

## Queensland rainforest

```
library(ppjsdm)
library(spatstat)
#> Loading required package: spatstat.data
#> Loading required package: nlme
#> Loading required package: rpart
#>
#> spatstat 1.62-2      (nickname: 'Shape-shifting lizard')
#> For an introduction to spatstat, type 'beginner'
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.

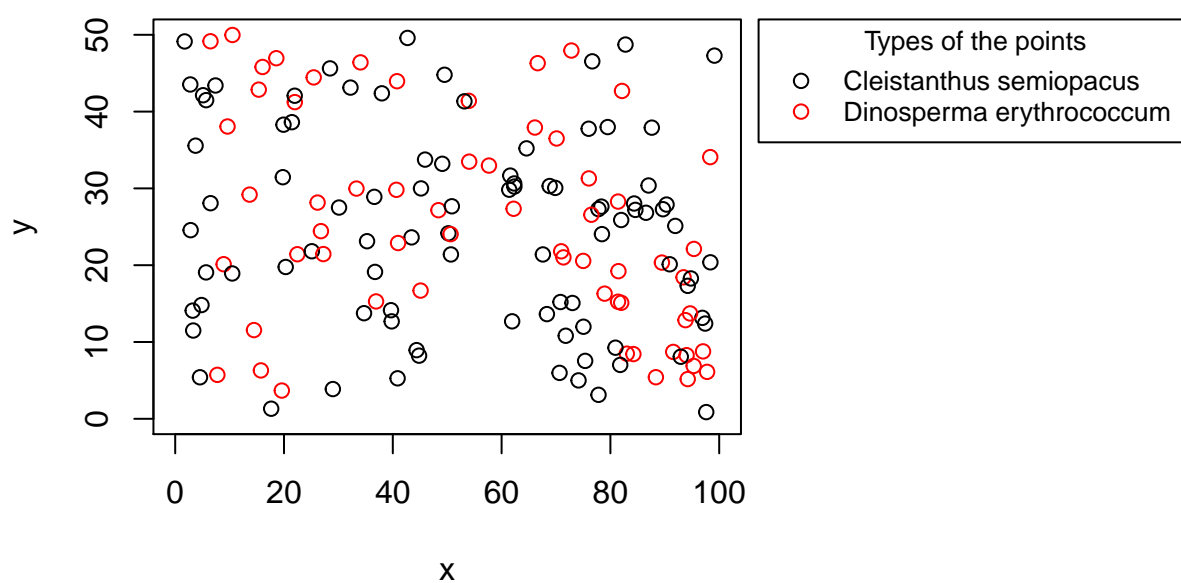
```
index_of_plot <- 3 # Between 1 and 20
year <- 2011 # Year of census
number_of_species <- 2 # Includes the most prevalent species from the plot

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.

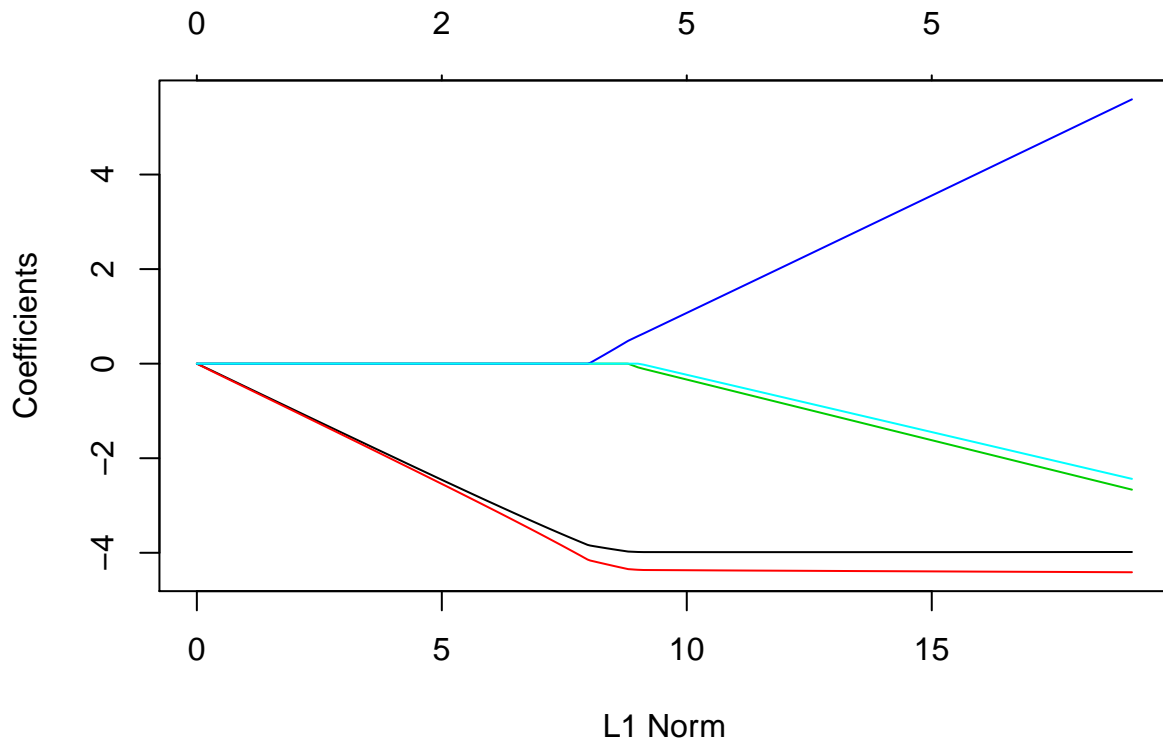
```
par(mar = c(5, 4, 4, 13) + 0.1)
plot(configuration, window = window)
```

## Points in the configuration



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

```
fit <- ppjsdm::gibbsm(configuration, window = window, use_glmnet = TRUE)
#> 6 x 1 sparse Matrix of class "dgCMatrix"
#>
#> (Intercept)      .
#> log_lambda_1 -3.983406
#> log_lambda_2 -4.411316
#> alpha_1_1    -2.664734
#> alpha_1_2     5.590466
#> alpha_2_2    -2.436428
fit <- fit$complete
plot(fit)
```



By default, the function fits the model that was introduced in the ARC grant [TODO: Add reference]. This model has many drawbacks, the most important of which is that the model with the fitted values is degenerate, as can be seen by drawing from the model.

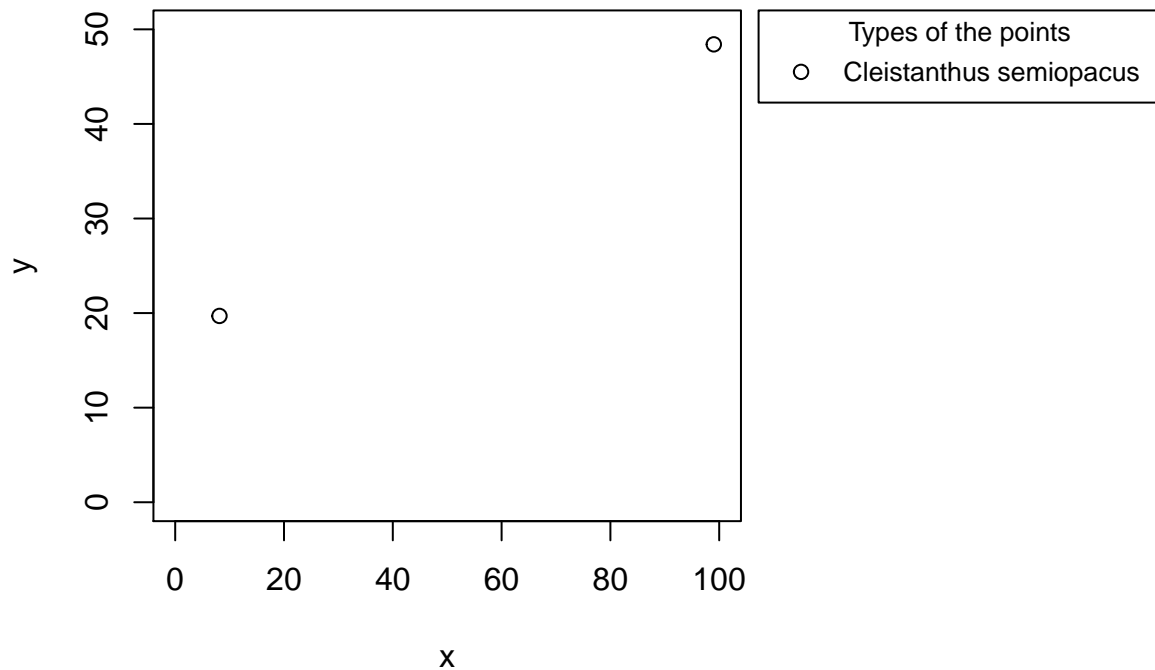
```

coefs <- coefficients(fit, s = 0)[-1]
print(coefs)
#> [1] -3.983406 -4.411316 -2.664734  5.590466 -2.436428
lambda <- exp(coefs[1:2])
alpha <- matrix(c(coefs[3],
                  coefs[4],
                  coefs[4],
                  coefs[5]),
                ncol = number_of_species,
                nrow = number_of_species)
draw <- ppjsdm::rgibbs(window = window,
                      alpha = alpha,
                      lambda = lambda,
                      types = levels(types(configuration)),
                      steps = 1000000)

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

```

## Points in the configuration

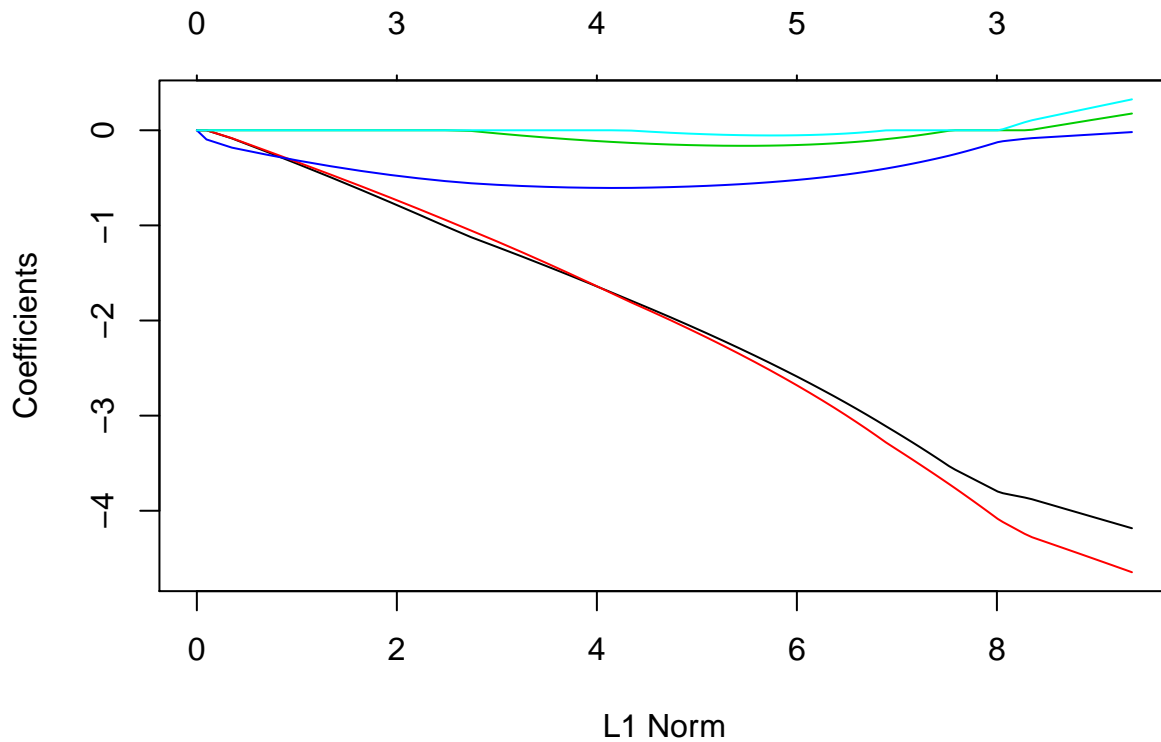


The Geyer model is better suited to most situations, but the user needs to specify some additional parameters before the fitting may take place.

```
radii <- matrix(5, number_of_species, number_of_species)
```

The matrix `radii` models interaction radii within a species, and between species. An interaction radius of 5 gives good results in the fitting procedure.

```
fit <- ppjsdm::gibbsm(configuration,
                      window = window,
                      model = "Geyer",
                      radius = radii,
                      use_glmnet = TRUE)
#> 6 x 1 sparse Matrix of class "dgCMatrix"
#>      1
#> (Intercept)      .
#> log_lambda_1 -4.18398487
#> log_lambda_2 -4.64642500
#> alpha_1_1      0.17595439
#> alpha_1_2     -0.01892858
#> alpha_2_2      0.32513234
fit <- fit$complete
plot(fit)
```



Note that in this case the model is not degenerate; indeed we can sample from it as follows.

```

coefs <- coefficients(fit, s = 0)[-1]
print(coefs)
#> [1] -4.18398487 -4.64642500  0.17595439 -0.01892858  0.32513234
lambda <- exp(coefs[1:2])
alpha <- matrix(c(coefs[3],
                  coefs[4],
                  coefs[4],
                  coefs[5]),
                ncol = number_of_species,
                nrow = number_of_species)
draw <- ppjsdm::rgibbs(window = window,
                      alpha = alpha,
                      lambda = lambda,
                      model = "Geyer",
                      radius = radii,
                      types = levels(types(configuration)))

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

```

## Points in the configuration

