Kimboto dataset

```r
library(ppjsdm)
#> Registered S3 method overwritten by 'spatstat':
#>   method     from
#>   print.boxx cli
library(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'
library(plot.matrix)
remove(list = ls())

set.seed(1)
```

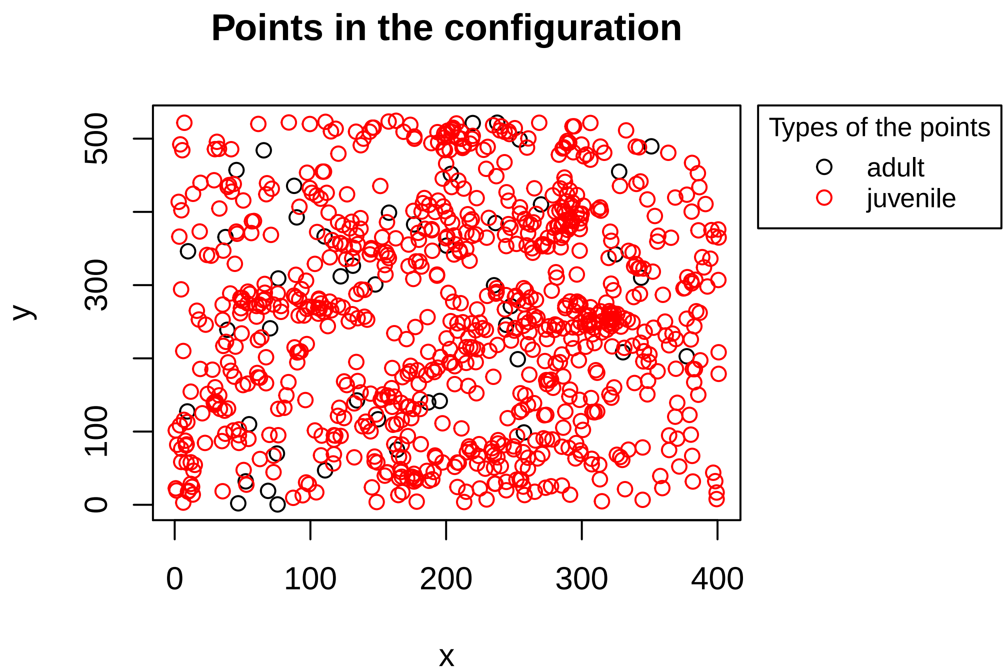
This vignette explains how to use the `ppjsdm` package with the kimboto dataset from `spatstat`. This dataset is a point pattern of adult and juvenile Kimboto trees (Pradosia cochlearia or P. ptychandra) recorded at Paracou in French Guiana. See Flores (2005).

The dataset paracou is a point pattern (object of class "ppp") containing the spatial coordinates of each tree, marked by age (a factor with levels adult and juvenile. The survey region is a rectangle approximately 400 by 525 metres. Coordinates are given in metres. We begin by loading the data with all species.

```r
print(paracou)
#> Marked planar point pattern: 884 points
#> Multitype, with levels = adult, juvenile
#> window: rectangle = [0, 400.8568] x [0, 524.4037] metres
configuration <- Configuration(paracou)
#> Warning in Configuration(paracou): There are duplicate points in the
#> configuration.
window <- Rectangle_window(c(0, 400.8568), c(0, 524.4037))
```

The point configuration is plotted below.

```r
par(mar = c(5, 4, 4, 13) + 0.1)
plot(configuration, window = window)
```



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

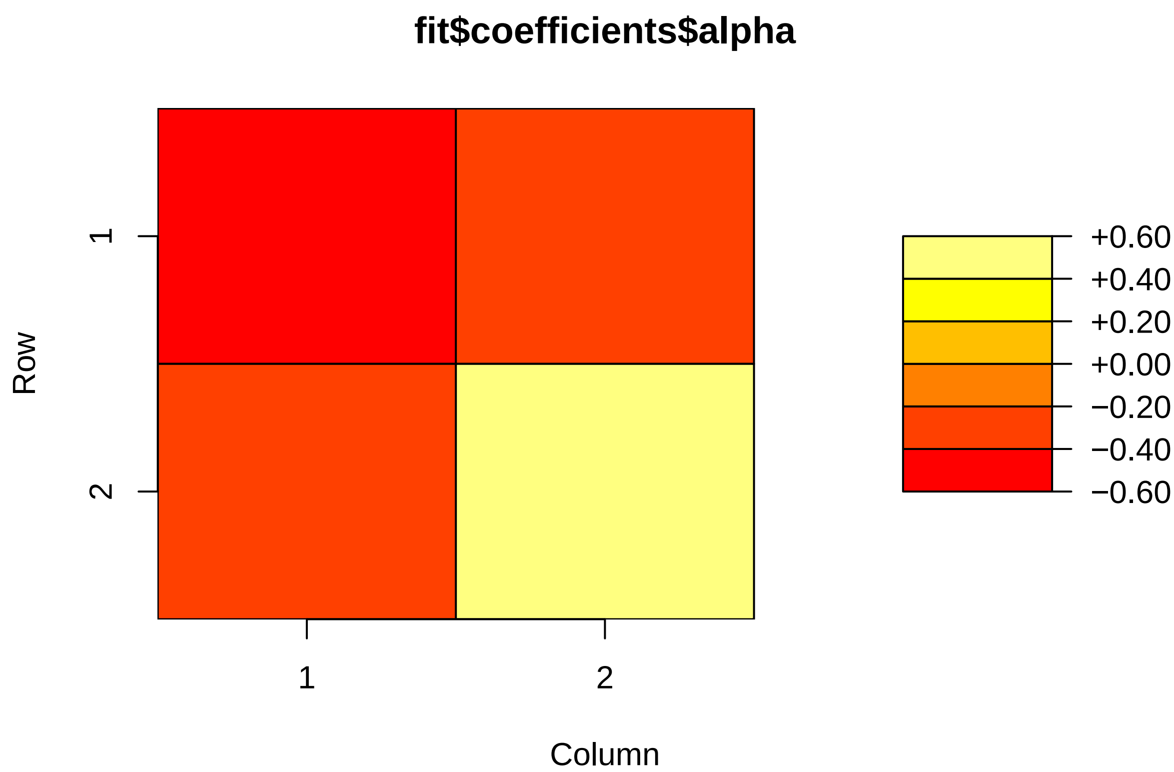
```
short_range <- c(0, 20)
medium_range <- c(0, 20)
long_range <- c(0, 20)
model <- "square_exponential"
medium_range_model <- "square_exponential"
steps <- 100000
max_points <- 1000
saturation <- 2
```

We can now call the fitting function.

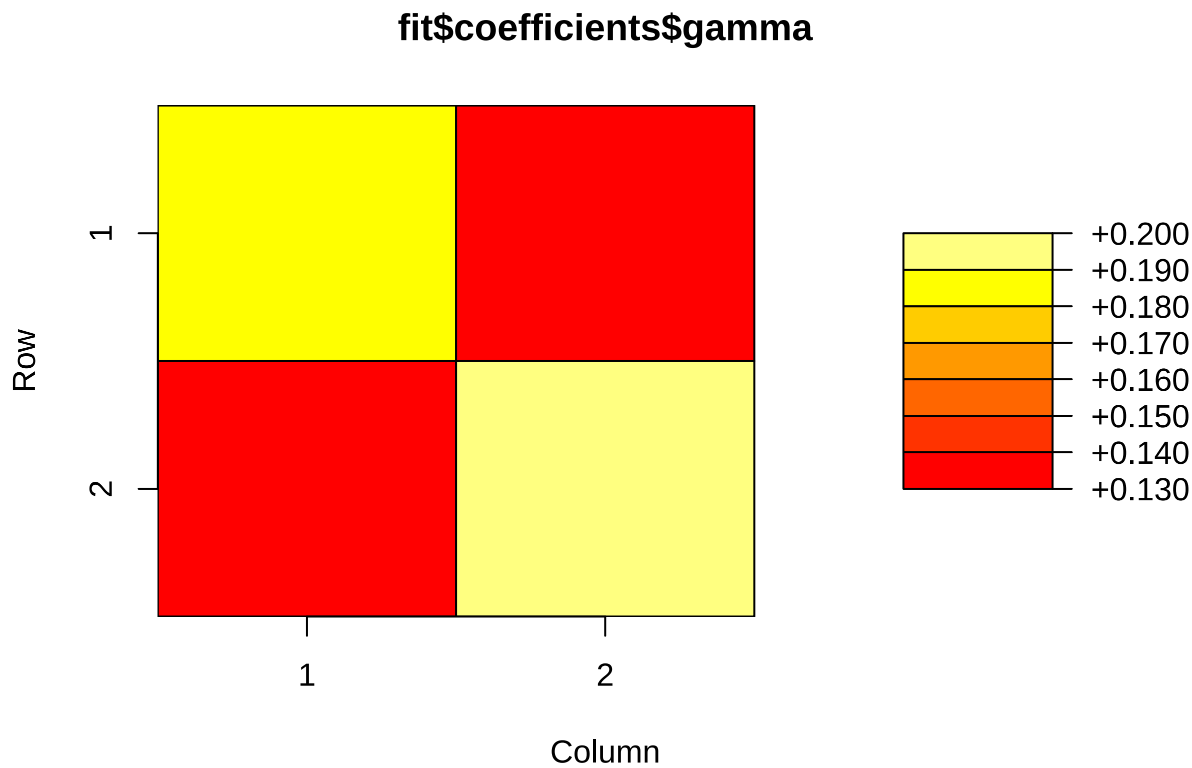
```
fit <- suppressWarnings(ppjsdm::gibbsm(configuration,
                        window = window,
                        model = model,
                        medium_range_model = medium_range_model,
                        short_range = short_range,
                        medium_range = medium_range,
                        long_range = long_range,
                        use_glmnet = FALSE))
#> $beta0
#> [1] -8.468144 -6.508426
#>
#> $alpha
#>            [,1]       [,2]
#> [1,] -0.5577473 -0.2028786
#> [2,] -0.2028786  0.5688890
#>
#> $gamma
```

```r
#>           [,1]      [,2]
#> [1,] 0.1842345 0.1366565
#> [2,] 0.1366565 0.1971540
#>
#> $beta
#>
#> [1,]
#> [2,]
print(summary(fit))
#>              coefficients         se     CI95_lo     CI95_hi Ztest          Pval
#> log_lambda1    -8.4681436 0.36336995 -9.18033563 -7.75595162   *** 3.994046e-120
#> log_lambda2    -6.5084257 0.05554590 -6.61729370 -6.39955778   ***  0.000000e+00
#> alpha_1_1      -0.5577473 0.52905614 -1.59467827  0.47918371        2.917773e-01
#> alpha_1_2      -0.2028786 0.06672953 -0.33366606 -0.07209109    **  2.363335e-03
#> alpha_2_2       0.5688890 0.04219056  0.48619699  0.65158093   ***  1.948132e-41
#> gamma_1_1       0.1842345 0.21513007 -0.23741272  0.60588164        3.917841e-01
#> gamma_1_2       0.1366565 0.04599909  0.04649996  0.22681309    **  2.969744e-03
#> gamma_2_2       0.1971540 0.03402145  0.13047319  0.26383482   ***  6.832470e-09
#>                      Zval
#> log_lambda1  -23.3044690
#> log_lambda2 -117.1720337
#> alpha_1_1     -1.0542308
#> alpha_1_2     -3.0403116
#> alpha_2_2     13.4837985
#> gamma_1_1      0.8563864
#> gamma_1_2      2.9708525
#> gamma_2_2      5.7949913
print(fit$coefficients)
#> $beta0
#> [1] -8.468144 -6.508426
#>
#> $alpha
#>            [,1]       [,2]
#> [1,] -0.5577473 -0.2028786
#> [2,] -0.2028786  0.5688890
#>
#> $gamma
#>           [,1]      [,2]
#> [1,] 0.1842345 0.1366565
#> [2,] 0.1366565 0.1971540
#>
#> $beta
#>
#> [1,]
#> [2,]
#>
#> $short_range
#>         [,1]      [,2]
#> [1,] 14.7101 13.453897
#> [2,] 13.4539  6.220489
#>
#> $medium_range
#>          [,1]     [,2]
```

```
#> [1,] 24.74726 19.29326
#> [2,] 19.29326 12.28019
#>
#> $long_range
#>           [,1]     [,2]
#> [1,] 37.76617 26.60547
#> [2,] 26.60547 18.68409
par(mar=c(5.1, 5.1, 4.1, 4.1))
plot(fit$coefficients$alpha)
```




```
plot(fit$coefficients$gamma)
```




```r
print(fit$aic)
#> [1] 4327.855
print(fit$bic)
#> [1] 4379.559
```

We may then plot the corresponding Papangelou conditional intensity.

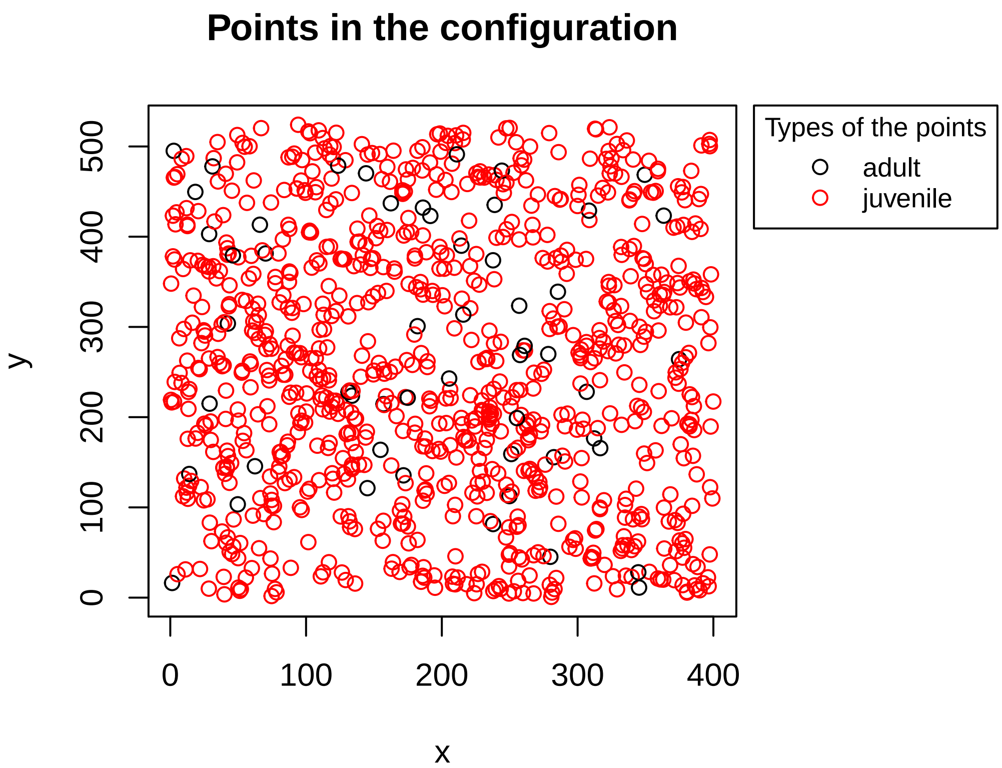
```r
parameters <- fit$coefficients
# plot_papangelou(window = window,
#                 configuration = configuration,
#                 type = 1,
#                 model = model,
#                 medium_range_model = medium_range_model,
#                 alpha = parameters$alpha,
#                 lambda = parameters$lambda,
#                 beta = matrix(0, 2, 0),
#                 gamma = parameters$gamma,
#                 covariates = list(),
#                 short_range = parameters$short_range,
#                 medium_range = parameters$medium_range,
#                 long_range = parameters$long_range,
#                 saturation = saturation,
#                 max_points = max_points)
#
# plot_papangelou(window = window,
#                 configuration = configuration,
#                 type = 2,
```

```r
#                   model = model,
#                   medium_range_model = medium_range_model,
#                   alpha = parameters$alpha,
#                   lambda = parameters$lambda,
#                   beta = matrix(0, 2, 0),
#                   gamma = parameters$gamma,
#                   covariates = list(),
#                   short_range = parameters$short_range,
#                   medium_range = parameters$medium_range,
#                   long_range = parameters$long_range,
#                   saturation = saturation,
#                   max_points = max_points)
```

It is also possible to draw from the model.

```r
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: 972.

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



Points in the configuration
Types of the points
adult
juvenile
y
x
0
100
200
300
400
500
0
100
200
300
400