Fithian et al. (2014) NSW

```r
library(maptools)
#> Loading required package: sp
#> Checking rgeos availability: FALSE
#>      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcl
#>      which has a restricted licence. It is disabled by default;
#>      to enable gpclib, type gpclibPermit()
library(ppjsdm)
library(raster)
library(sf)
#> Linking to GEOS 3.6.2, GDAL 2.2.3, PROJ 4.9.3
library(spatstat)
#> Loading required package: spatstat.data
#> Loading required package: nlme
#>
#> Attaching package: 'nlme'
#> The following object is masked from 'package:raster':
#>
#>     getData
#> Loading required package: rpart
#>
#> spatstat 1.63-0       (nickname: 'Space camouflage')
#> For an introduction to spatstat, type 'beginner'
#>
#> Attaching package: 'spatstat'
#> The following objects are masked from 'package:raster':
#>
#>     area, rotate, shift
remove(list = ls())

source("../R/get_nsw.R")

set.seed(1)
```

This vignette explains how to use the `ppjsdm` package with the NSW dataset from Fithian et al. (2014). We begin by loading the data with only the most prevalent species.

```r
number_of_species <- 2 # Includes the most prevalent species from the plot

nsw <- get_nsw(prevalent = number_of_species)
configuration <- nsw$configuration
window <- nsw$window
covariates <- nsw$covariates
```

Some of the covariates give `NA` values at some locations in the configuration, so we have to remove these points from the configuration.

```r
for(covariate in append(covariates, list(window))) {
  ok <- inside.owin(x_coordinates(configuration), y_coordinates(configuration), covariate)
```

```
  configuration <- Configuration(x_coordinates(configuration)[ok],
                                 y_coordinates(configuration)[ok],
                                 types(configuration)[ok])
}
```

The point configuration is plotted below.

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

Points in the configuration
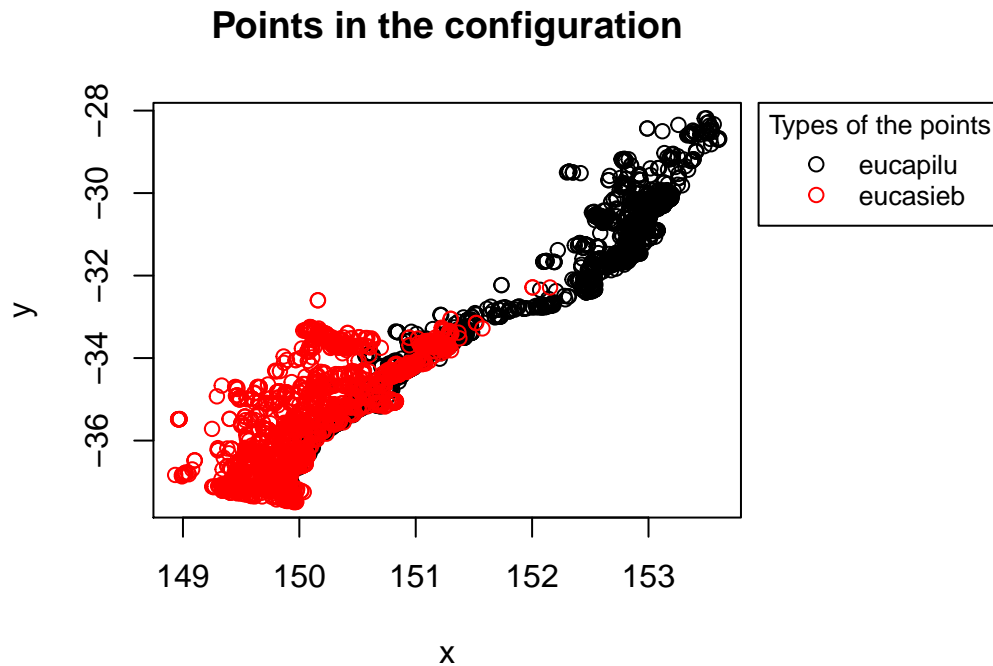
The matrix `radii` defined below models interaction radii within a species (on the diagonal), and between species (outside the diagonal).

```
radii <- matrix(0.1, number_of_species, number_of_species)
```

Fitting the model to the dataset is then quite easy.

```
fit <- ppjsdm::gibbsm(configuration,
                      window = window,
                      covariates = covariates,
                      model = "Geyer",
                      radius = radii,
                      use_glmnet = TRUE)
#> 36 x 1 sparse Matrix of class "dgCMatrix"
#>                              1
#> (Intercept)           .
#> log_lambda1           8.051799e+00
#> log_lambda2           9.333398e+00
#> alpha_1_1             2.482746e+00
#> alpha_1_2             1.640093e-01
#> alpha_2_2             2.940043e+00
```
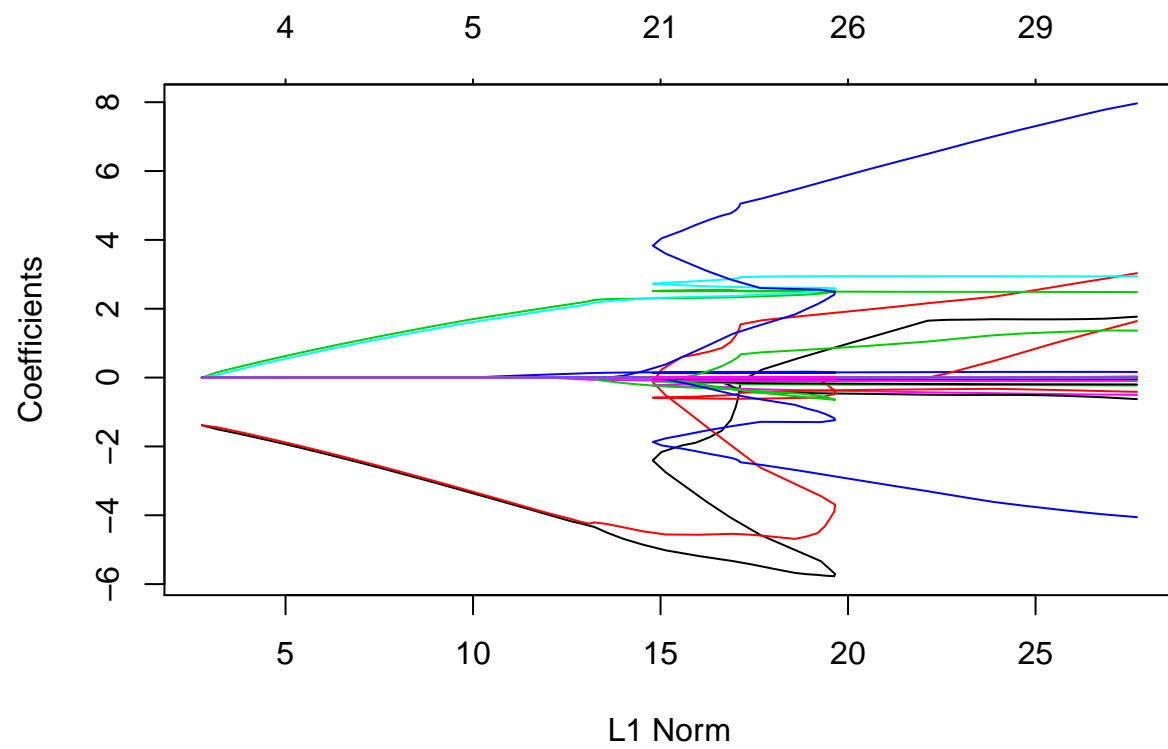
```
#> unnamed_covariate1_1  -5.035252e-01
#> unnamed_covariate1_2    .
#> unnamed_covariate2_1  -6.264725e-01
#> unnamed_covariate2_2  -4.148911e-01
#> unnamed_covariate3_1    .
#> unnamed_covariate3_2  -2.288268e-01
#> unnamed_covariate4_1   2.062387e-03
#> unnamed_covariate4_2  -1.998219e-04
#> unnamed_covariate5_1   3.333867e-02
#> unnamed_covariate5_2  -1.992398e-01
#> unnamed_covariate6_1    .
#> unnamed_covariate6_2    .
#> unnamed_covariate7_1   1.640687e+00
#> unnamed_covariate7_2   1.364919e+00
#> unnamed_covariate8_1    .
#> unnamed_covariate8_2   7.966527e+00
#> unnamed_covariate9_1  -3.440542e-02
#> unnamed_covariate9_2  -9.620604e-02
#> unnamed_covariate10_1 -1.108561e-02
#> unnamed_covariate10_2 -6.486333e-03
#> unnamed_covariate11_1 -5.927619e-02
#> unnamed_covariate11_2 -4.054196e+00
#> unnamed_covariate12_1 -1.941785e-02
#> unnamed_covariate12_2 -3.045412e-03
#> unnamed_covariate13_1 -3.718314e-02
#> unnamed_covariate13_2  1.150573e-02
#> unnamed_covariate14_1 -2.032925e-04
#> unnamed_covariate14_2 -2.467814e-04
#> unnamed_covariate15_1 -2.297123e-05
#> unnamed_covariate15_2 -6.647723e-05
plot(fit$complete)
```

```r
plot(fit$cv)
```