectionExclude.

ctrlKppm — list; a named list to control the kppm fit

covmodel — list; a named list with the covariance options for the LGCP model

window — owin; An observation window of the point pattern of class. The default is to estimate the window as a convex hull around the available points as estimated by the function spatstat.geom::ripras. If users provide instead window = NULL a rectanglular window surrounding min and max is estimated instead. A custom window can be supplied too.

... — 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 ~ spatstat.geom::distfun(CD8_T_cell)"
    ),
    method.kppm = "mincon" # to reduce compute time
)
```

---

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 = "kppm",
  imageId,
  imageLs = NULL,
  marks,
  formula,
  family = spatstat.model::dppGauss(),
  threshold = NULL,
  interaction = "Hardcore",
  cellspacing = NA,
  lambda = NULL,
  sharedModel = FALSE,
  ncores = 1,
  verbose = TRUE,
  window = "ripras",
  ...
)
```

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
window	owin; An observation window of the point pattern of class. The default is to estimate the window as a convex hull around the available points as estimated by the function spatstat.geom::ripras. If users provide instead window = NULL a rectanglular window surrounding min and max is estimated instead. A custom window can be supplied too.
...	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"),
    marks = "cellType",
    formula = as.formula(
        "Keratin_Tumour ~ spatstat.geom::distfun(CD8_T_cell)"
    ),
    threshold = 10,
    method.kppm = "mincon" # to reduce compute time
)
```

---

get_fun_call *Title*

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 *Function to obtain the innermost call of a formula expression*

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 funciton

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"), fine = FALSE, fast = TRUE)
```

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
fast	logical Logical flag passed to summary.multipointRppm controlling whether tapering (using sparse matrices from Matrix) should be used to speed up calculations. Warning: This is expected to underestimate the true asymptotic variances/covariances.

Value

DataFrame of the model coefficients

Examples

```
## Not run:
spe <- SpatialDatasets::spe_Keren_2018()

mdlLs <- fitModelAcrossImages(
    spe = spe,
    imageId = "imageID",
    imageLs = list("1"),
    marks = "cellType",
    sharedModel = FALSE,
    formula = as.formula(
        "Keratin_Tumour ~ spatstat.geom::distfun(CD8_T_cell)"
    ),
    threshold = 10,
    method.kppm = "mincon" # to reduce compute time
)

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

## End(Not run)
```

---

plot.multipointRppm *Plot multipointR* ppm *objects*

Description

Plot multipointR 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` *convert a SpatialExperiment object into a point pattern*

Description

convert a SpatialExperiment object into a point pattern

Usage

```
speToPPP(spe, marks, continuous = FALSE, window = "ripras")
```

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. The default is to estimate the window as a convex hull around the available points as estimated by the function `spatstat.geom::ripras`. If users provide instead `window = NULL` a rectanglular window surrounding min and max is estimated instead. A custom window can be supplied too.

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")
```

summary.multipointRppm *generic summary of multipointRppm objects*

Description

generic summary of multipointRppm objects

Usage

```
## S3 method for class 'multipointRppm'
summary(object, quick = FALSE, fast = TRUE, eps.rmax = 0.01, ...)
```

Arguments

object	ppm; a model fit with spatstat.model::ppm or the wrapper multipointR::fitModel
quick	logical whether or not to calculate standard errors
fast	logical; whether or not to use the default summary.ppm/ summary.kppm functions or make use of the sparse matrix calculation of vcov.kppm. Has no effect for non kppm objects and defaults to the normal summary
eps.rmax	numerical from vcov.kppm, check directly
...	other parameters passed on to summary

Value

A list with summary information, see summary.kppm() or summary.ppm()

Author(s)

Martin Emons, adapted from summary.kppm() written by Adrian Baddeley, Rolf Turner and Ege Rubak

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)"
    ),
    method.kppm = "mincon" # to reduce compute time
)

summary(mdl)
```

Index

.dfToppp, 2
.fitSharedModelAcrossImages, 3
.fitSingelModelsPerImage, 4
.speToDf, 6

defineInteractionModel, 6
deparseFormula, 7

fitModel, 8
fitModelAcrossImages, 10

get_fun_call, 12
get_innermost_call, 13

mdlToDf, 13

plot.multipointRppm, 14

speToPPP, 15
summary.kppm, *16*
summary.multipointRppm, 16
summary.ppm, *16*