

Queensland rainforest

```
library(ppjsdm)
library(spatstat)
#> Loading required package: spatstat.data
#> Loading required package: nlme
#> Loading required package: rpart
#>
#> spatstat 1.63-0      (nickname: 'Space camouflage')
#> 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.

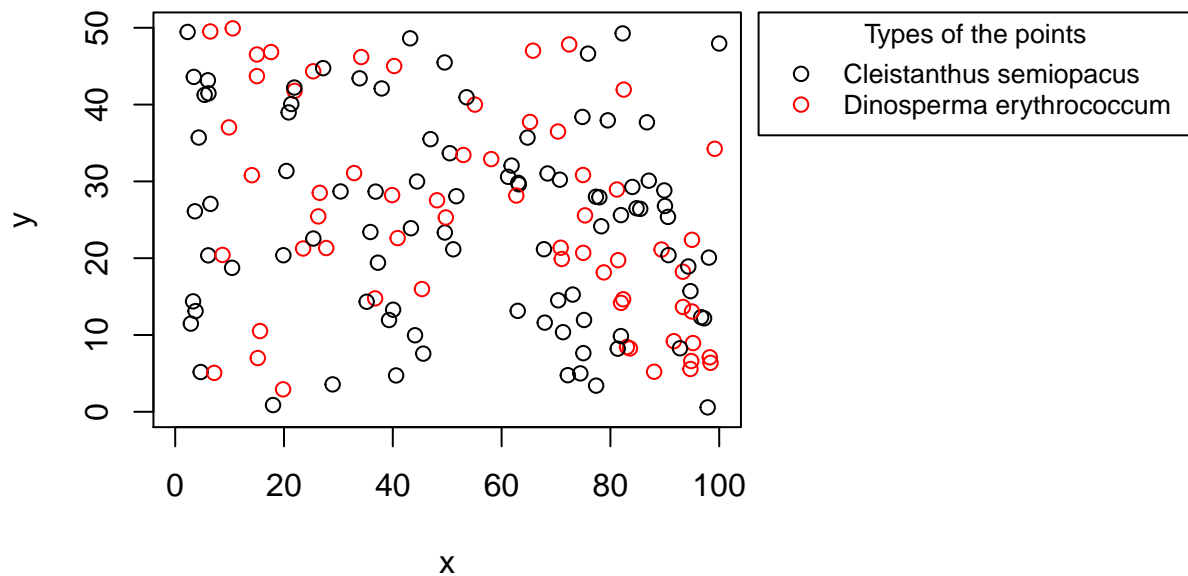
```
index_of_plot <- 3 # Between 1 and 20
year <- 2011 # Year of census
number_of_species <- 2 # Includes the most prevalent species from the plot

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

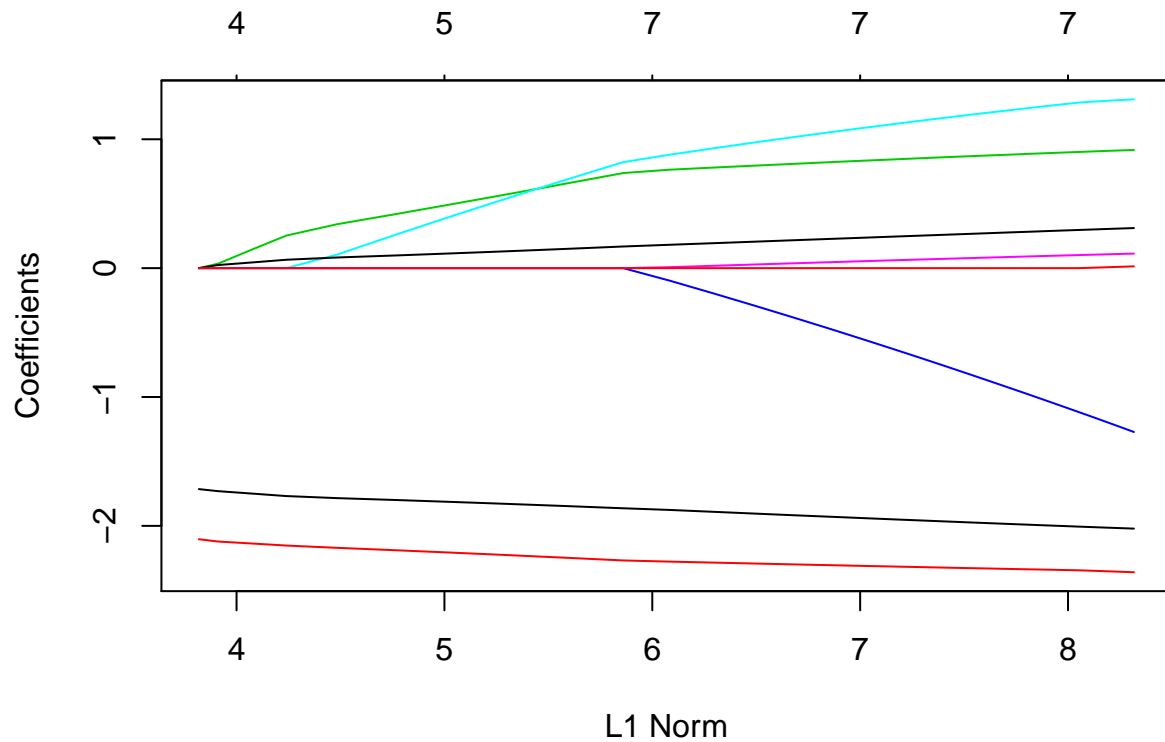


The function `gibbsm` fits a multivariate Gibbs point process to our dataset. For example,

```
short_range <- matrix(0.5, number_of_species, number_of_species)
medium_range <- matrix(1, number_of_species, number_of_species)
long_range <- matrix(3, number_of_species, number_of_species)

fit <- ppjsdm::gibbsm(configuration,
                      short_range = short_range,
                      medium_range = medium_range,
                      long_range = long_range,
                      window = window,
                      use_glmnet = TRUE)

#> 9 x 1 sparse Matrix of class "dgCMatrix"
#> 1
#> (Intercept) .
#> log_lambda1 -4.32404670
#> log_lambda2 -4.66243293
#> alpha_1_1 0.91696459
#> alpha_1_2 -1.27163170
#> alpha_2_2 1.30971762
#> gamma_1_1 0.11276433
#> gamma_1_2 0.31100574
#> gamma_2_2 0.01380922
plot(fit$complete)
```



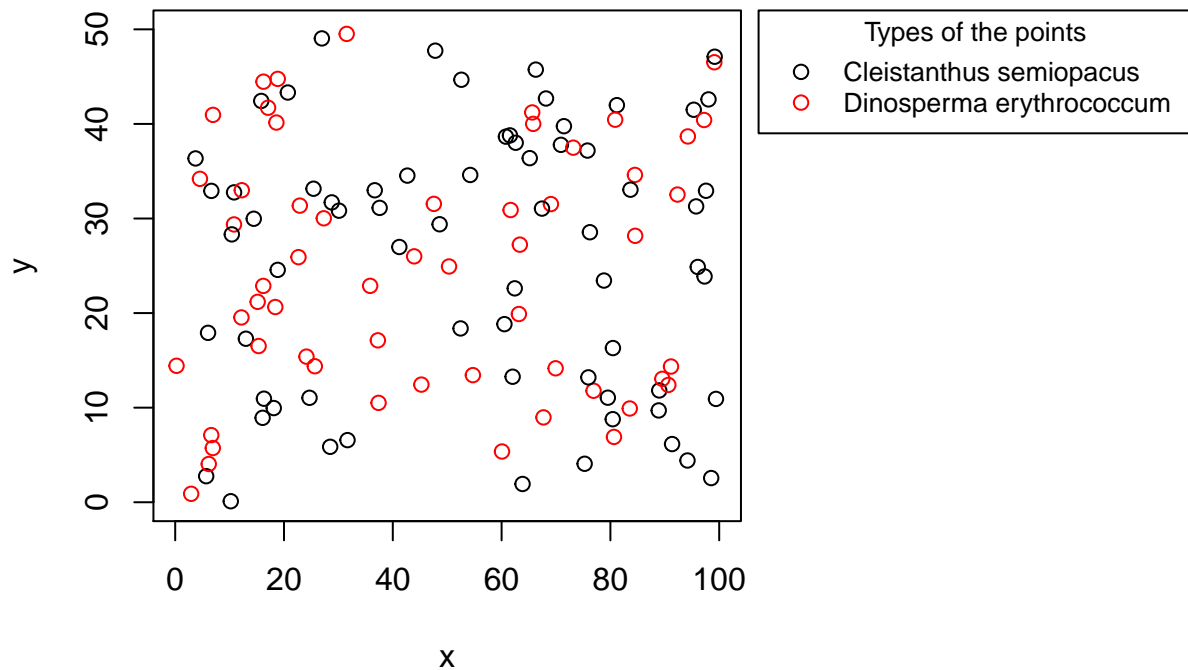
It is then possible to draw from the model, as can be seen below.

```
parameters <- get_parameters_from_fit(fit)
lambda <- parameters$lambda
alpha <- parameters$alpha
gamma <- parameters$gamma
draw <- ppjsdm::rgibbs(window = window,
                        alpha = alpha,
                        lambda = lambda,
                        gamma = gamma,
                        short_range = short_range,
                        medium_range = medium_range,
                        long_range = long_range,
                        types = levels(types(configuration)))

print(draw)
#> An S3 object representing a configuration.
#>
#> Number of points: 122.

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

Points in the configuration



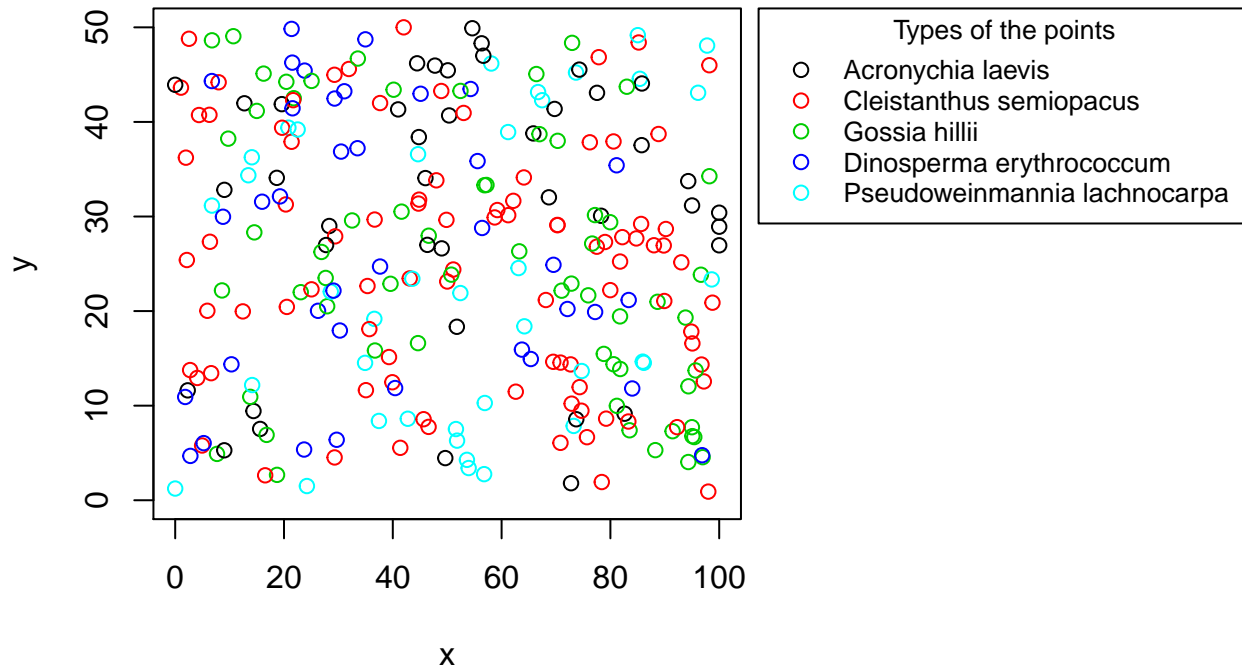
Let us increase the number of species accounted for.

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

Points in the configuration



```

short_range <- matrix(4, number_of_species, number_of_species)
medium_range <- diag(0, number_of_species)
long_range <- medium_range + diag(0, number_of_species)

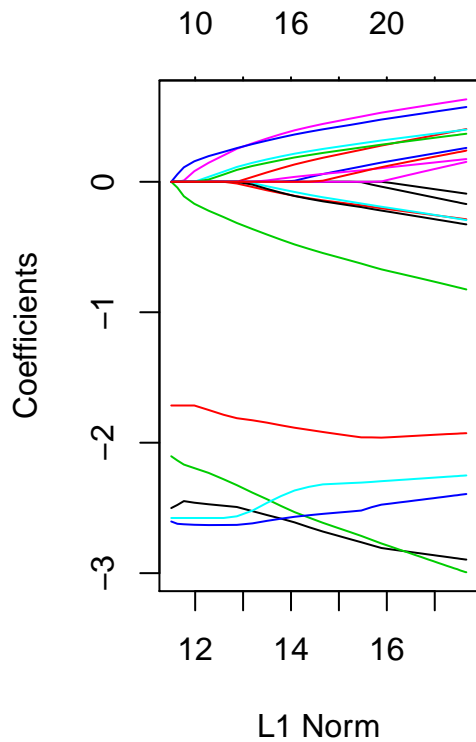
model <- "square_bump"
medium_range_model <- "square_exponential"

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 = TRUE)

#> 36 x 1 sparse Matrix of class "dgCMatrix"
#>      1
#> (Intercept) .
#> log_lambda1 -5.19841861
#> log_lambda2 -4.22996378
#> log_lambda3 -5.29547734
#> log_lambda4 -4.69598008
#> log_lambda5 -4.55360518
#> alpha_1_1 0.63199839
#> alpha_1_2 -0.09104299
#> alpha_1_3 0.40413949

```

```
#> alpha_1_4 -0.82532399
#> alpha_1_5 0.25969326
#> alpha_2_2 0.40046692
#> alpha_2_3 0.17405108
#> alpha_2_4 -0.17042999
#> alpha_2_5 -0.28753755
#> alpha_3_3 0.36808600
#> alpha_3_4 0.57313142
#> alpha_3_5 -0.29412684
#> alpha_4_4 0.15281379
#> alpha_4_5 -0.32649943
#> alpha_5_5 0.24000007
#> gamma_1_1 .
#> gamma_1_2 .
#> gamma_1_3 .
#> gamma_1_4 .
#> gamma_1_5 .
#> gamma_2_2 .
#> gamma_2_3 .
#> gamma_2_4 .
#> gamma_2_5 .
#> gamma_3_3 .
#> gamma_3_4 .
#> gamma_3_5 .
#> gamma_4_4 .
#> gamma_4_5 .
#> gamma_5_5 .
plot(fit$complete)
```



We may then plot the corresponding Papangelou conditional intensity.

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

as.im(t(z), W = window)

