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)
```

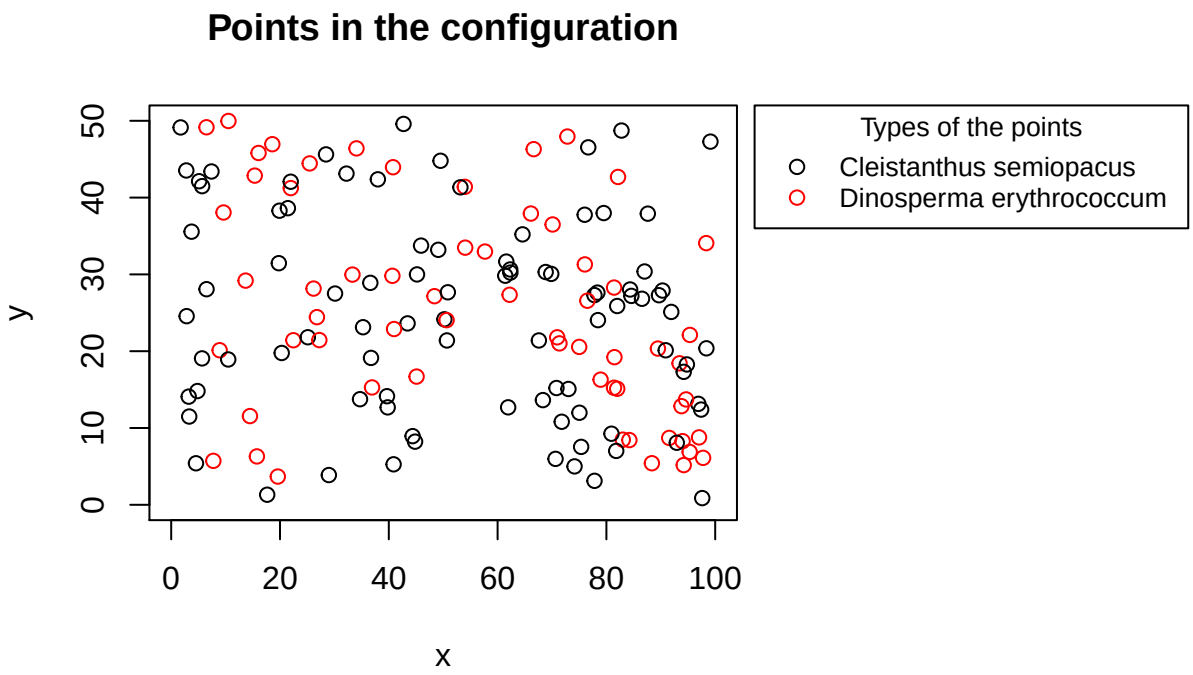
This vignette explains how to use the `ppjsdm` package with the Queensland rainforest dataset from CSIRO. We begin by loading the data with the most prevalent species.

```
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)
```



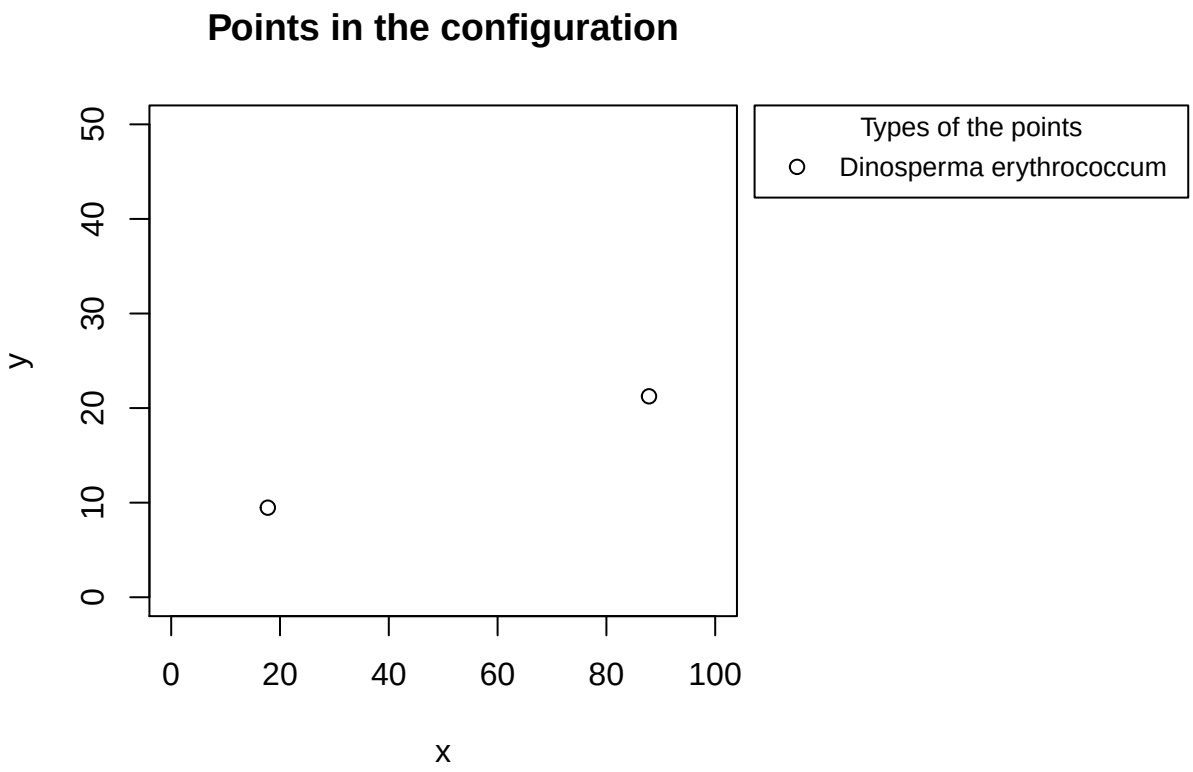
The function `gibbsm` fits a multivariate Gibbs point process to our dataset. For example,

```
fit <- ppjsdm::gibbsm(configuration, window = window)
#> log_lambda_1 log_lambda_2    alpha_1_1    alpha_1_2    alpha_2_2
#>    -3.970526    -4.430065    -4.710366     9.591091    -4.374614
```

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.

```
lambda <- exp(fit$coefficients[1:number_of_species])
alpha <- matrix(c(fit$coefficients[3],
                  fit$coefficients[4],
                  fit$coefficients[4],
                  fit$coefficients[5]), ncol = number_of_species, nrow = number_of_species)
draw <- ppjsdm::rmultigibbs(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)
```



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.

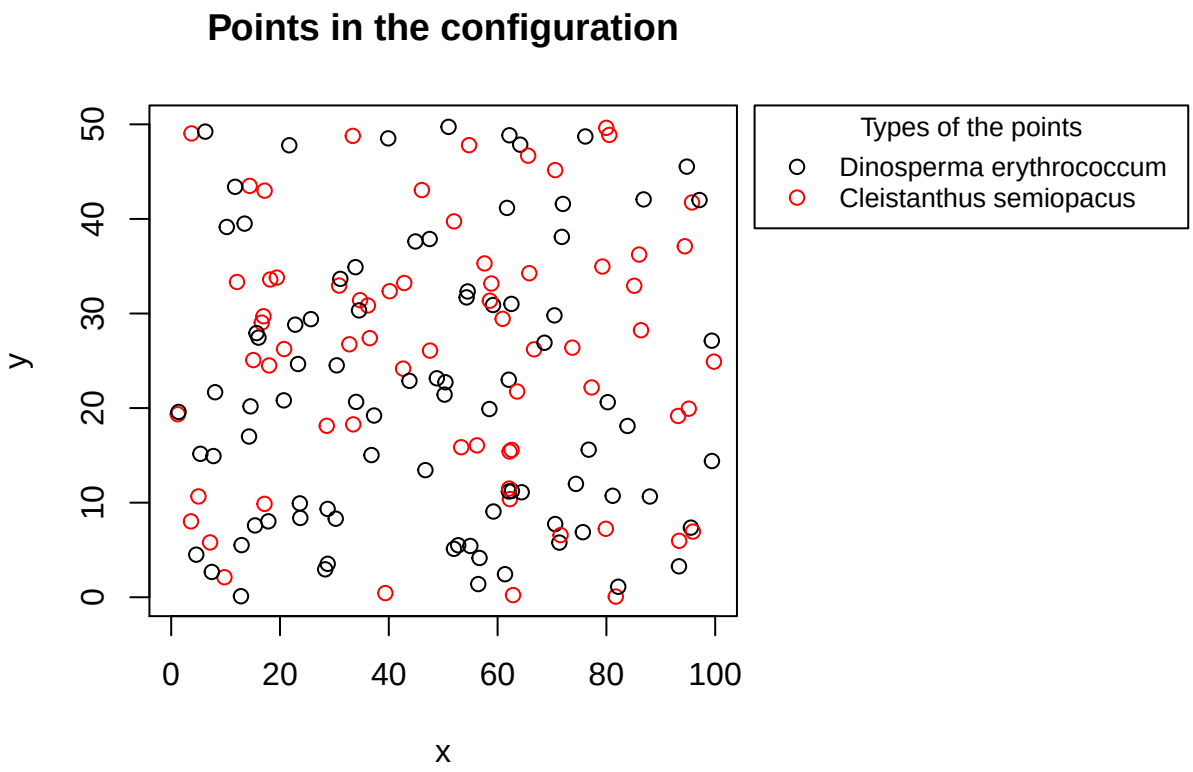
```
fit <- ppjsdm::gibbsm(configuration, window = window, model = "Geyer", radius = radii)
#> log_lambda_1 log_lambda_2    alpha_1_1    alpha_1_2    alpha_2_2
#>  -4.12136802  -4.71144863   0.12682397  -0.05099062   0.41186500
summary(fit)
#>
#> Call:
#> glm(formula = as.formula(gibbsm_data$formula), family = binomial(),
#>     data = as.data.frame(gibbsm_data$data))
#>
#> Deviance Residuals:
#>     Min       1Q   Median       3Q      Max
#> -0.8374  -0.6849  -0.6447  -0.5681   1.9735
#>
#> Coefficients:
#>              Estimate Std. Error z value Pr(>|z|)
#> log_lambda_1 -4.12137    0.22632 -18.210   <2e-16 ***
#> log_lambda_2 -4.71145    0.25094 -18.775   <2e-16 ***
#> alpha_1_1     0.12682    0.14904   0.851   0.3948
#> alpha_1_2    -0.05099    0.11519  -0.443   0.6580
```

```
#> alpha_2_2     0.41187    0.17937   2.296   0.0217 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for binomial family taken to be 1)
#>
#>     Null deviance: 3459.79  on 755  degrees of freedom
#> Residual deviance:  749.44  on 750  degrees of freedom
#> AIC: 759.44
#>
#> Number of Fisher Scoring iterations: 4
```

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

```
lambda <- exp(fit$coefficients[1:number_of_species])
alpha <- matrix(c(fit$coefficients[3],
                  fit$coefficients[4],
                  fit$coefficients[4],
                  fit$coefficients[5]), ncol = number_of_species, nrow = number_of_species)
draw <- ppjsdm::rmultigibbs(window = window,
                            alpha = alpha,
                            lambda = lambda,
                            model = "Geyer",
                            radius = radii,
                            types = levels(types(configuration)),
                            steps = 1000000)
```

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


Points in the configuration
Types of the points
Dinosperma erythrococcum
Cleistanthus semiopacus
y
x
0
10
20
30
40
50
0
20
40
60
80
100