Queensland rainforest

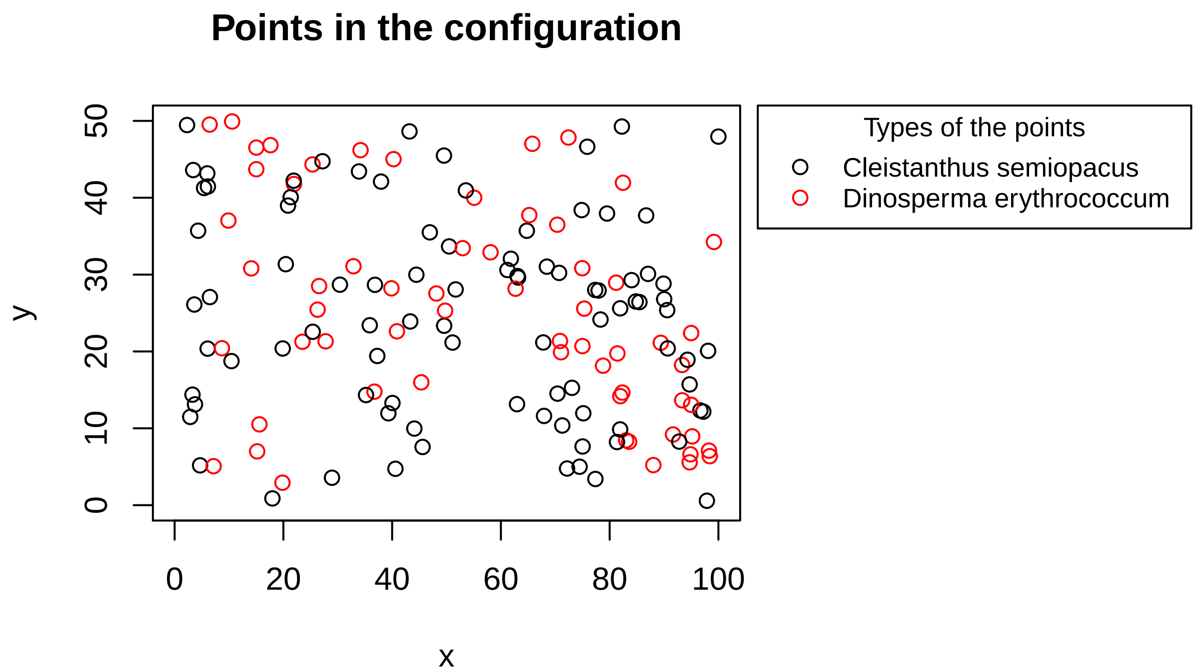
```r
library(ppjsdm)
#> Registered S3 method overwritten by 'spatstat':
#>   method     from
#>   print.boxx cli
library(spatstat)
#> Loading required package: spatstat.data
#> Loading required package: nlme
#> Loading required package: rpart
#>
#> spatstat 1.64-0       (nickname: 'Susana Distancia')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

source("../R/get_qld.R")

set.seed(1)
```

This vignette explains how to use the `ppjsdm` package with the Queensland rainforest dataset from CSIRO. We begin by loading the data with the most prevalent species.

```r
index_of_plot <- 3 # Between 1 and 20
year <- 2011 # Year of census
number_of_species <- 2 # Includes the most prevalent species from the plot

qld <- get_qld(index = index_of_plot,
               year = year,
               prevalent = number_of_species)
#> The chosen index corresponds to ep2.
configuration <- qld$configuration
window <- qld$window
```

The point configuration is plotted below.

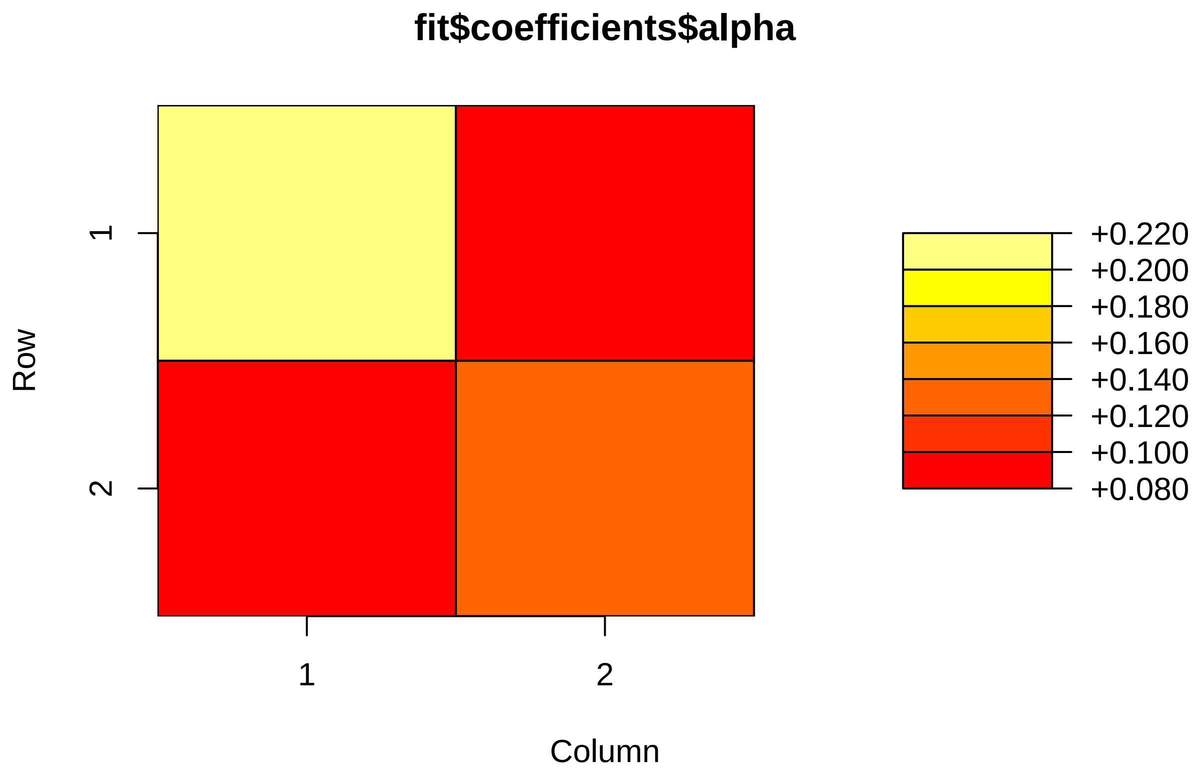
```r
par(mar = c(5, 4, 4, 13) + 0.1)
plot(configuration, window = window)
```



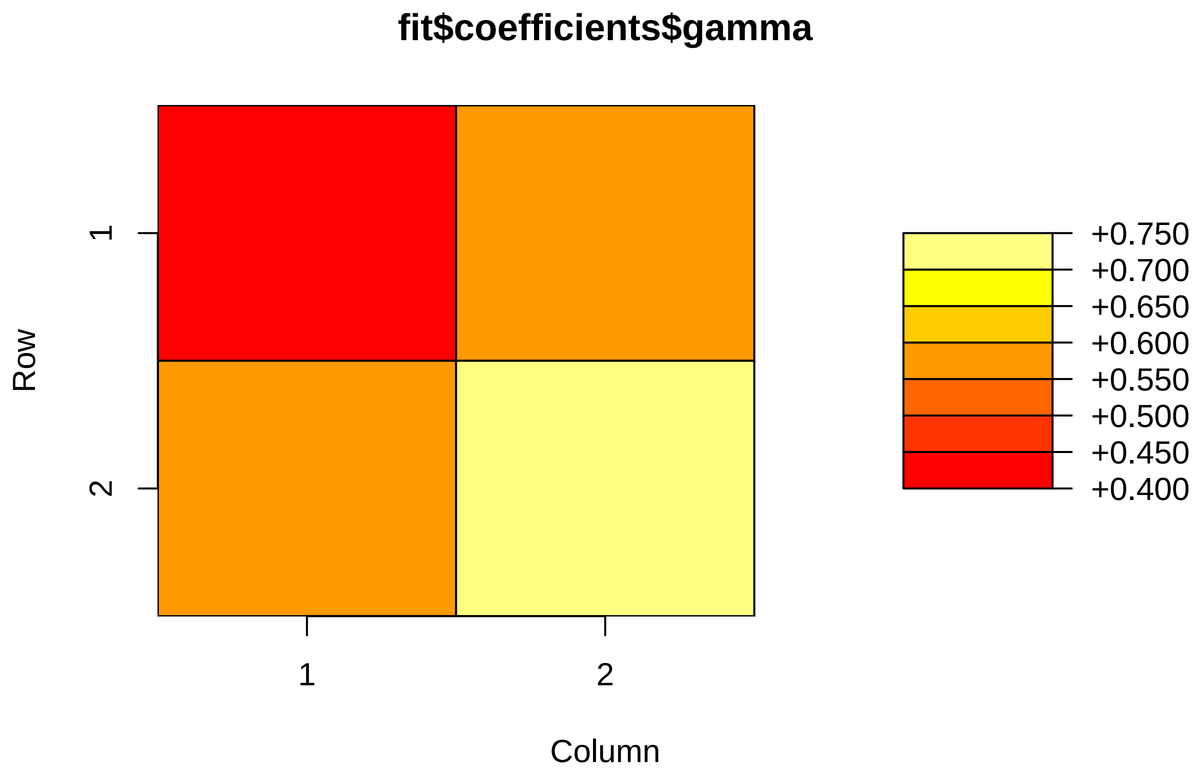
The function `gibbsm` fits a multivariate Gibbs point process to our dataset. For example,

```
short_range <- c(0, 10)
medium_range <- c(0, 10)
long_range <- c(0, 10)

fit <- ppjsdm::gibbsm(configuration,
                      short_range = short_range,
                      medium_range = medium_range,
                      long_range = long_range,
                      window = window,
                      use_glmnet = FALSE)
#> $beta0
#> [1] -5.495084 -7.458457
#>
#> $alpha
#>            [,1]       [,2]
#> [1,] 0.21393339 0.08807536
#> [2,] 0.08807536 0.12277606
#>
#> $gamma
#>           [,1]      [,2]
#> [1,] 0.4416140 0.5531471
#> [2,] 0.5531471 0.7312516
#>
#> $beta
#>
#> [1,]
#> [2,]
print(fit$coefficients)
```

```
#> $beta0
#> [1] -5.495084 -7.458457
#>
#> $alpha
#>            [,1]       [,2]
#> [1,] 0.21393339 0.08807536
#> [2,] 0.08807536 0.12277606
#>
#> $gamma
#>           [,1]      [,2]
#> [1,] 0.4416140 0.5531471
#> [2,] 0.5531471 0.7312516
#>
#> $beta
#>
#> [1,]
#> [2,]
#>
#> $short_range
#>          [,1]     [,2]
#> [1,] 1.305345 3.384901
#> [2,] 3.384901 3.915574
#>
#> $medium_range
#>           [,1]      [,2]
#> [1,]  7.460803 10.028878
#> [2,] 10.028878  5.100327
#>
#> $long_range
#>           [,1]     [,2]
#> [1,]  7.775195 13.83506
#> [2,] 13.835055 10.60002
par(mar = c(5.1, 5.1, 4.1, 4.1))
plot(fit$coefficients$alpha)
```




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




```
print(fit$aic)
#> [1] 850.9574
print(fit$bic)
#> [1] 891.3445
```

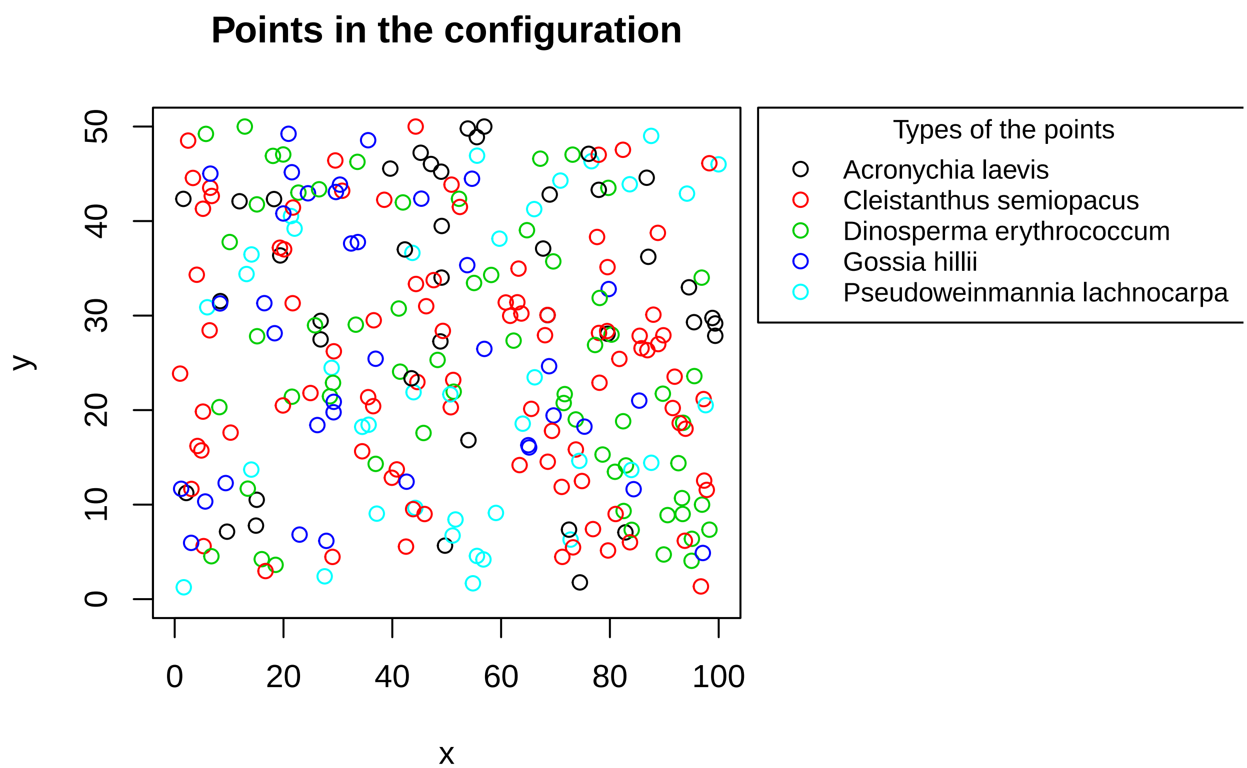
It is then possible to draw from the model, as can be seen below.

```
# parameters <- fit$coefficients
# draw <- ppjsdm::rgibbs(window = window,
#                         alpha = parameters$alpha,
#                         lambda = parameters$lambda,
#                         gamma = parameters$gamma,
#                         short_range = parameters$short_range,
#                         medium_range = parameters$medium_range,
#                         long_range = parameters$long_range,
#                         types = levels(types(configuration)))
# print(draw)
#
# par(mar = c(5, 4, 4, 13) + 0.1)
# plot(draw, window = window)
```

Let us increase the number of species accounted for.

```
saturation <- 2
max_points <- 1000

index_of_plot <- 3 # Between 1 and 20
year <- 2011 # Year of census
```

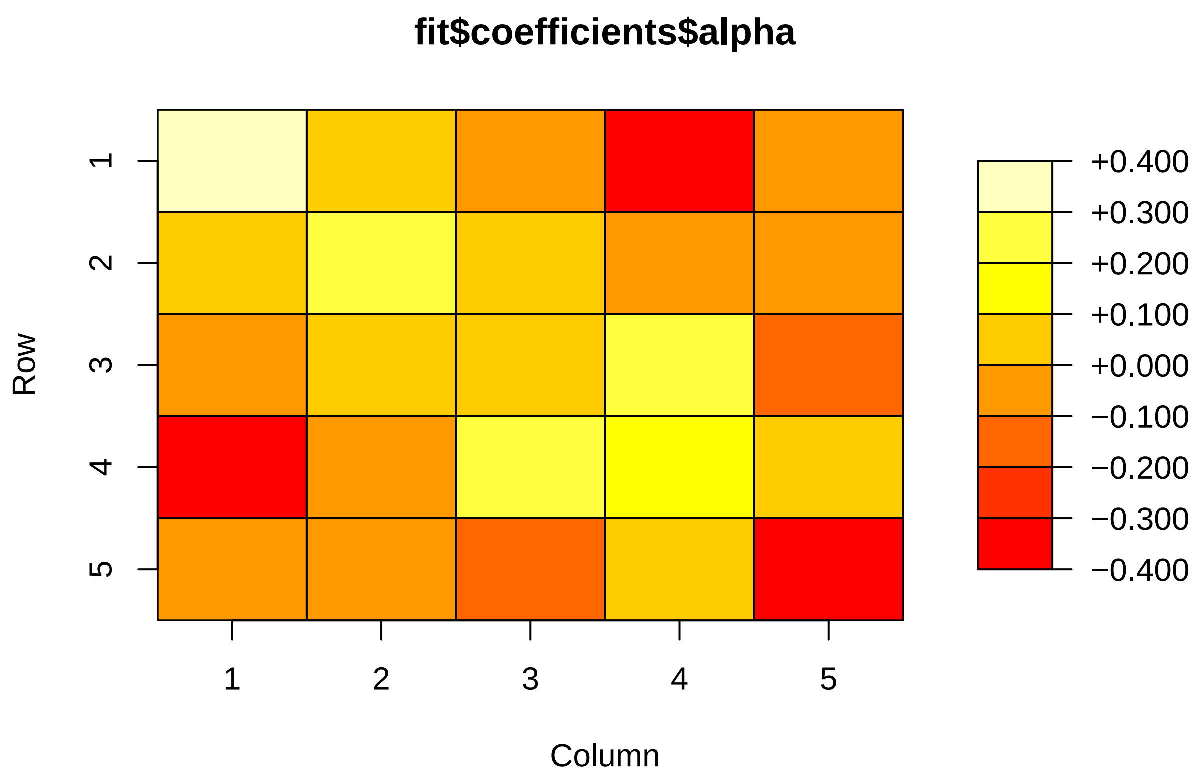
```r
number_of_species <- 5 # Includes the most prevalent species from the plot

qld <- get_qld(index = index_of_plot,
               year = year,
               prevalent = number_of_species)
#> The chosen index corresponds to ep2.
configuration <- qld$configuration
window <- qld$window

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



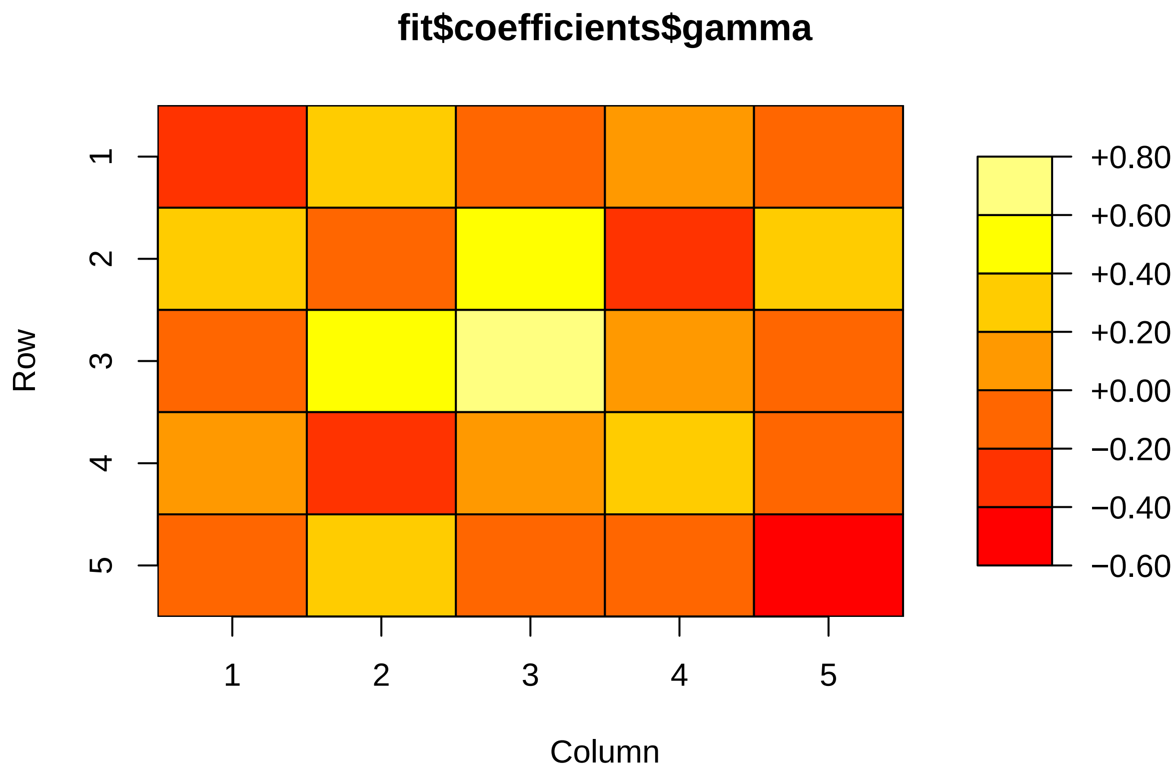

```r
short_range <- c(0, 10)
medium_range <- c(0, 10)
long_range <- c(0, 10)

model <- "square_bump"
medium_range_model <- "square_exponential"

fit <- ppjsdm::gibbsm(configuration,
                      window = window,
                      short_range = short_range,
                      medium_range = medium_range,
                      long_range = long_range,
                      model = model,
                      medium_range_model = medium_range_model,
```

```
                       use_glmnet = FALSE)
#> $beta0
#> [1] -4.447082 -5.400677 -7.009141 -5.509288 -3.150485
#>
#> $alpha
#>             [,1]        [,2]         [,3]          [,4]          [,5]
#> [1,]  0.32158228  0.05866608 -0.037897561 -0.3201649446 -0.0790902514
#> [2,]  0.05866608  0.23530055  0.067063810 -0.0787089981 -0.0374425354
#> [3,] -0.03789756  0.06706381  0.002565087  0.2382941391 -0.1675383387
#> [4,] -0.32016494 -0.07870900  0.238294139  0.1938443426  0.0008356026
#> [5,] -0.07909025 -0.03744254 -0.167538339  0.0008356026 -0.3085872031
#>
#> $gamma
#>            [,1]       [,2]       [,3]        [,4]        [,5]
#> [1,] -0.3583673  0.2653982 -0.1030516  0.15022472 -0.17176553
#> [2,]  0.2653982 -0.1826036  0.4959855 -0.20444113  0.22963138
#> [3,] -0.1030516  0.4959855  0.7001943  0.13230223 -0.12107518
#> [4,]  0.1502247 -0.2044411  0.1323022  0.38074749 -0.06868716
#> [5,] -0.1717655  0.2296314 -0.1210752 -0.06868716 -0.40615496
#>
#> $beta
#>
#> [1,]
#> [2,]
#> [3,]
#> [4,]
#> [5,]
print(fit$coefficients)
#> $beta0
#> [1] -4.447082 -5.400677 -7.009141 -5.509288 -3.150485
#>
#> $alpha
#>             [,1]        [,2]         [,3]          [,4]          [,5]
#> [1,]  0.32158228  0.05866608 -0.037897561 -0.3201649446 -0.0790902514
#> [2,]  0.05866608  0.23530055  0.067063810 -0.0787089981 -0.0374425354
#> [3,] -0.03789756  0.06706381  0.002565087  0.2382941391 -0.1675383387
#> [4,] -0.32016494 -0.07870900  0.238294139  0.1938443426  0.0008356026
#> [5,] -0.07909025 -0.03744254 -0.167538339  0.0008356026 -0.3085872031
#>
#> $gamma
#>            [,1]       [,2]       [,3]        [,4]        [,5]
#> [1,] -0.3583673  0.2653982 -0.1030516  0.15022472 -0.17176553
#> [2,]  0.2653982 -0.1826036  0.4959855 -0.20444113  0.22963138
#> [3,] -0.1030516  0.4959855  0.7001943  0.13230223 -0.12107518
#> [4,]  0.1502247 -0.2044411  0.1323022  0.38074749 -0.06868716
#> [5,] -0.1717655  0.2296314 -0.1210752 -0.06868716 -0.40615496
#>
#> $beta
#>
#> [1,]
#> [2,]
#> [3,]
#> [4,]
```

```
#> [5,]
#>
#> $short_range
#>          [,1]     [,2]     [,3]     [,4]     [,5]
#> [1,] 4.345604 5.508050 5.508050 5.508050 5.508050
#> [2,] 5.508050 4.814746 5.508050 5.508050 5.508050
#> [3,] 5.508050 5.508050 2.699721 5.508050 5.508050
#> [4,] 5.508050 5.508050 5.508050 1.939803 5.508050
#> [5,] 5.508050 5.508050 5.508050 5.508050 6.107433
#>
#> $medium_range
#>           [,1]     [,2]     [,3]     [,4]      [,5]
#> [1,] 10.761297 8.602712 8.602712 8.602712  8.602712
#> [2,]  8.602712 7.071343 8.602712 8.602712  8.602712
#> [3,]  8.602712 8.602712 6.091297 8.602712  8.602712
#> [4,]  8.602712 8.602712 8.602712 6.787958  8.602712
#> [5,]  8.602712 8.602712 8.602712 8.602712 11.617277
#>
#> $long_range
#>          [,1]     [,2]     [,3]     [,4]     [,5]
#> [1,] 17.77479 12.89652 12.89652 12.89652 12.89652
#> [2,] 12.89652 11.71936 12.89652 12.89652 12.89652
#> [3,] 12.89652 12.89652 11.33403 12.89652 12.89652
#> [4,] 12.89652 12.89652 12.89652 13.47600 12.89652
#> [5,] 12.89652 12.89652 12.89652 12.89652 15.75529
par(mar = c(5.1, 5.1, 4.1, 4.1))
plot(fit$coefficients$alpha)
```




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




```r
print(fit$aic)
#> [1] 1707.063
print(fit$bic)
#> [1] 1914.457
```

We may then plot the corresponding Papangelou conditional intensity.

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