

Fithian et al. (2014) NSW

```
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 1.62-2 (nickname: 'Shape-shifting lizard')
#> 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)
```

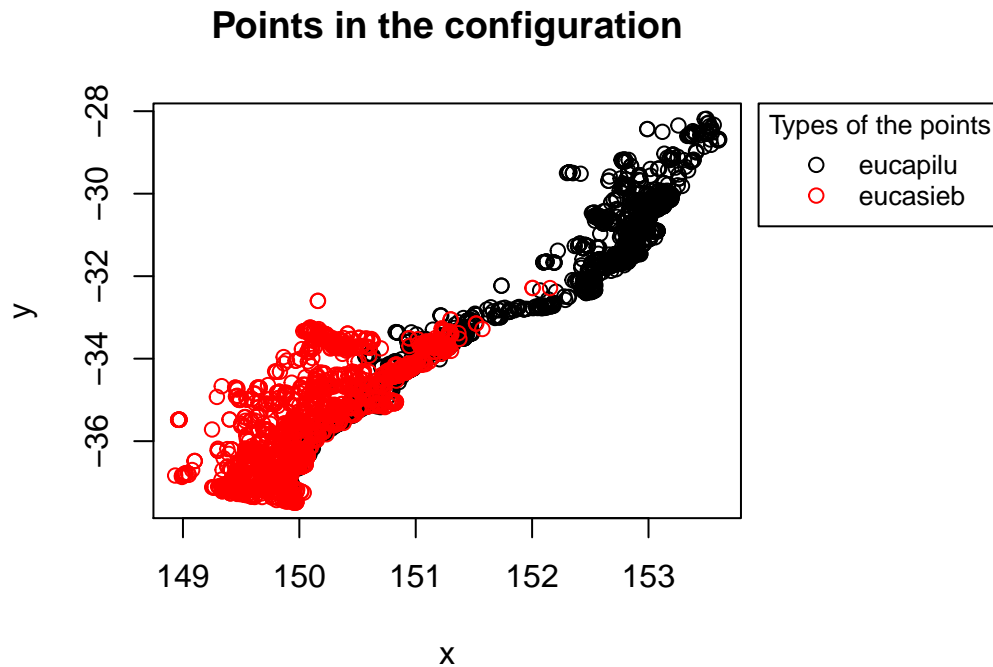
This vignette explains how to use the `ppjsdm` package with the NSW dataset from Fithian et al. (2014). We begin by loading the data with only the most prevalent species.

```
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
```

The point configuration is plotted below.

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



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)
```

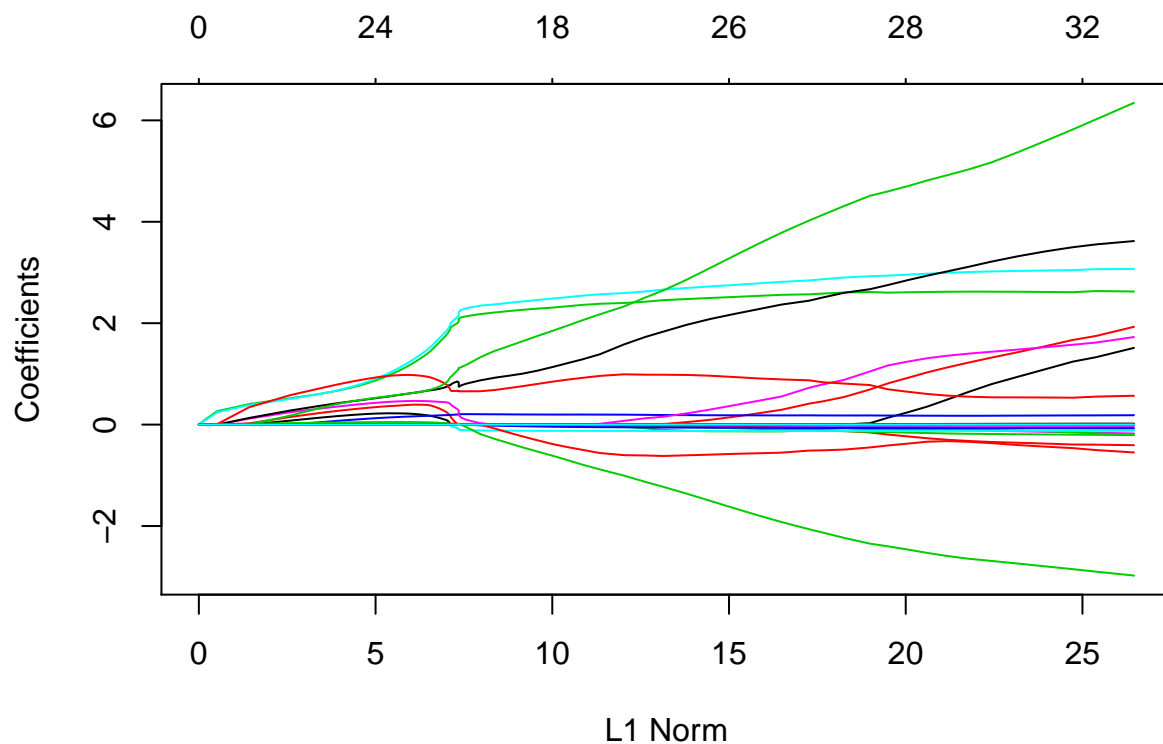
Fitting the model to the dataset is then quite easy.

```
fit <- ppjsdm::gibbsm(configuration,
  window = window,
  covariates = covariates,
  model = "Geyer",
  radius = radii,
  use_glmnet = TRUE)
#> 36 x 1 sparse Matrix of class "dgCMatrix"
#>
#> (Intercept)
#> log_lambda_1 1.512493e+00
#> log_lambda_2 1.929830e+00
#> alpha_1_1 2.624344e+00
#> alpha_1_2 1.854130e-01
#> alpha_2_2 3.069858e+00
#> unnamed_covariate1_1 -1.755234e-01
#> unnamed_covariate1_2 .
#> unnamed_covariate2_1 -4.060712e-01
#> unnamed_covariate2_2 -1.988787e-01
#> unnamed_covariate3_1 .
#> unnamed_covariate3_2 -8.520734e-02
#> unnamed_covariate4_1 1.883980e-03
#> unnamed_covariate4_2 -7.127276e-04
#> unnamed_covariate5_1 8.530415e-03
```

```

#> unnamed_covariate5_2 -2.064711e-01
#> unnamed_covariate6_1 8.454234e-03
#> unnamed_covariate6_2 6.679143e-05
#> unnamed_covariate7_1 1.726491e+00
#> unnamed_covariate7_2 3.620651e+00
#> unnamed_covariate8_1 5.670274e-01
#> unnamed_covariate8_2 6.345885e+00
#> unnamed_covariate9_1 -6.887193e-02
#> unnamed_covariate9_2 -1.215930e-01
#> unnamed_covariate10_1 -4.521396e-03
#> unnamed_covariate10_2 .
#> unnamed_covariate11_1 -5.473880e-01
#> unnamed_covariate11_2 -2.979797e+00
#> unnamed_covariate12_1 -1.662031e-02
#> unnamed_covariate12_2 -1.462107e-03
#> unnamed_covariate13_1 -2.274461e-02
#> unnamed_covariate13_2 1.976707e-02
#> unnamed_covariate14_1 -6.699342e-05
#> unnamed_covariate14_2 -1.530649e-04
#> unnamed_covariate15_1 -2.045616e-05
#> unnamed_covariate15_2 -6.865772e-05
plot(fit$complete)

```



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

