

Seedlings dataset

```
library(ppjsdm)
#> Registered S3 method overwritten by 'spatstat':
#>   method      from
#> print.boxx cli
library(ecespa)
#> Loading required package: spatstat
#> Loading required package: spatstat.data
#> Loading required package: nlme
#> Loading required package: rpart
#>
#> spatstat 1.64-0      (nickname: 'Susana Distancia')
#> For an introduction to spatstat, type 'beginner'
remove(list = ls())

set.seed(1)
```

This vignette explains how to use the `ppjsdm` package with the `seedlings1` dataset from `ecespa`. This dataset is a marked point pattern of seedlings of *H. squamatum* growing in a gyp-sophylous plant community in Central Spain. The datasets contains the locations of the seedlings marked with their height. Both the coordinates and the height of the seedlings are given in cm.

Taking marks into account

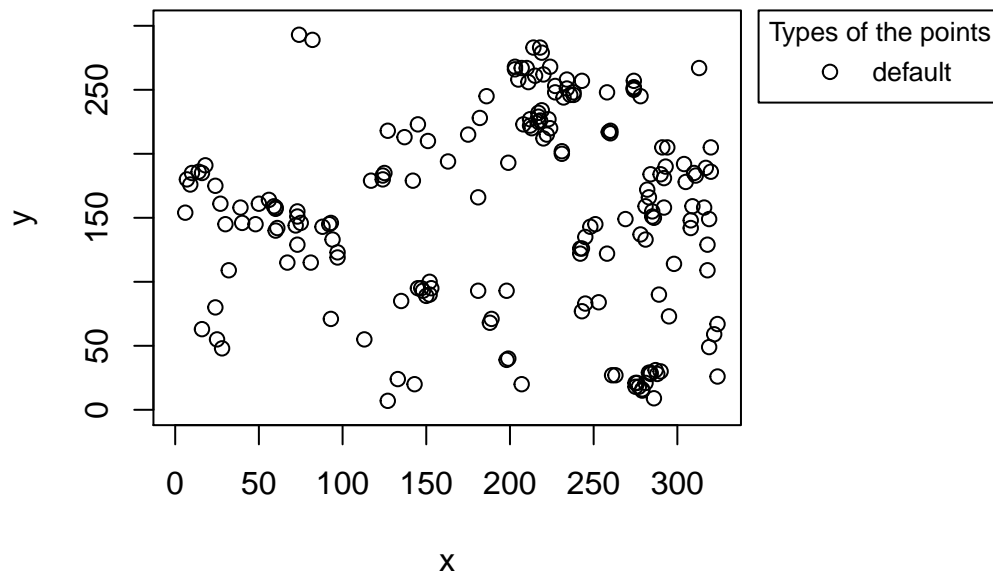
If marks (i.e. their height) are provided, the interaction radii are proportional to the marks. We begin with that setting.

```
configuration <- Configuration(seedlings1$x, seedlings1$y, marks = seedlings1$marks)
window <- Rectangle_window(c(0, 325), c(0, 300))
```

The point configuration is plotted below.

```
print(configuration)
#> An S3 object representing a configuration.
#>
#> Number of points: 184.
par(mar = c(5, 4, 4, 13) + 0.1)
plot(configuration, window = window)
```

Points in the configuration



We provide a series of ranges for the interaction radii, and let the fitting function calibrate the model.

```
short_range <- c(0, 20)
medium_range <- c(0, 20)
long_range <- c(0, 20)
model <- "square_exponential"
medium_range_model <- "square_exponential"
saturation <- 2
steps <- 100000
```

We can now call the fitting function.

```
fit <- ppjsdm::gibbsm(configuration,
  window = window,
  model = model,
  medium_range_model = medium_range_model,
  short_range = short_range,
  medium_range = medium_range,
  long_range = long_range,
  use_glmnet = FALSE,
  use_aic = TRUE,
  saturation = saturation)

#> $beta0
#> [1] -6.555334
#>
#> $alpha
#>      [,1]
#> [1,] 0.8144046
#>
#> $gamma
```

```

#>           [,1]
#> [1,] -0.3224347
#>
#> $beta
#>
#> [1,]
print(summary(fit))
#>      coefficients      se    CI95_lo    CI95_hi Ztest      Pval
#> log_lambda1 -6.5553338 0.19257605 -6.9327760 -6.177892 *** 5.661279e-254
#> alpha_1_1    0.8144046 0.10562800  0.6073775  1.021432 *** 1.256998e-14
#> gamma_1_1    -0.3224347 0.07376089 -0.4670033 -0.177866 *** 1.234806e-05
#>      Zval
#> log_lambda1 -34.04023
#> alpha_1_1    7.71012
#> gamma_1_1    -4.37135
print(fit$coefficients)
#> $beta0
#> [1] -6.555334
#>
#> $alpha
#>           [,1]
#> [1,] 0.8144046
#>
#> $gamma
#>           [,1]
#> [1,] -0.3224347
#>
#> $beta
#>
#> [1,]
#>
#> $short_range
#>           [,1]
#> [1,] 5.114931
#>
#> $medium_range
#>           [,1]
#> [1,] 13.95521
#>
#> $long_range
#>           [,1]
#> [1,] 19.87263
print(fit$aic)
#> [1] 817.9467
print(fit$bic)
#> [1] 832.4198

```

We may then plot the corresponding Papangelou conditional intensity.

```

parameters <- fit$coefficients
# plot_papangelou(window = window,
#                 configuration = configuration,
#                 type = 1,
#                 mark = mean(get_marks(configuration)),

```

```

#           model = model,
#           medium_range_model = medium_range_model,
#           alpha = parameters$alpha,
#           beta0 = parameters$beta0,
#           beta = matrix(0, 1, 0),
#           gamma = parameters$gamma,
#           covariates = list(),
#           short_range = parameters$short_range,
#           medium_range = parameters$medium_range,
#           long_range = parameters$long_range,
#           saturation = saturation)

```

It is also possible to draw from the model.

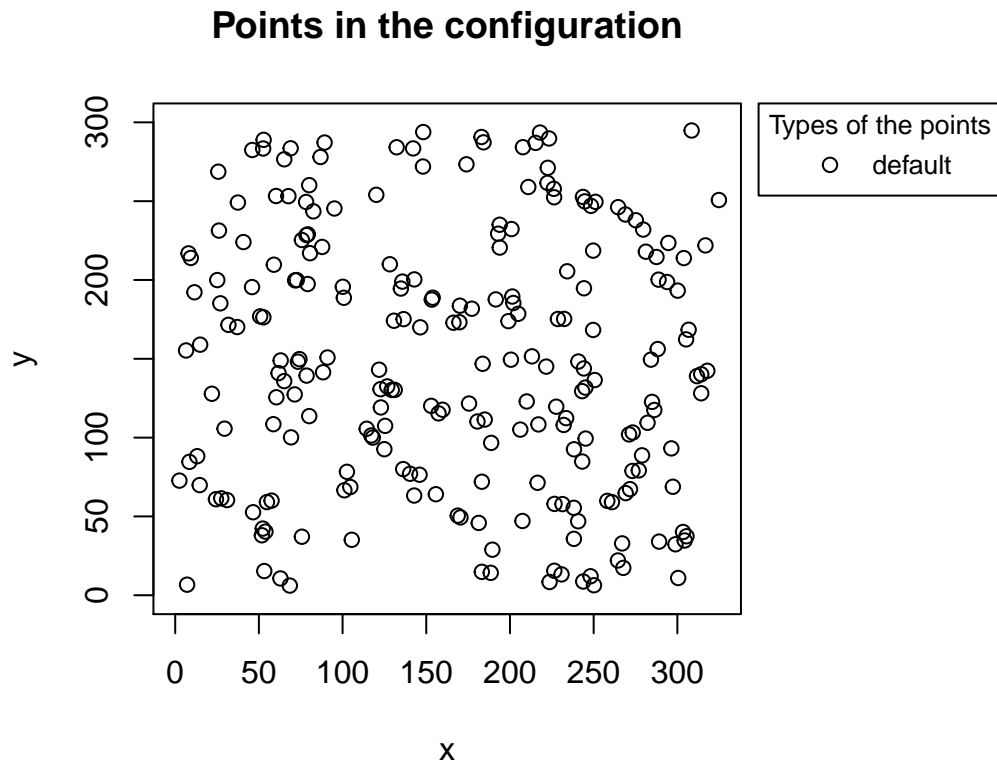
```

draw <- ppjsdm::rgibbs(window = window,
  alpha = parameters$alpha,
  beta0 = parameters$beta0,
  gamma = parameters$gamma,
  model = model,
  medium_range_model = medium_range_model,
  short_range = parameters$short_range,
  medium_range = parameters$medium_range,
  long_range = parameters$long_range,
  types = levels(types(configuration)),
  mark_range = c(min(get_marks(configuration)), max(get_marks(configuration))),
  saturation = saturation,
  steps = steps)

print(draw)
#> An S3 object representing a configuration.
#>
#> Number of points: 232.

par(mar = c(5, 4, 4, 13) + 0.1)
plot(draw, window = window)

```



Interaction radii not proportional to marks

In this section, we disregard the height of the seedlings, and consider interaction radii in centimetres.

```
configuration <- Configuration(seedlings1$x, seedlings1$y)
```

We call the fitting function on this unmarked point process.

```
fit <- ppjsdm::gibbsm(configuration,
  window = window,
  model = model,
  medium_range_model = medium_range_model,
  short_range = short_range,
  medium_range = medium_range,
  long_range = long_range,
  use_glmnet = FALSE,
  use_aic = TRUE,
  saturation = saturation)

#> $beta0
#> [1] -7.932073
#>
#> $alpha
#>      [,1]
#> [1,] 1.174084
#>
```

```

#> $gamma
#>           [,1]
#> [1,] 0.2866842
#>
#> $beta
#>
#> [1,]
print(summary(fit))
#>      coefficients      se  CI95_lo  CI95_hi Ztest      Pval
#> log_lambda1 -7.9320730 0.18176210 -8.2883202 -7.5758258 *** 0.000000e+00
#> alpha_1_1    1.1740839 0.11699442  0.9447791  1.4033888 *** 1.065443e-23
#> gamma_1_1    0.2866842 0.09032952  0.1096416  0.4637268 ** 1.504781e-03
#>      Zval
#> log_lambda1 -43.63986
#> alpha_1_1    10.03538
#> gamma_1_1    3.17376
print(fit$coefficients)
#> $beta0
#> [1] -7.932073
#>
#> $alpha
#>           [,1]
#> [1,] 1.174084
#>
#> $gamma
#>           [,1]
#> [1,] 0.2866842
#>
#> $beta
#>
#> [1,]
#>
#> $short_range
#>           [,1]
#> [1,] 8.061338
#>
#> $medium_range
#>           [,1]
#> [1,] 13.89822
#>
#> $long_range
#>           [,1]
#> [1,] 20.10473
print(fit$aic)
#> [1] 738.6893
print(fit$bic)
#> [1] 753.1624

```

We may then plot the corresponding Papangelou conditional intensity.

```

parameters <- fit$coefficients
# plot_papangelou(window = window,
#                 configuration = configuration,
#                 type = 1,

```

```

#           mark = mean(get_marks(configuration)),
#           model = model,
#           medium_range_model = medium_range_model,
#           alpha = parameters$alpha,
#           beta0 = parameters$beta0,
#           beta = matrix(0, 1, 0),
#           gamma = parameters$gamma,
#           covariates = list(),
#           short_range = parameters$short_range,
#           medium_range = parameters$medium_range,
#           long_range = parameters$long_range,
#           saturation = saturation)

```

And as previously, we draw from the model.

```

draw <- ppjsdm::rgibbs(window = window,
  alpha = parameters$alpha,
  beta0 = parameters$beta0,
  gamma = parameters$gamma,
  model = model,
  medium_range_model = medium_range_model,
  short_range = parameters$short_range,
  medium_range = parameters$medium_range,
  long_range = parameters$long_range,
  types = levels(types(configuration)),
  mark_range = c(min(get_marks(configuration)), max(get_marks(configuration))),
  saturation = saturation,
  steps = steps)

print(draw)
#> An S3 object representing a configuration.
#>
#> Number of points: 272.

par(mar = c(5, 4, 4, 13) + 0.1)
plot(draw, window = window)

```

Points in the configuration

