Helianthemum dataset

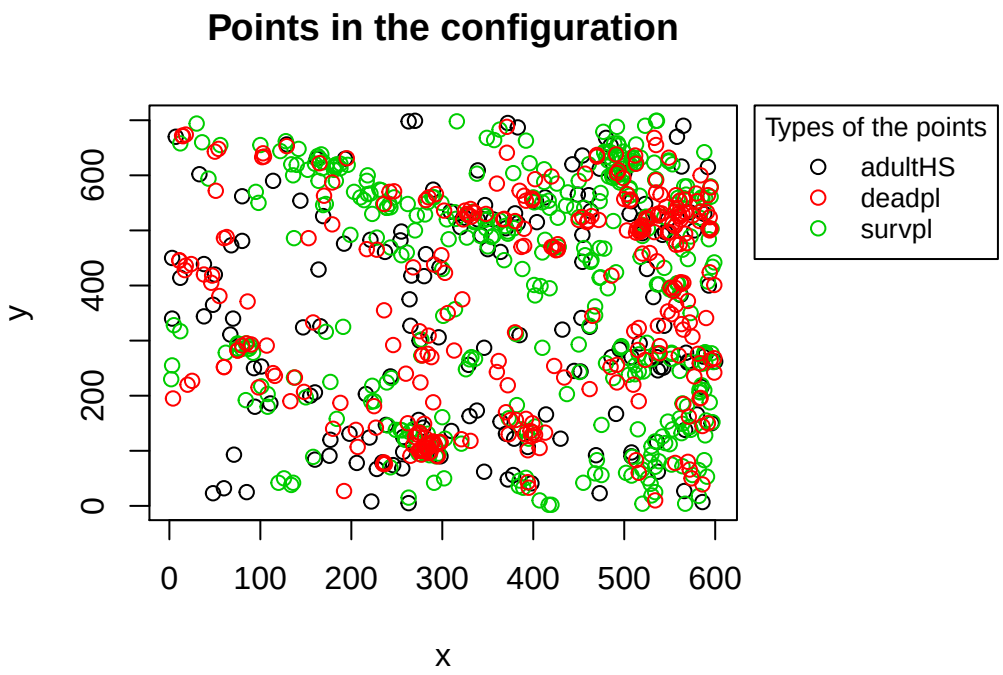
```
library(ppjsdm)
#> Registered S3 method overwritten by 'spatstat':
#>   method     from
#>   print.boxx cli
library(ecespa)
#> Loading required package: spatstat
#> Loading required package: spatstat.data
#> Loading required package: nlme
#> Loading required package: rpart
#>
#> spatstat 1.64-0       (nickname: 'Susana Distancia')
#> For an introduction to spatstat, type 'beginner'
library(plot.matrix)
remove(list = ls())

set.seed(1)
```

This vignette explains how to use the `ppjsdm` package with the `Helianthemum` dataset from `ecespa`. Locations of H. squamatum adult plants and seedlings in a 6 m x 7 m plot over gypsum soil in Chinchón (near Madrid, Spain). These are part of the data collected by Romao (2003) that have been analyzed several times (Escudero et al.2005, De la Cruz 2006, De la Cruz et al. in press.). The coordinates of the plans are given in cm.

```
configuration <- Configuration(Helianthemum)
#> Warning in Configuration(Helianthemum): There are duplicate points in the
#> configuration.
window <- Rectangle_window(c(2, 600), c(2, 699))
```

The point configuration is plotted below.

```
print(configuration)
#> An S3 object representing a configuration.
#>
#> Number of points: 866.
par(mar = c(5, 4, 4, 13) + 0.1)
plot(configuration, window = window)
```



We provide a series of ranges for the interaction radii, and let the fitting function calibrate the model.

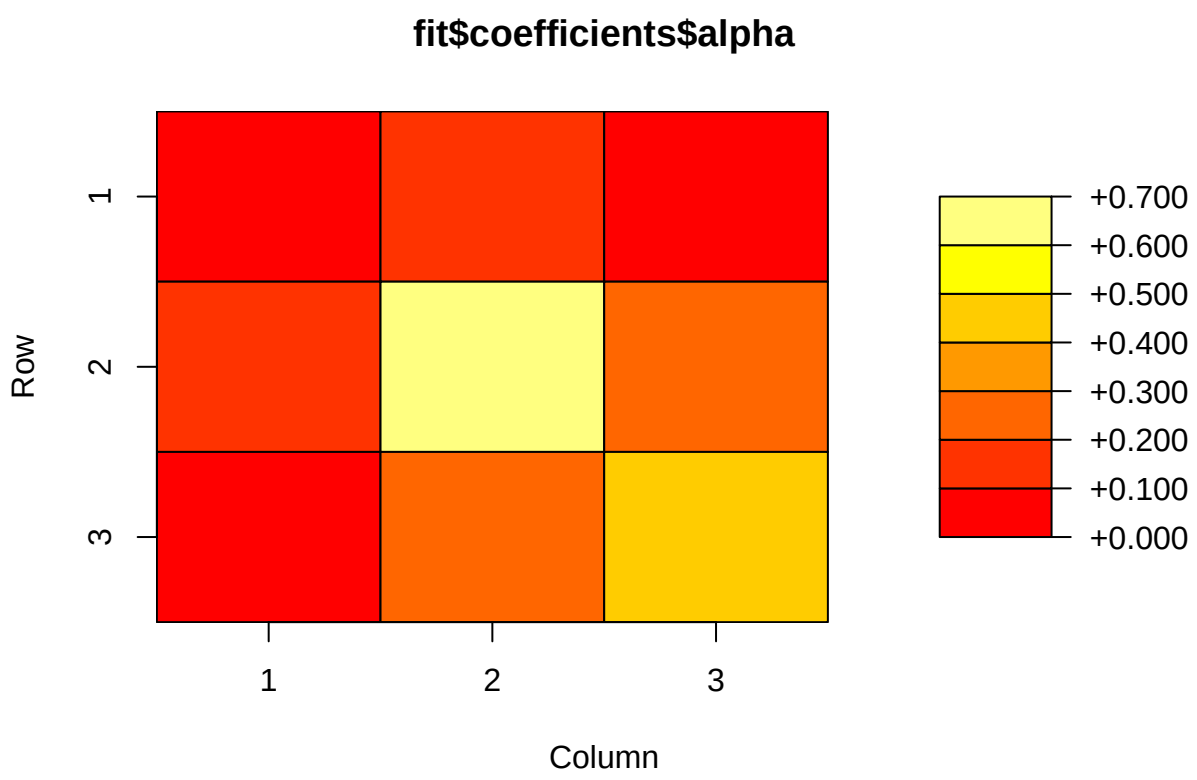
```
short_range <- c(0, 50)
medium_range <- c(0, 50)
long_range <- c(0, 50)
model <- "Geyer"
medium_range_model <- "Geyer"
saturation <- 2
max_points <- 1000
steps <- 100000
```

We can now call the fitting function.

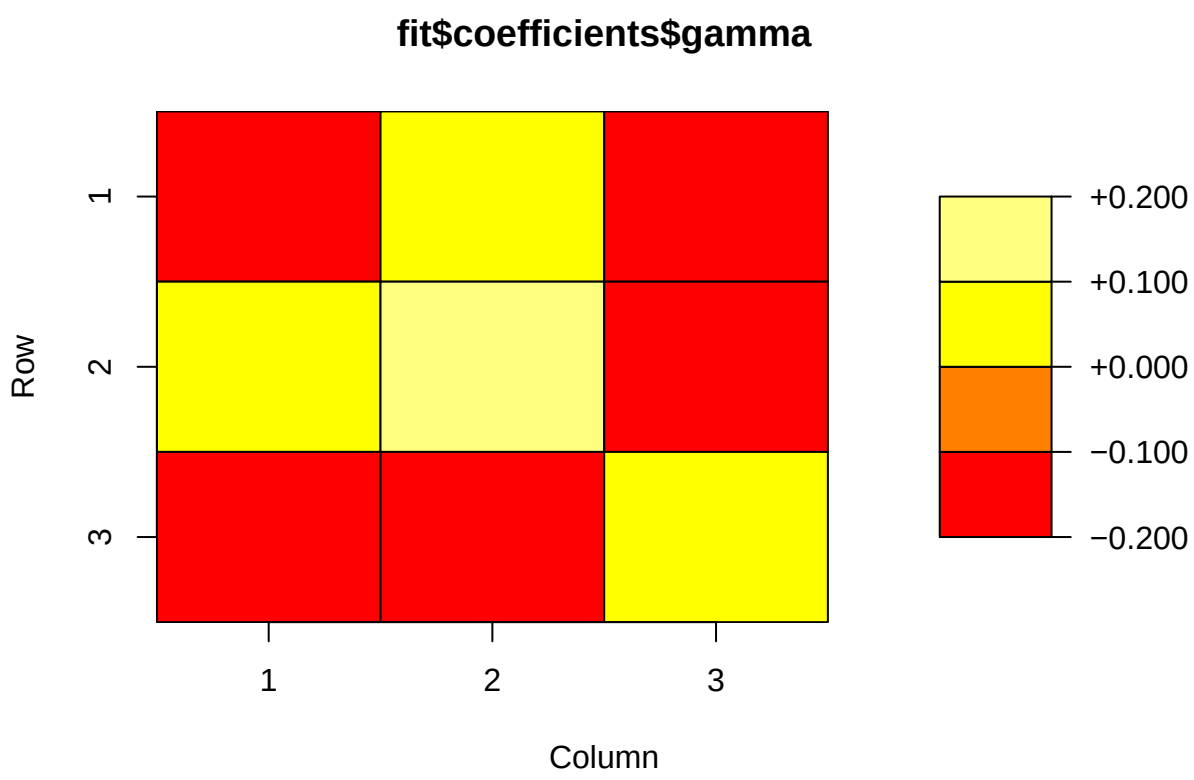
```
fit <- ppjsdm::gibbsm(configuration,
                      window = window,
                      model = model,
                      medium_range_model = medium_range_model,
                      short_range = short_range,
                      medium_range = medium_range,
                      long_range = long_range,
                      use_glmnet = FALSE,
                      use_aic = TRUE,
                      saturation = saturation)
#> $beta0
#> [1] -7.443806 -8.516448 -7.595674
#>
#> $alpha
#>             [,1]      [,2]       [,3]
#> [1,] 0.02447864 0.1018940 0.08972518
#> [2,] 0.10189400 0.6367581 0.20110183
```

```
#> [3,] 0.08972518 0.2011018 0.47966462
#>
#> $gamma
#>              [,1]         [,2]          [,3]
#> [1,] -0.173629070  0.003245451 -0.1196740701
#> [2,]  0.003245451  0.198861945 -0.1037667147
#> [3,] -0.119674070 -0.103766715  0.0004785686
#>
#> $beta
#>
#> [1,]
#> [2,]
#> [3,]
print(summary(fit))
#>              coefficients         se     CI95_lo     CI95_hi Ztest
#> log_lambda1 -7.4438056891 0.27261504 -7.97812135 -6.90949002   ***
#> log_lambda2 -8.5164482843 0.21635828 -8.94050272 -8.09239385   ***
#> log_lambda3 -7.5956739920 0.15279054 -7.89513794 -7.29621004   ***
#> alpha_1_1    0.0244786378 0.09142108 -0.15470339  0.20366066
#> alpha_1_2    0.1018940013 0.03638897  0.03057292  0.17321508    **
#> alpha_1_3    0.0897251787 0.03449798  0.02211038  0.15733998    **
#> alpha_2_2    0.6367581204 0.06118393  0.51683982  0.75667642   ***
#> alpha_2_3    0.2011018345 0.02964252  0.14300356  0.25920011   ***
#> alpha_3_3    0.4796646223 0.05040945  0.38086392  0.57846533   ***
#> gamma_1_1   -0.1736290698 0.09584359 -0.36147905  0.01422091
#> gamma_1_2    0.0032454506 0.05719100 -0.10884684  0.11533774
#> gamma_1_3   -0.1196740701 0.03785823 -0.19387483 -0.04547331    **
#> gamma_2_2    0.1988619449 0.08980920  0.02283914  0.37488475     *
#> gamma_2_3   -0.1037667147 0.03354951 -0.16952254 -0.03801088    **
#> gamma_3_3    0.0004785686 0.05506162 -0.10744023  0.10839737
#>                      Pval          Zval
#> log_lambda1 3.680444e-164 -27.305190603
#> log_lambda2  0.000000e+00 -39.362710019
#> log_lambda3  0.000000e+00 -49.712986560
#> alpha_1_1    7.888863e-01   0.267757038
#> alpha_1_2    5.108139e-03   2.800134024
#> alpha_1_3    9.298444e-03   2.600881967
#> alpha_2_2    2.296978e-25  10.407277038
#> alpha_2_3    1.167035e-11   6.784234678
#> alpha_3_3    1.810652e-21   9.515371278
#> gamma_1_1    7.004992e-02  -1.811587776
#> gamma_1_2    9.547463e-01   0.056747581
#> gamma_1_3    1.571685e-03  -3.161111261
#> gamma_2_2    2.681012e-02   2.214271343
#> gamma_2_3    1.981823e-03  -3.092942833
#> gamma_3_3    9.930653e-01   0.008691509
print(fit$coefficients)
#> $beta0
#> [1] -7.443806 -8.516448 -7.595674
#>
#> $alpha
#>             [,1]      [,2]       [,3]
#> [1,] 0.02447864 0.1018940 0.08972518
```

```
#> [2,] 0.10189400 0.6367581 0.20110183
#> [3,] 0.08972518 0.2011018 0.47966462
#>
#> $gamma
#>              [,1]         [,2]          [,3]
#> [1,] -0.173629070  0.003245451 -0.1196740701
#> [2,]  0.003245451  0.198861945 -0.1037667147
#> [3,] -0.119674070 -0.103766715  0.0004785686
#>
#> $beta
#>
#> [1,]
#> [2,]
#> [3,]
#>
#> $short_range
#>          [,1]     [,2]     [,3]
#> [1,] 20.56593 20.30669 20.30669
#> [2,] 20.30669 13.19339 20.30669
#> [3,] 20.30669 20.30669 21.55467
#>
#> $medium_range
#>          [,1]     [,2]     [,3]
#> [1,] 41.40553 43.36162 43.36162
#> [2,] 43.36162 29.47655 43.36162
#> [3,] 43.36162 43.36162 54.57887
#>
#> $long_range
#>          [,1]     [,2]     [,3]
#> [1,] 63.89003 64.26253 64.26253
#> [2,] 64.26253 54.87211 64.26253
#> [3,] 64.26253 64.26253 74.92618
par(mar=c(5.1, 5.1, 4.1, 4.1))
plot(fit$coefficients$alpha)
```




```
plot(fit$coefficients$gamma)
```




```
print(fit$aic)
#> [1] 3916.226
print(fit$bic)
#> [1] 4011.826
```

We may then plot the corresponding Papangelou conditional intensity.

```
parameters <- fit$coefficients
# plot_papangelou(window = window,
#                 configuration = configuration,
#                 type = 1,
#                 mark = mean(get_marks(configuration)),
#                 model = model,
#                 medium_range_model = medium_range_model,
#                 alpha = parameters$alpha,
#                 beta0 = parameters$beta0,
#                 beta = matrix(0, 1, 0),
#                 gamma = parameters$gamma,
#                 covariates = list(),
#                 short_range = parameters$short_range,
#                 medium_range = parameters$medium_range,
#                 long_range = parameters$long_range,
#                 saturation = saturation,
#                 max_points = max_points)
```

It is also possible to draw from the model.

```r
draw <- ppjsdm::rgibbs(window = window,
                       alpha = parameters$alpha,
                       beta0 = parameters$beta0,
                       gamma = parameters$gamma,
                       model = model,
                       medium_range_model = medium_range_model,
                       short_range = parameters$short_range,
                       medium_range = parameters$medium_range,
                       long_range = parameters$long_range,
                       types = levels(types(configuration)),
                       mark_range = c(min(get_marks(configuration)), max(get_marks(configuration))),
                       steps = steps)
print(draw)
#> An S3 object representing a configuration.
#>
#> Number of points: 1228.

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