Simulations of Ecological dataset using R

Emy Guilbault

The University of Newcastle - Faculty of Science
School of Mathematics and Physical Sciences
Emy.Guilbault@uon.edu.au

Ian Renner

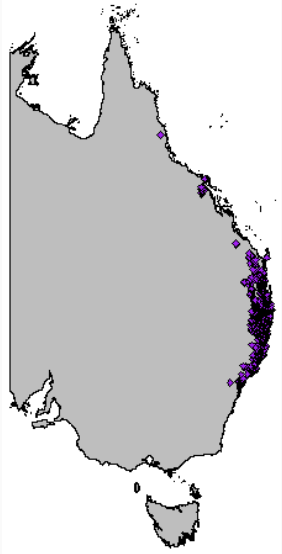
March2021

1. Data and SDM in Citizen science
2. Simulate environmental covariates
3. Aspect 1: Variation in recording due to the landscape
4. Aspect 2 : Observer recording and accuracy
5. Aspect 3: Data-dynamic

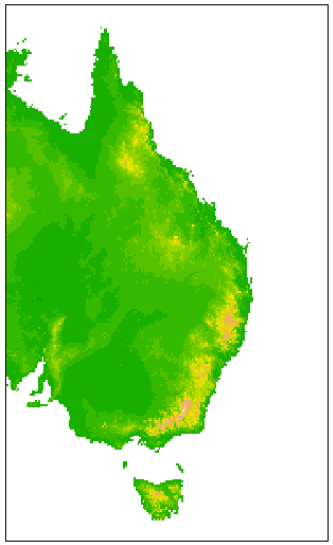
Data and SDM in Citizen science

Species distribution modelling [1]

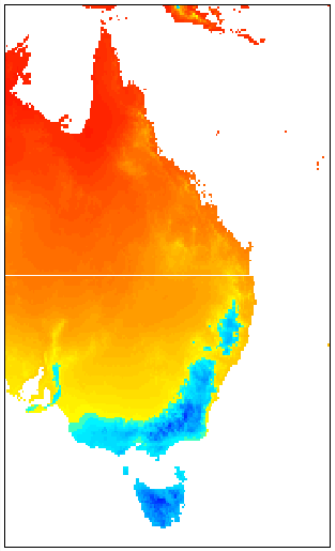
Observation



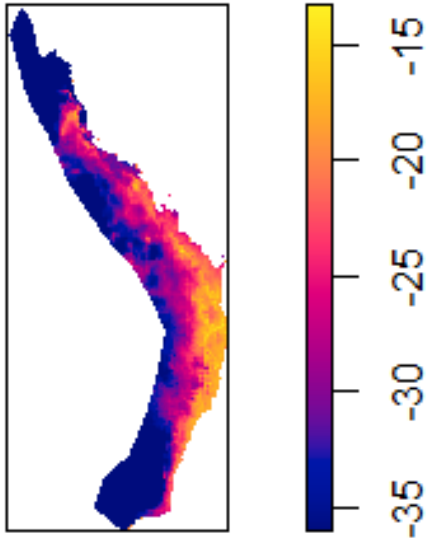
Altitude



Temperature



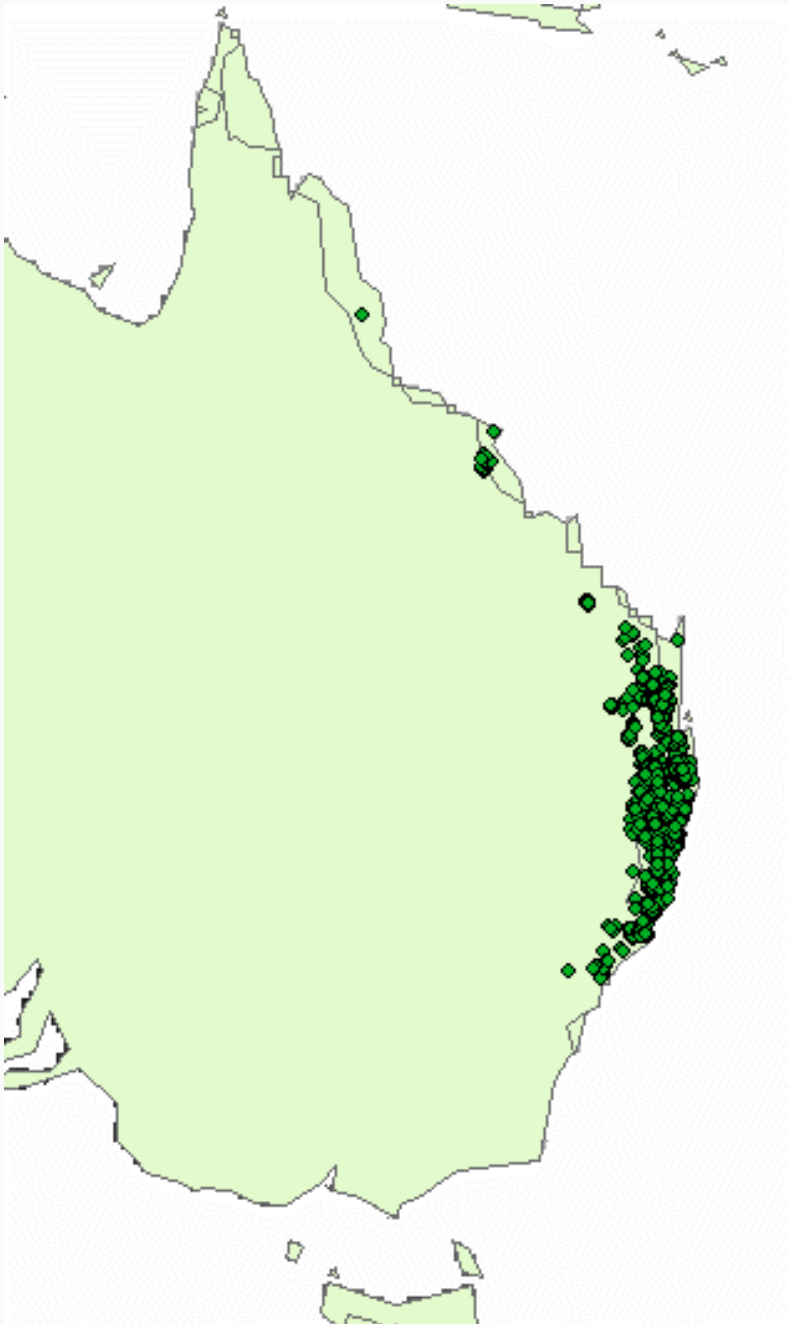
Prediction



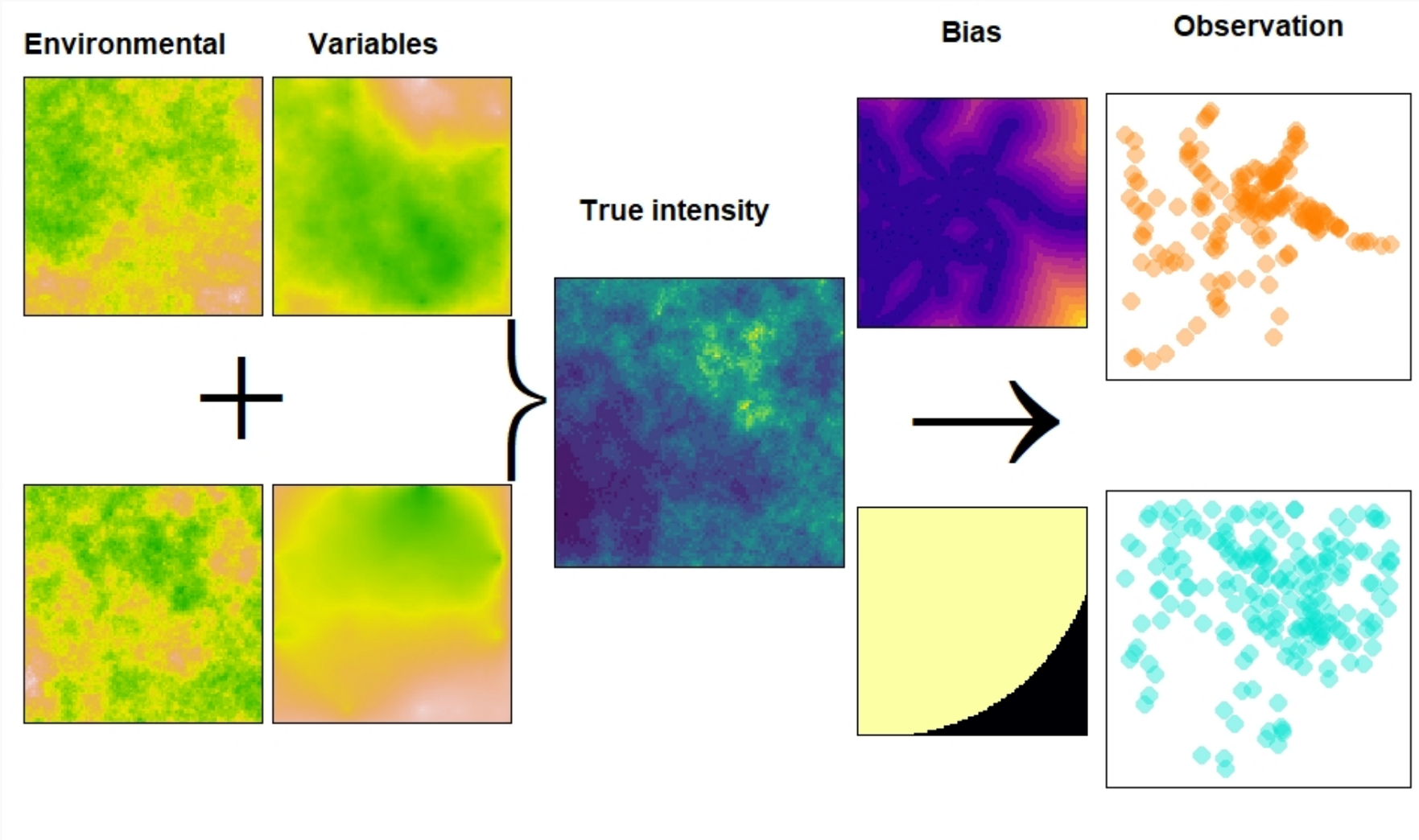
[1] Baddeley et al., 2015

Data types

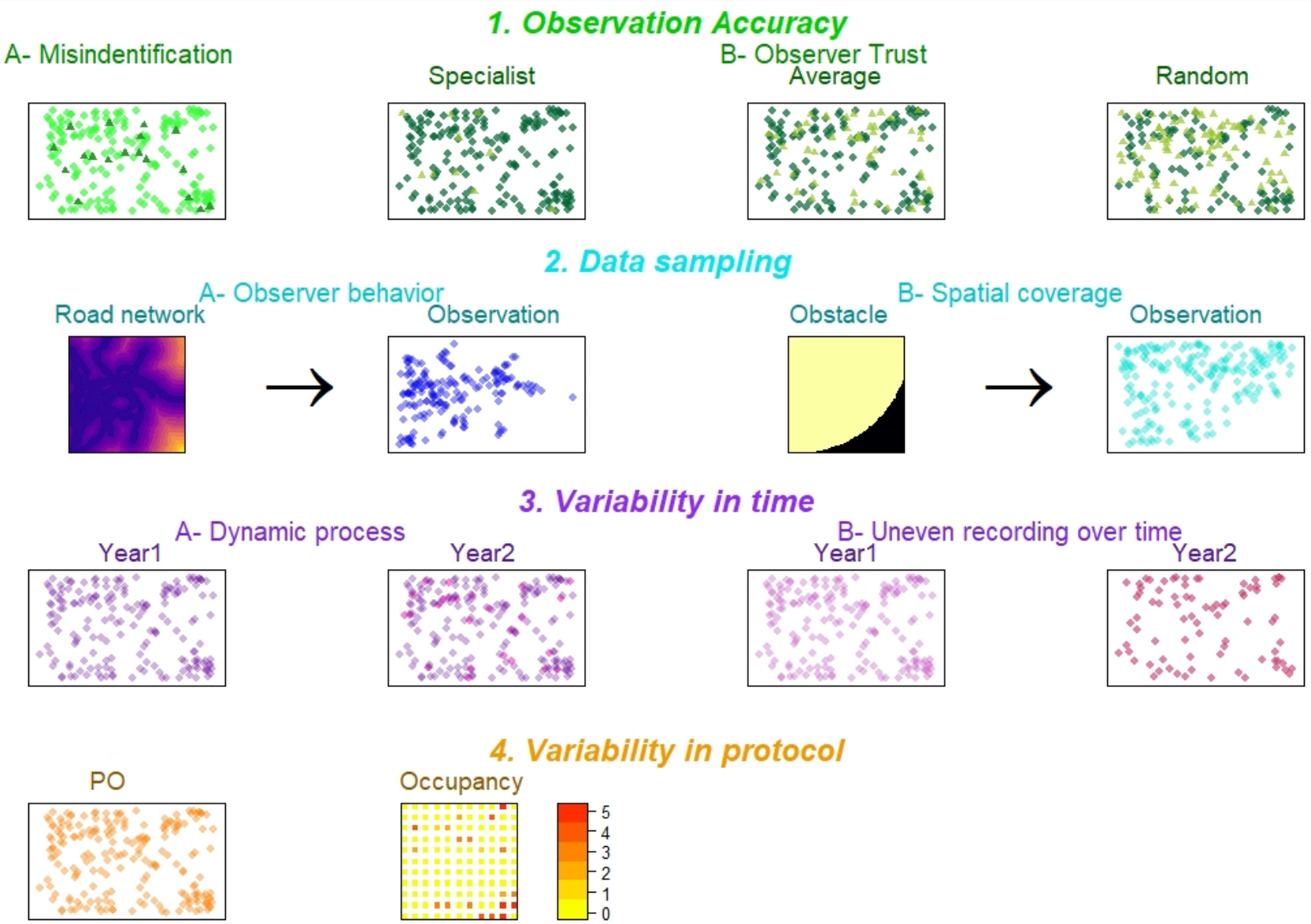
Presence only
Occupancy data
Presence - absence
Abundance



Data quality



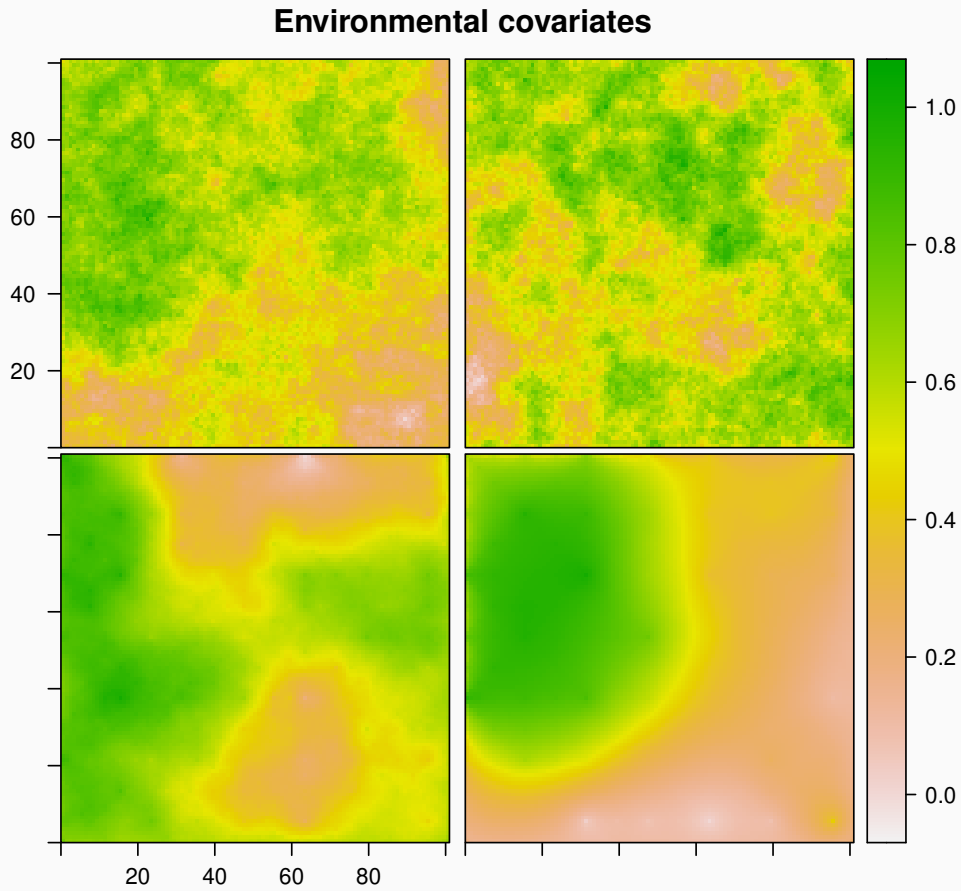
Data quality



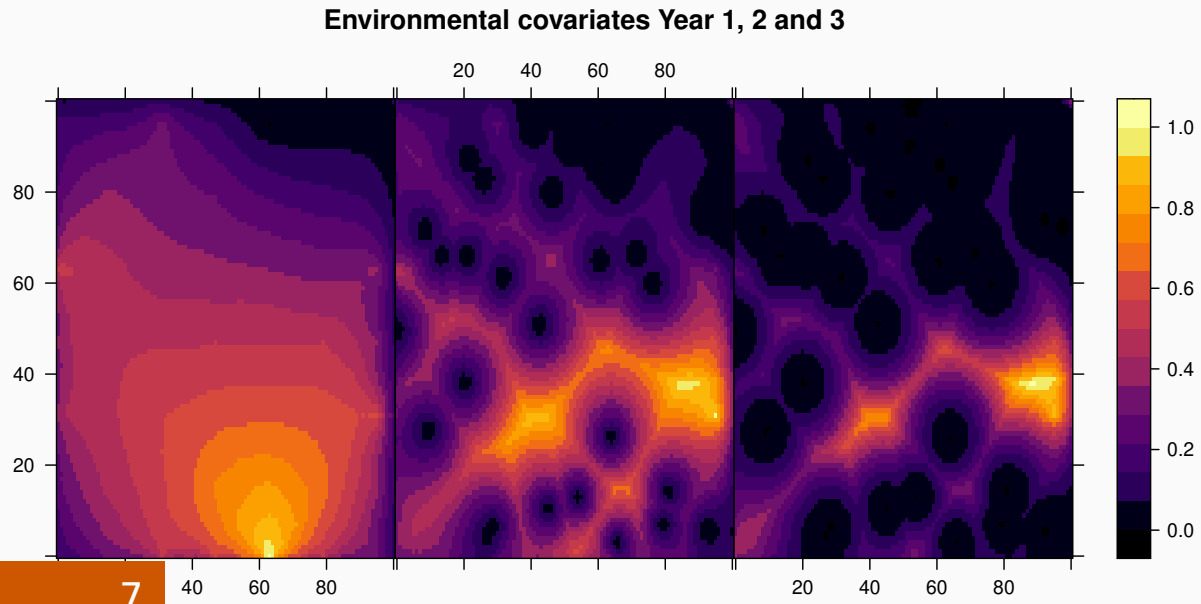
Environmental layers

```r
library(lattice)
library(spatstat)
library(raster)
library(NLMR)
library(viridis)
library(RColorBrewer)
library(scales)

source("Share Functions.R")

quad = expand.grid(seq(0, 100, 1), seq(0, 100, 1))
names(quad) = c("X", "Y")

win   = owin(xrange = c(-0.5, 100.5), yrange = c(-0.5, 100.5))
```

```r
set.seed(2)
v1 <- nlm_fbm(ncol = 101, nrow = 101, fract_dim = 0.8, user_seed = 1)

v2 <- nlm_gaussianfield(ncol = 101, nrow = 101, autocorr_range = 10,
                        mag_var = 80, nug = 1, user_seed = 1)

set.seed(5)
v3 <- nlm_mpd(ncol = 102, nrow = 102, roughness = 0.5)

v4 <- nlm_mpd(ncol = 102, nrow = 102, roughness = 0.2)
```

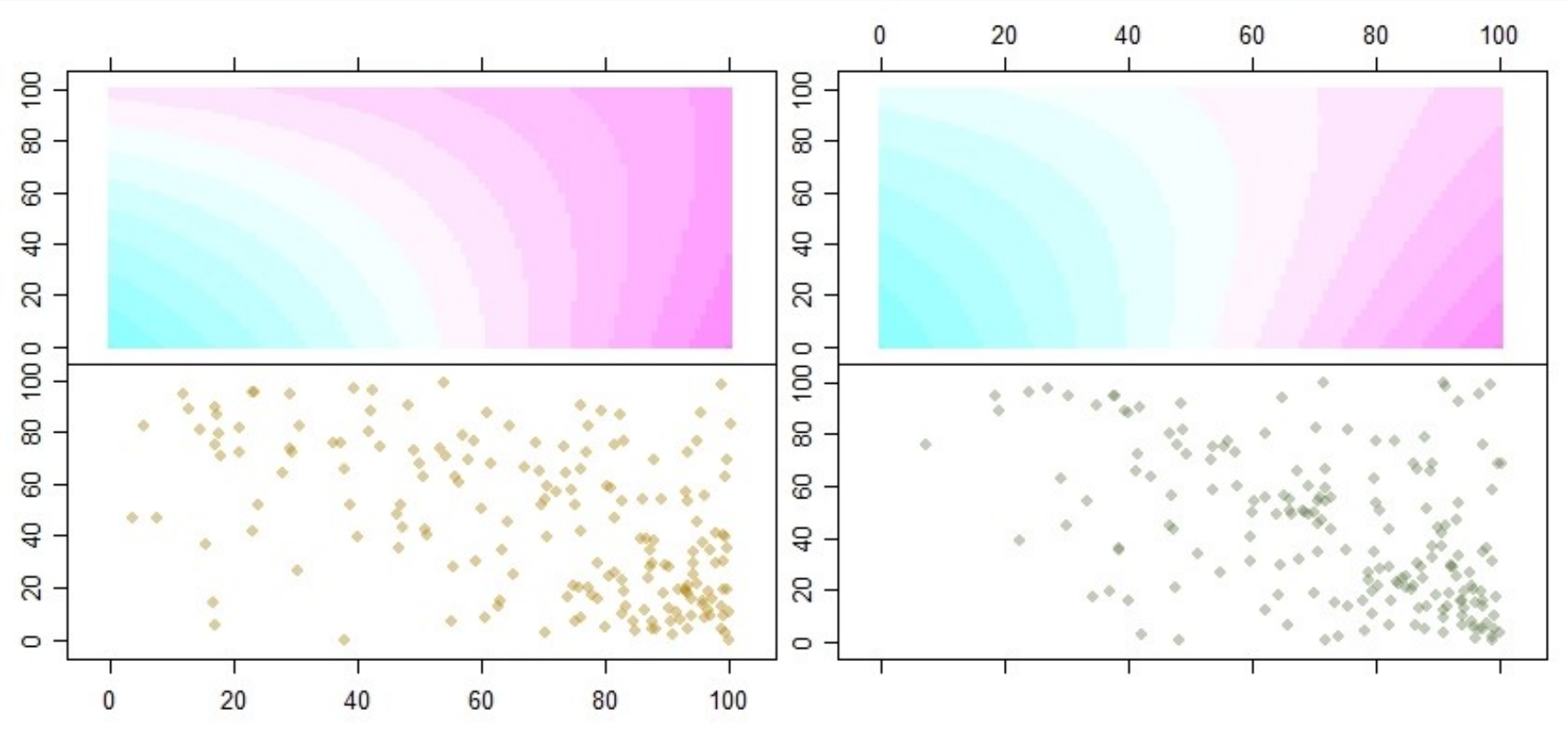


```r
## test with multiple centers randomly created
c1 = runif(30, min=0, max=100)
c2 = runif(30, min=0, max=100)
xy2 <- matrix(cbind(c1, c2),ncol=2, byrow=F)
p2 <- SpatialPoints(xy2)

dfp2 = distanceFromPoints(v4, p2)
distfP2 = as.data.frame(dfp2)

# Changes for time 2 or year 2
v4tre.Y2 = v4.df*distfP2$layer
v4tre.Y2.Lrsc = (v4tre.Y2$layer-min(v4tre.Y2$layer))/(max(v4tre.Y2$layer)-min(v4tre.Y2$
     layer))

# Changes for time 3 or year 3
v4tre.Y3 = v4tre.Y2*distfP2$layer*10
v4tre.Y3.Lrsc = (v4tre.Y3$layer-min(v4tre.Y3$layer))/(max(v4tre.Y3$layer)-min(v4tre.Y3$
     layer))
```



Disturbance and Data sampling

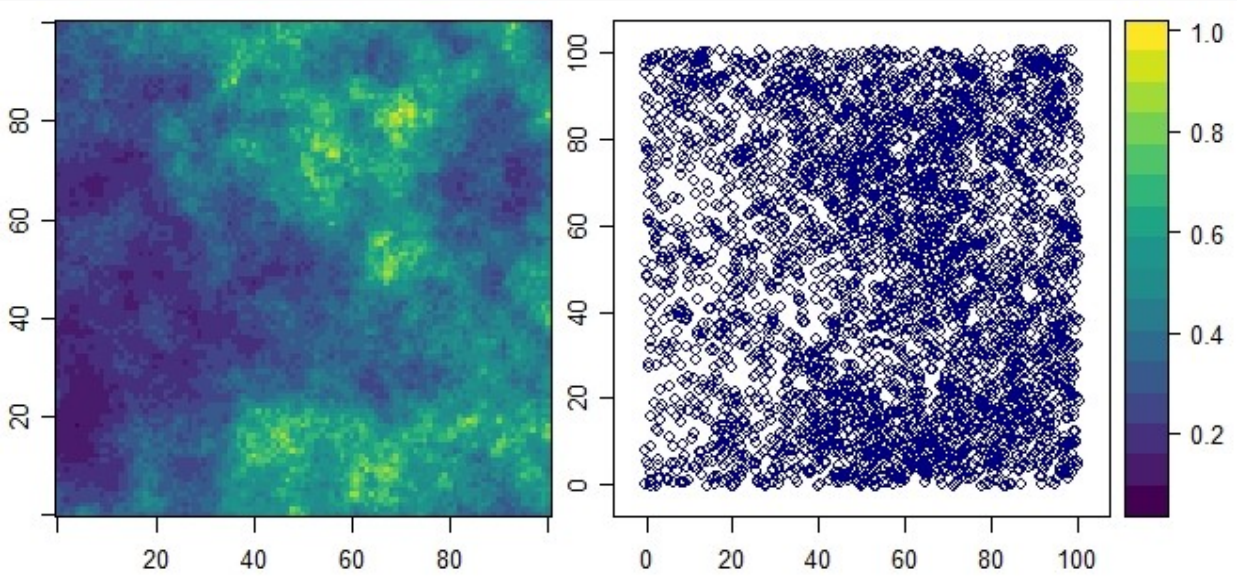
```
# generate X and Y axis
XY = expand.grid(seq(0, 100, 1), seq(0, 100, 1))
X = XY[,1]
Y = XY[,2]

# Generate a new covariate for year 1
vd = (X + 300)^2 + (Y + 160)^2 - 5*X*Y

vd = -1*scale(vd)
Lvd = levelplot(vd ~ X + Y)

# Generate a new covariate for year 2
vd.Y2 = (X + 300)^2 + (Y + 90)^2 - 5*X*Y
vd.Y2 = -1*scale(vd.Y2)

## Year 1
# environmental information at the quadrature points and species locations
quads4 = data.frame(X = quad$X, Y = quad$Y, V1 = vmat.df$v1,
V2 = vmat.df$v2, V3= vmat.df$v3, V4 = vmat.df$v4, Vd = vd)

## Year 2
# environmental information at the quadrature points and species locations
quads4.2 = data.frame(X = quad$X, Y = quad$Y, V1 = vmat.df$v1,
V2 = vmat.df$v2, V3= vmat.df$v3, V4 = vmat.df$v4, Vd = vd.Y2)
```


0
20
40
60
80
100

Aspect 1: Variation in recording due to the landscape

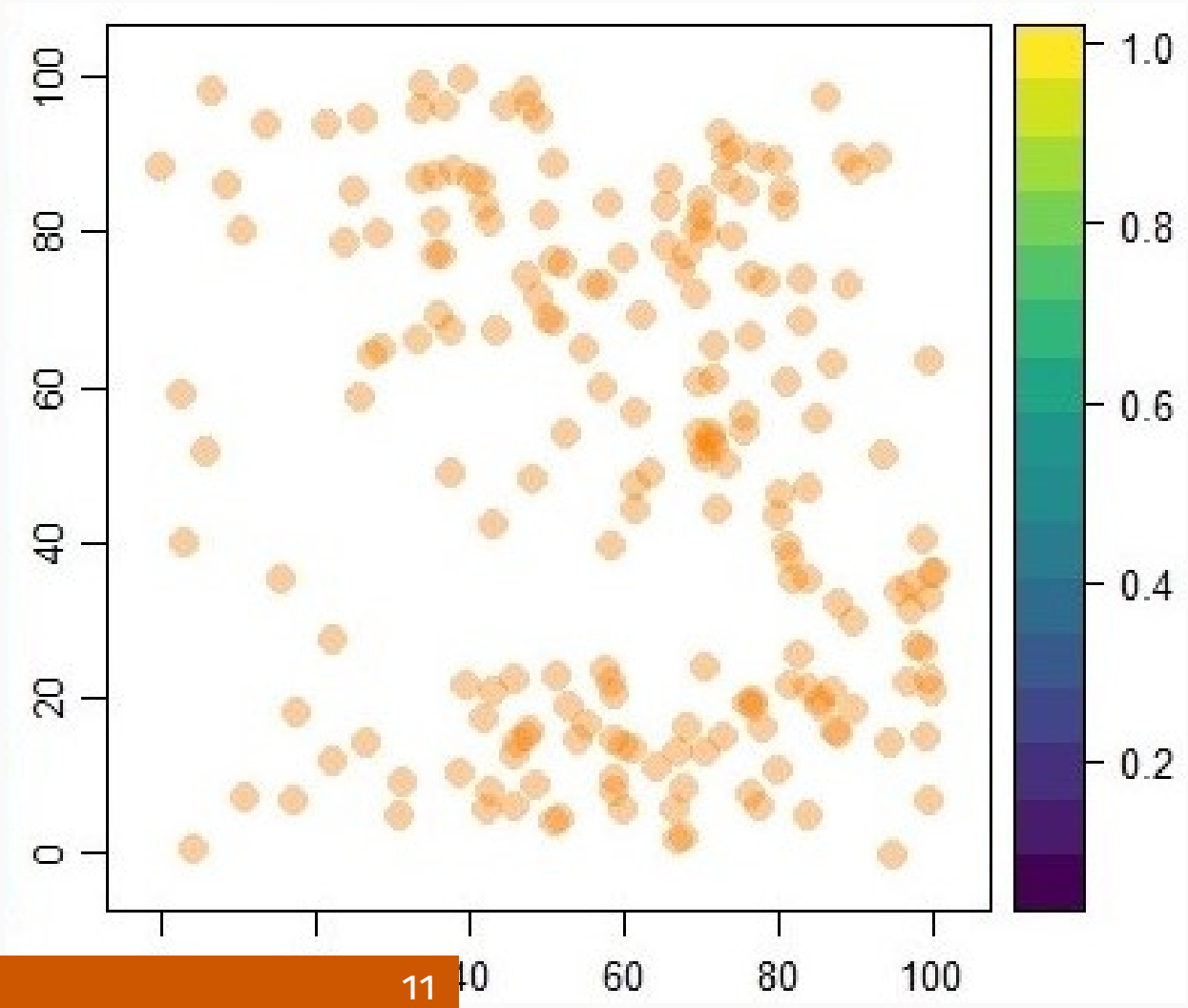
Generate PO data

```
sp1_coef = c(-1.5, 0.6, 2, -1, -0.8)
sp1_int = exp(env.mat %*% sp1_coef)
sp1_int_im = as.im(data.frame(x = quad$X, y = quad$Y, z = sp1_int))

# Generate points from the true intensity
sp1_sim = rpoispp(sp1_int_im)

# environmental information at the quadrature points and species locations
quads = data.frame(X = quad$X, Y = quad$Y, V1 = Datav1, V2 = Datav2,
                   V3= Datav3, V4 = Datav4)

sp_xy = data.frame(X = sp1_sim$x, Y = sp1_sim$y)
sp_env = newenv.var(sp.xy = sp_xy, env.grid = quads,
        env.scale = 0.5, coord = c("X", "Y"), file.name = NA)
```



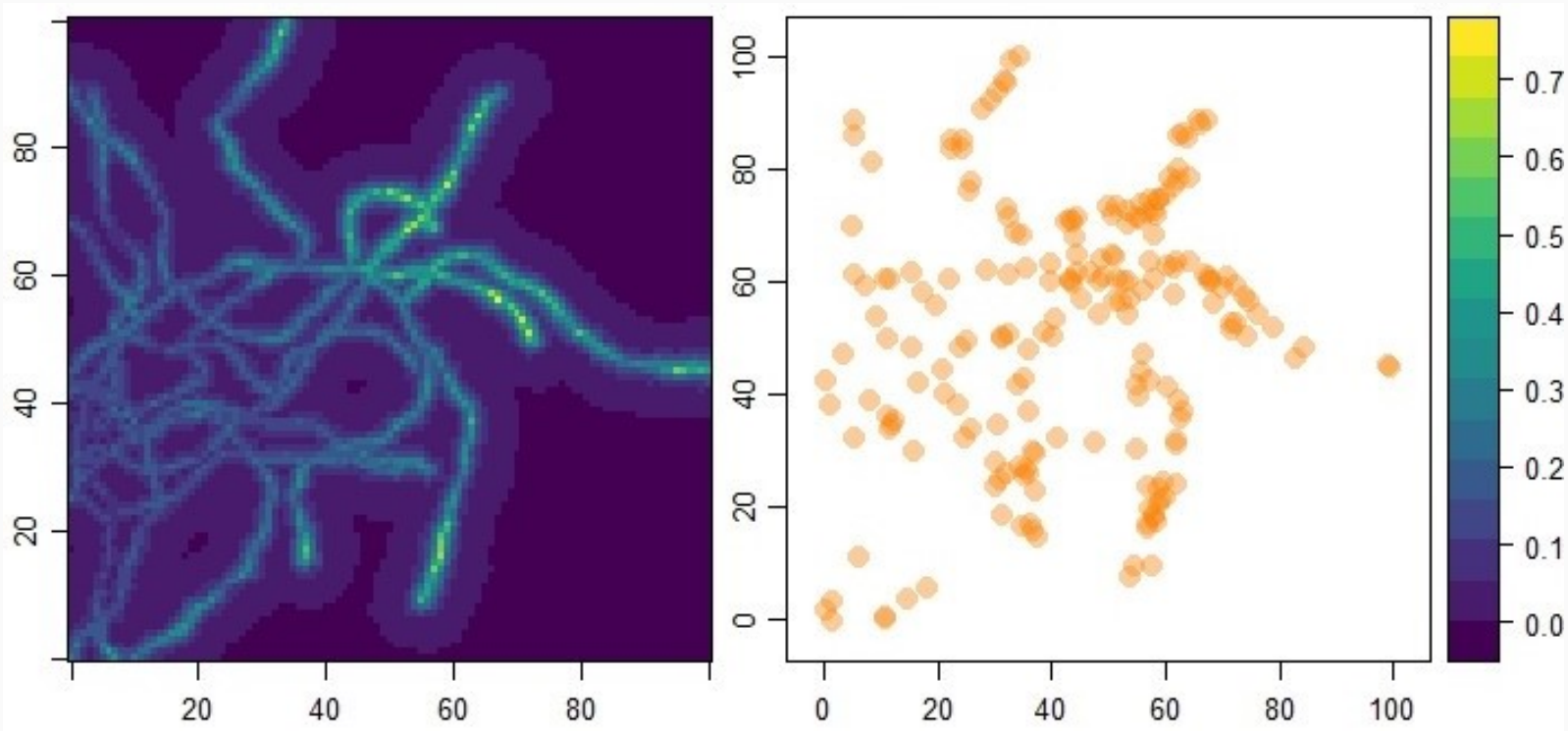
Sampling data

```
# point pattern sampling according to the covariates
po_X = data.frame(Intercept = 1, sp_env[,c(3:dim(quadsenv)[2])])
po_beta = c(intsamp, sp_coef[-1])
po_intensity = exp(as.matrix(po_X) %*% po_beta)
PO_rows = sample(1:sp_sim$n, npoint_samp, prob = po_intensity)
PO = sp_sim[PO_rows]
```



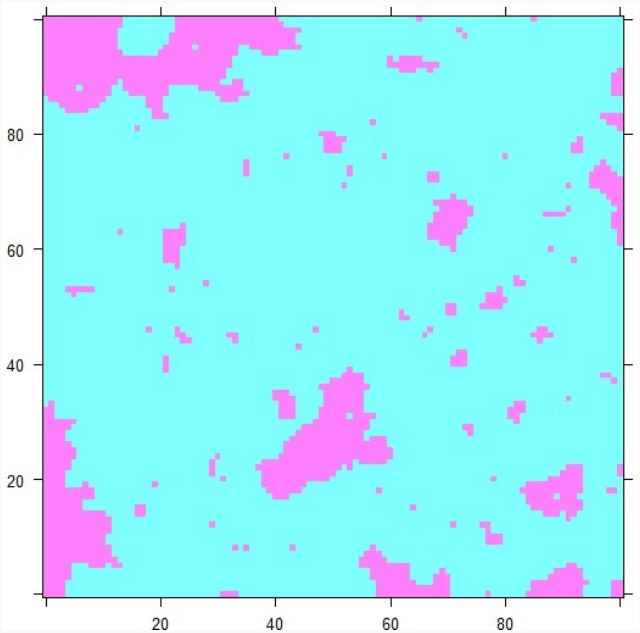
Road network

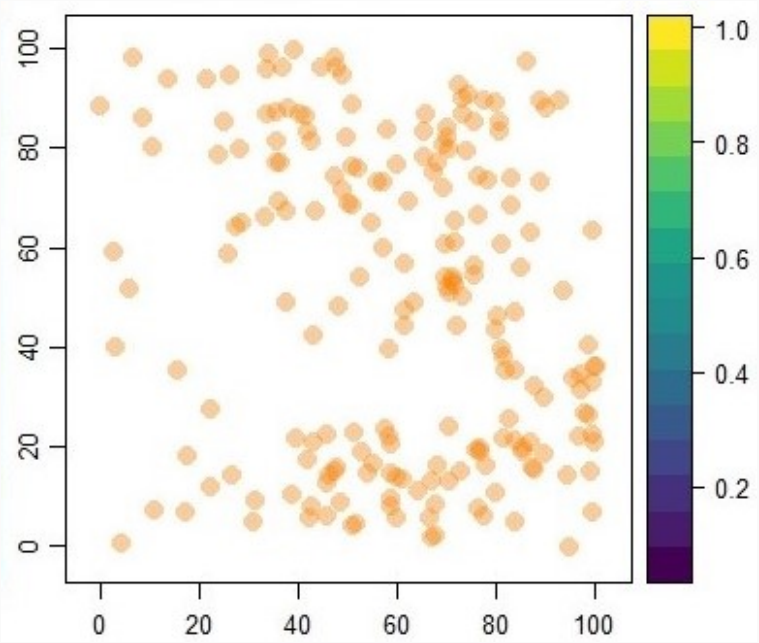
```
env.mat4 = as.matrix(data.frame(quad$X, quad$Y,
Datav1, Datav2, Datav3, Datav4, d_rd))
colnames(env.mat4)=c("X", "Y", "v1", "v2", "v3", "v4", "d_rd")
sp_coef4 = c(-1.5, 0.6, 2, -1, -0.8, -0.8)

simpo(quadsenv = as.data.frame(env.mat4), sp_coef=sp_coef4,
 intsamp=-1.5, npoint_samp=200, samp=TRUE, plot=TRUE, compPlot = TRUE)
```



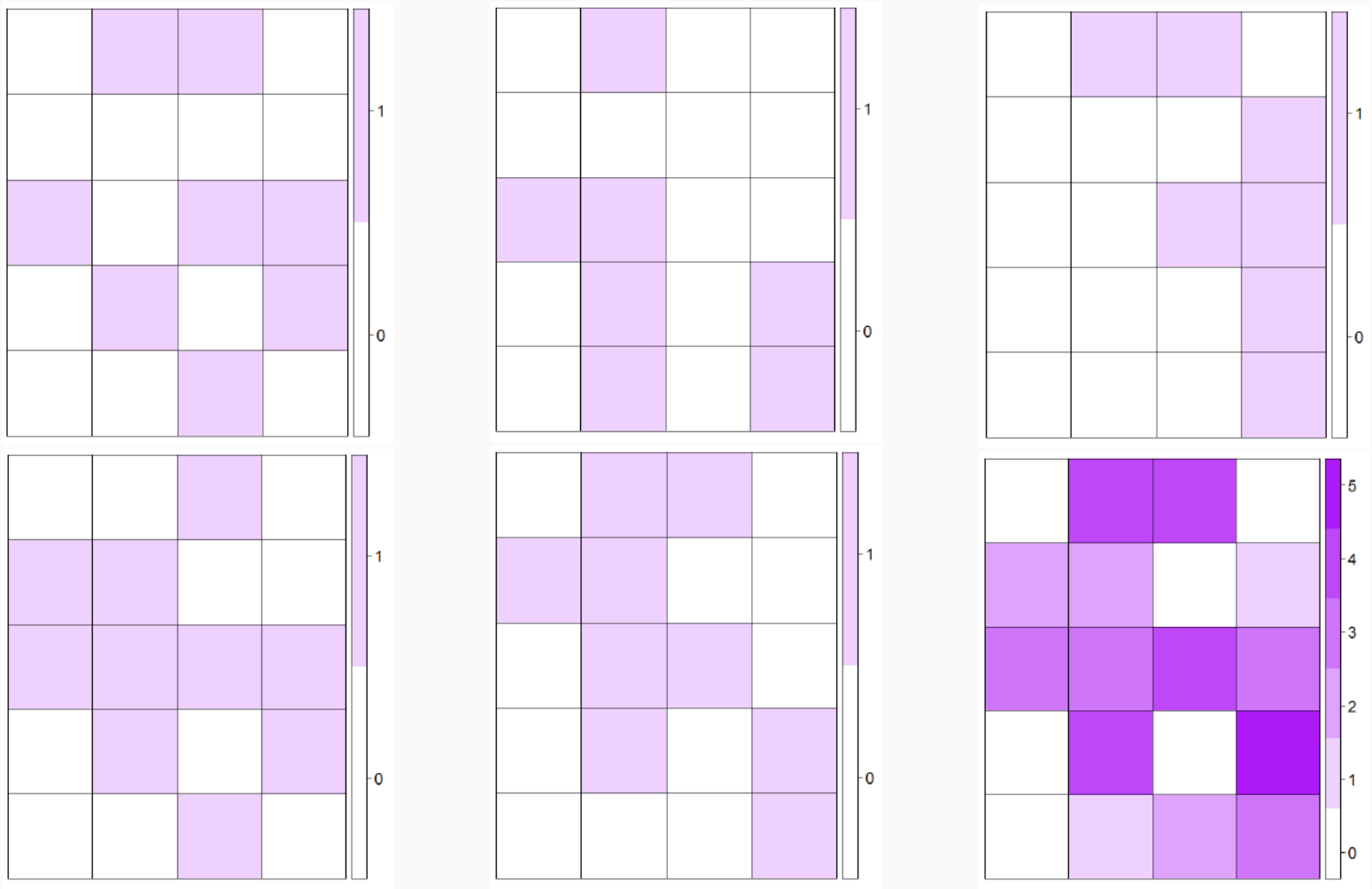
Vegetation or habitat facilitation

```
set.seed(13)
v7 = nlm_randomcluster(ncol = 101, nrow = 101, p = 0.6, ai = c(0.25, 0.25, 0.5))
Datav7 = as.data.frame(v7)
v8 = rep(0, length(v7)) # categorical for PO bias
v8[Datav7 >= quantile(v7, 0.75)] = 1

env.mat2 = as.matrix(data.frame(quad$X, quad$Y, Datav1, Datav2, Datav3, Datav4, v8))
colnames(env.mat2)=c("X", "Y", "v1", "v2", "v3", "v4", "v8")

sp_coef2 = c(-1.5, 0.6, 2, -1, -0.8, 2)

simpo(quadsenv = as.data.frame(env.mat2), sp_coef=sp_coef2, intsamp=-1.5,
        npoint_samp=150, samp=TRUE, plot=TRUE, compPlot = TRUE)
```



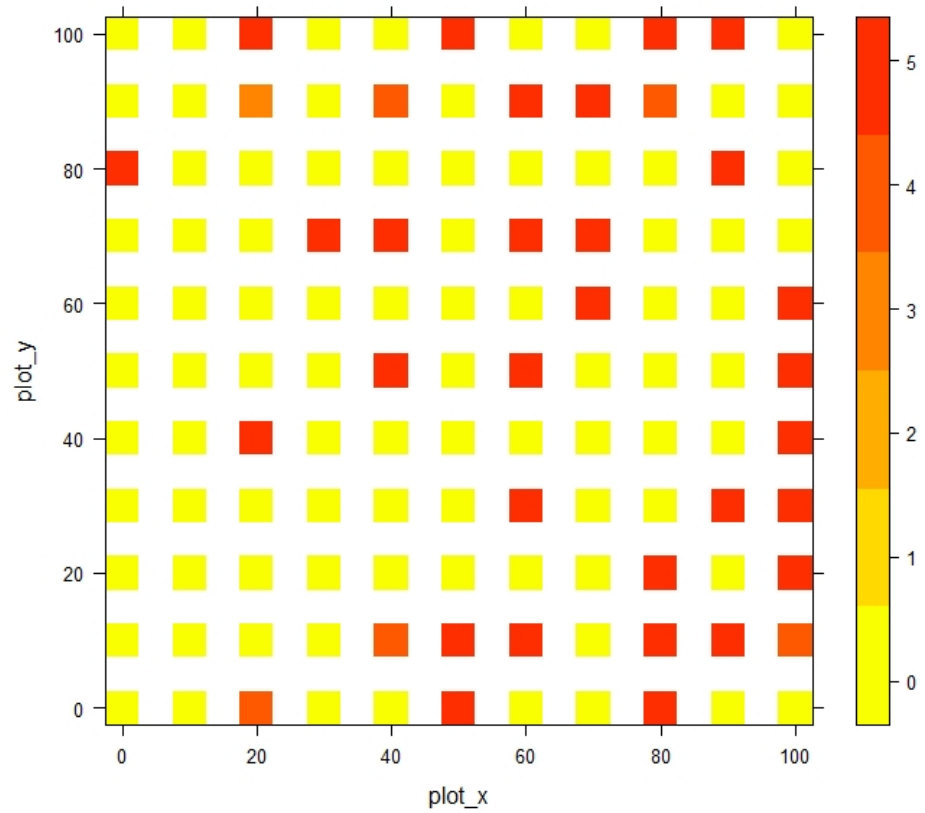


Occupancy data [2]



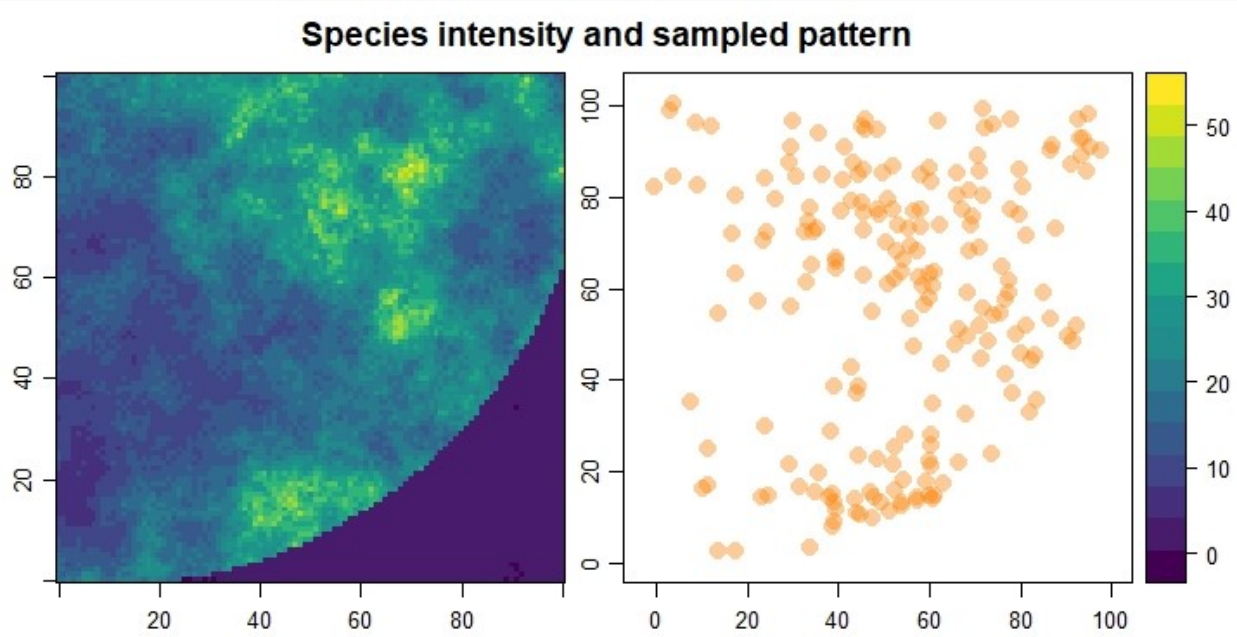
[2] Renner, I. W., Louvrier, J., & Gimenez, O. (2019). Combining multiple data sources in species distribution models while accounting for spatial dependence and overfitting with combined penalized likelihood maximization. Methods in Ecology and Evolution, 10(12), 2118-2128.

```
simocc(sp_coef=sp_coef2, quadsenv=as.data.frame(env.mat2),
        upres=0.2, vardetect=7, n_visits=5, occ_plot=TRUE)
```

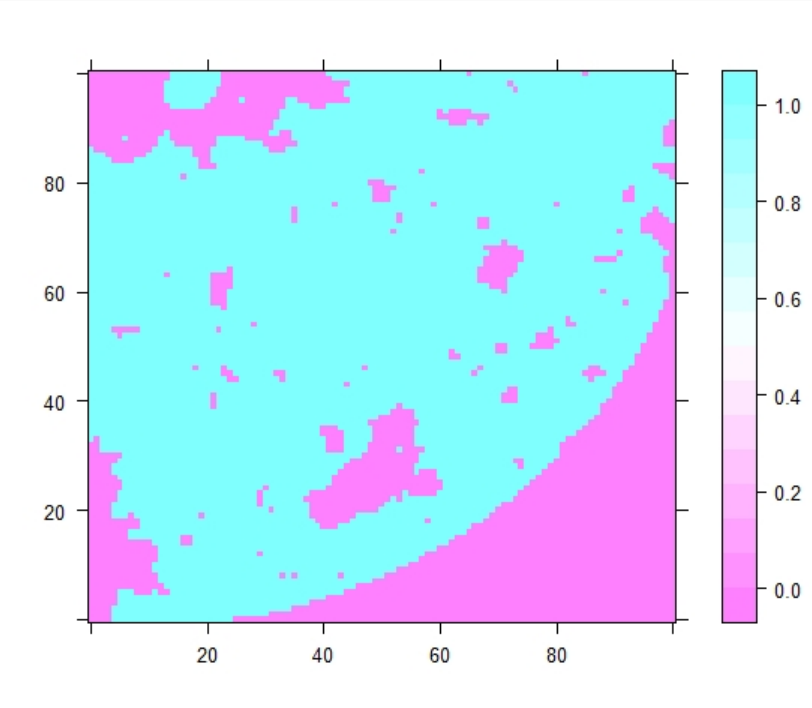


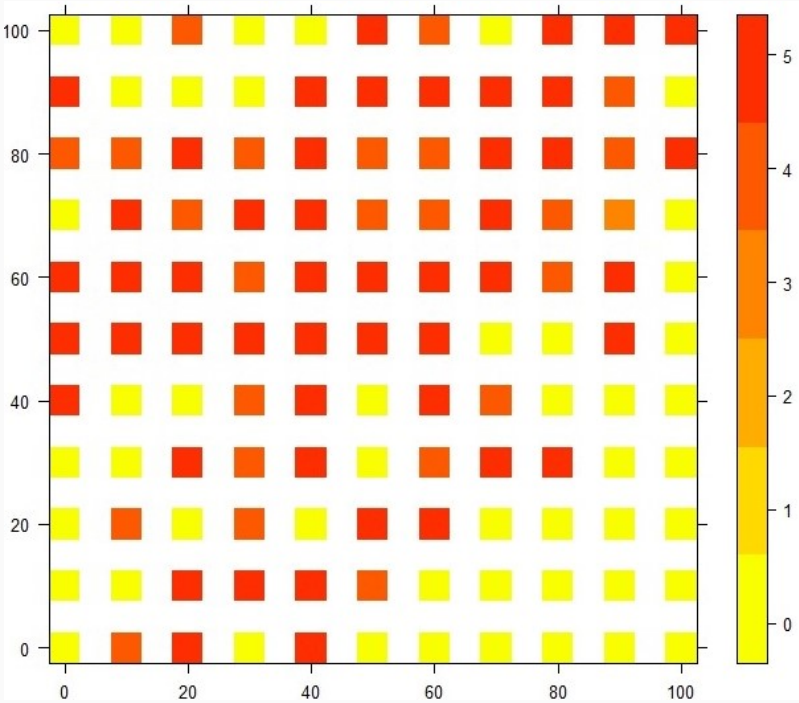
Uneven spatial coverage

```
# Creating a categorical variable
v5 = (X - 10)^2 + (Y - 90)^2 - 0.1*X*Y
v6 = rep(0, length(v5))
v6[v5 <= quantile(v5, 0.85)] = 1

env.mat3 = as.matrix(data.frame(quad$X, quad$Y,
Datav1, Datav2, Datav3, Datav4, v6))
colnames(env.mat3)=c("X", "Y", "v1", "v2", "v3", "v4", "v6")
sp_coef3 = c(-1.5, 0.6, 2, -1, -0.8, 4)

simpo(quadsenv = as.data.frame(env.mat3), sp_coef=sp_coef3,
        intsamp=-1.5, npoint_samp=200, samp=TRUE, plot=TRUE, compPlot = TRUE)
```



```r
v6.rast = rast1 <- raster(ncol=101, nrow=101, xmn=0, ymn=0, xmx=100, ymx=100)
values(v6.rast) = v6

# No detection in a certain area from v6
v8.uneven = v8
v8.uneven[values(v6.rast)==0] = 0

env.mat4 = as.matrix(data.frame(quad$X, quad$Y, Datav1, Datav2, Datav3,
         Datav4, v8.uneven))
colnames(env.mat4)=c("X", "Y", "v1", "v2", "v3", "v4", "v8")

Test7 = simocc(sp_coef=sp_coef3, quadsenv=as.data.frame(env.mat4),
        upres=0.2, vardetect=7, n_visits=5, occ_plot=TRUE)
```

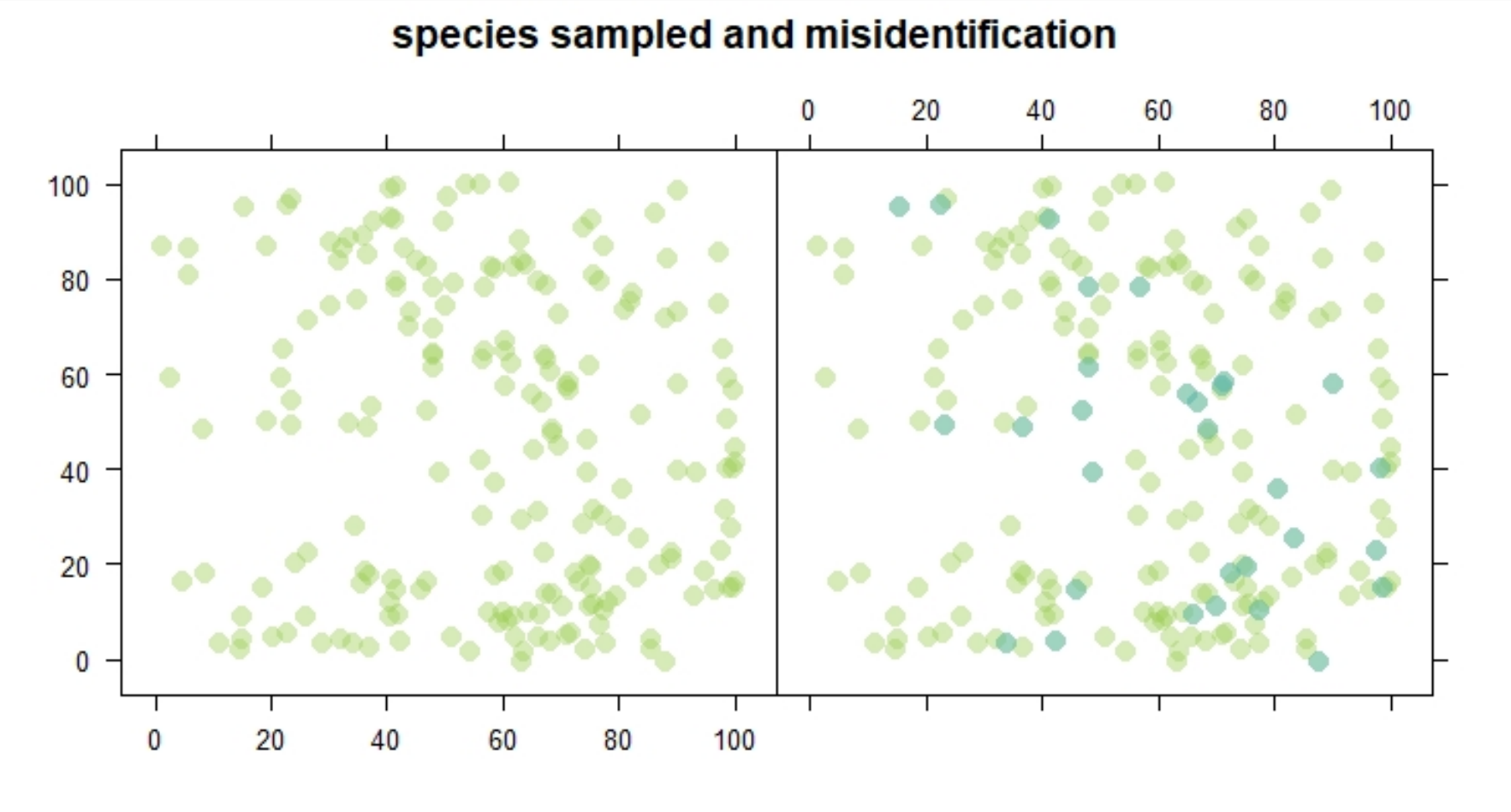




Aspect 2 : Observer recording and accuracy

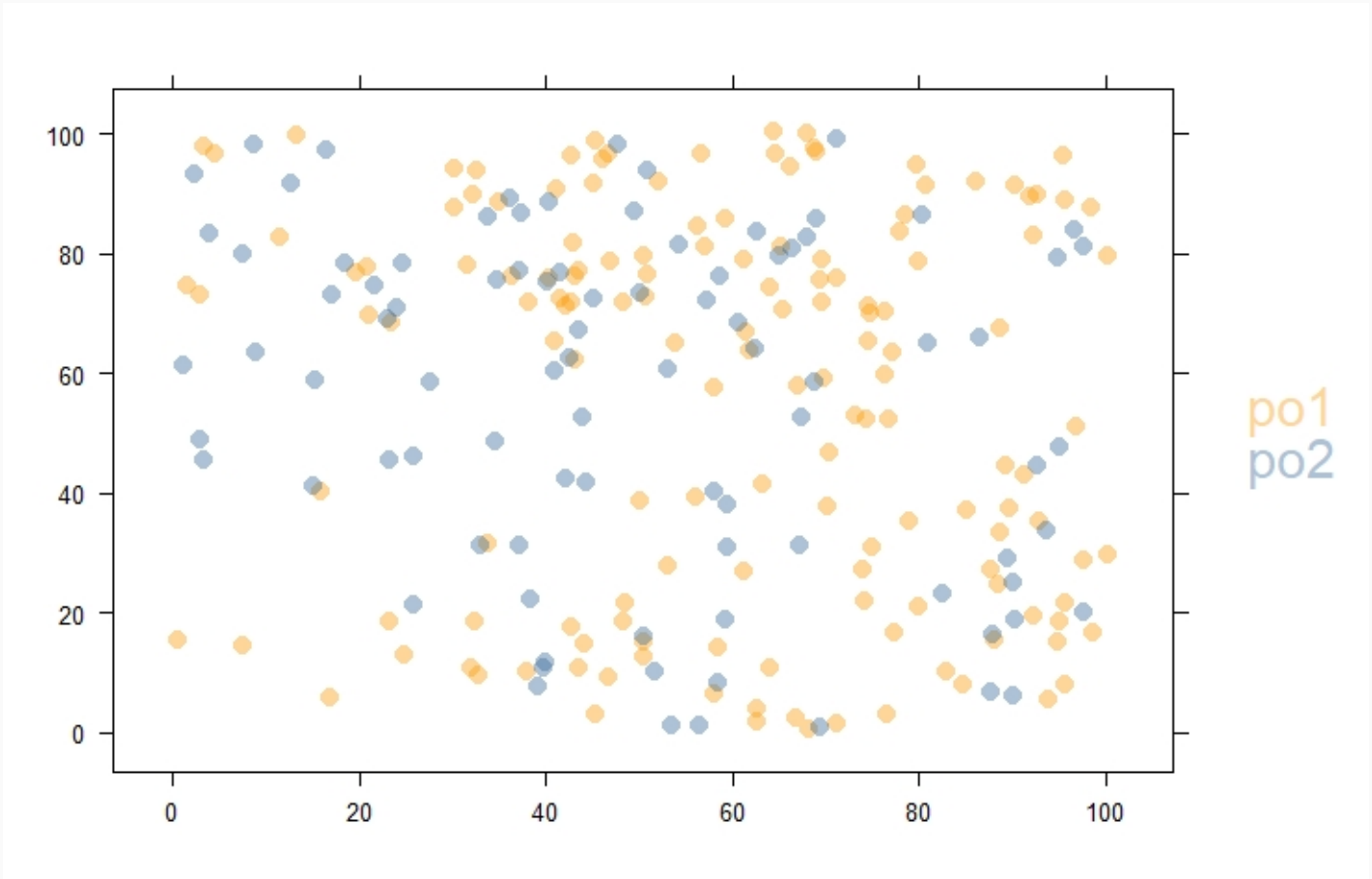
Misidentification

```
PO_rows = sample(1:sp_sim$n, npoint_samp, prob = po_intensity)
PO = sp_sim[PO_rows]

misidpoints(sp_coef=sp_coef, quadsenv=as.data.frame(env.mat),
        intsamp=-1.5, npoint_samp=200, plot=TRUE, perc_mis=0.15,
        compPlot=TRUE, Misid = TRUE)
```



Different species with different habitat preferences

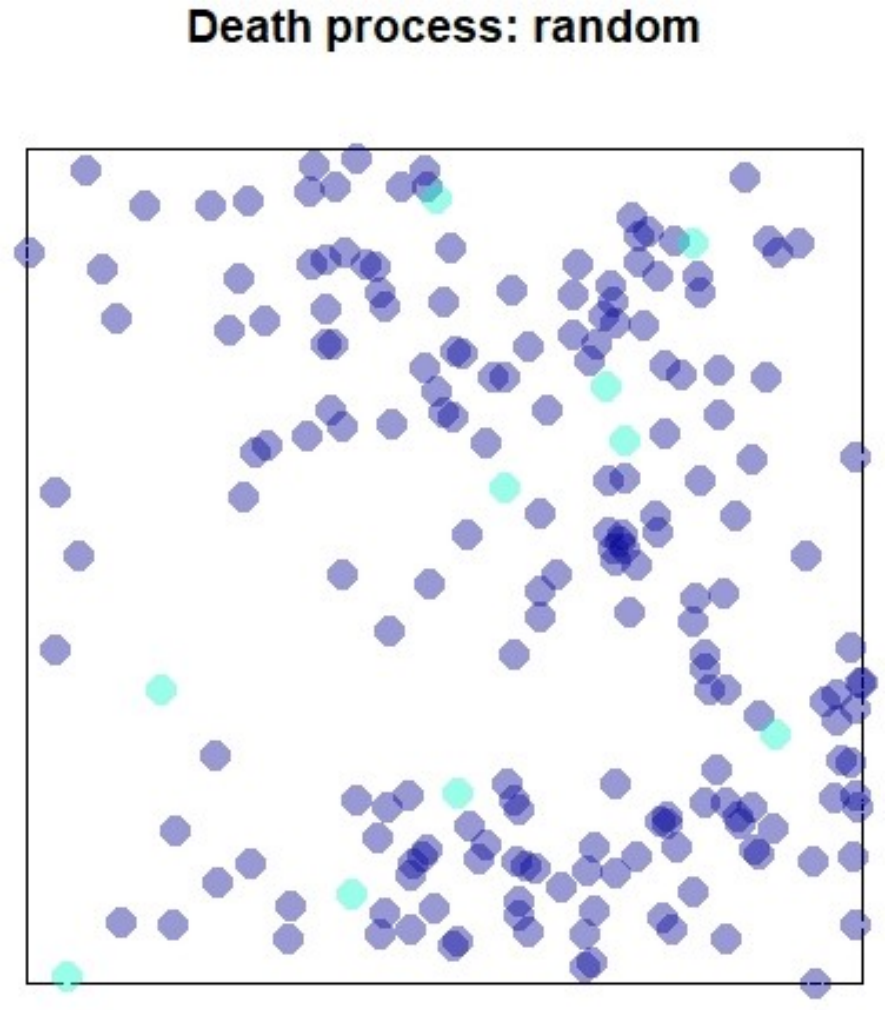
```
sp1_coef = c(-1.5, 0.6, 2, -1, -0.8)
spbis_coef = c(-1.5, 0.3, 1.8, -0.7, 0.4)

multiplepo(sp_coefmult=list(sp_coef, spbis_coef), quadsenv=as.data.frame(env.mat),
        intsampmult=list(-1.5, -1.5), npoint_sampmult=list(150, 90),
        plot=TRUE, compPlot=TRUE)
```



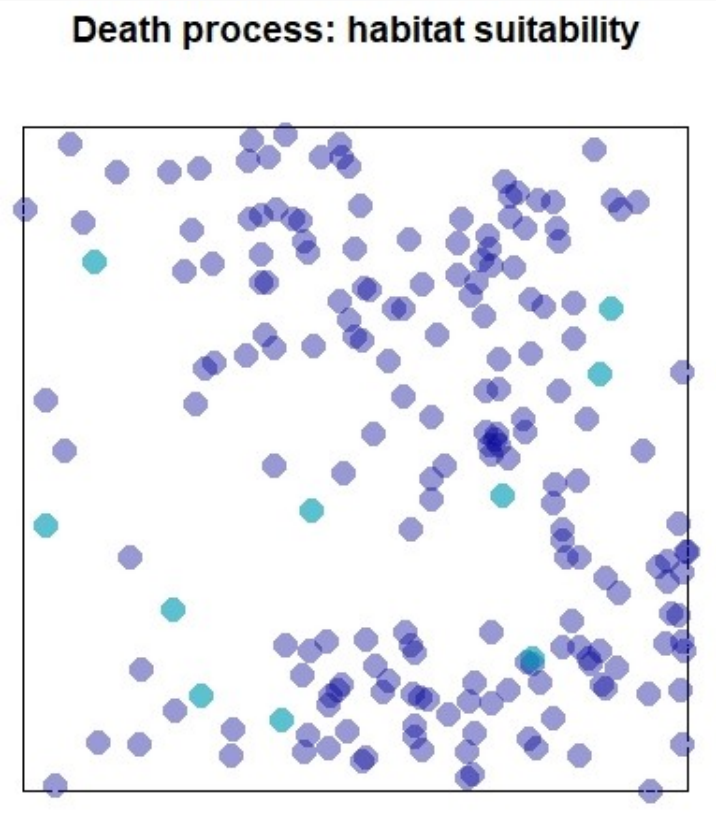
Aspect 3: Data-dynamic

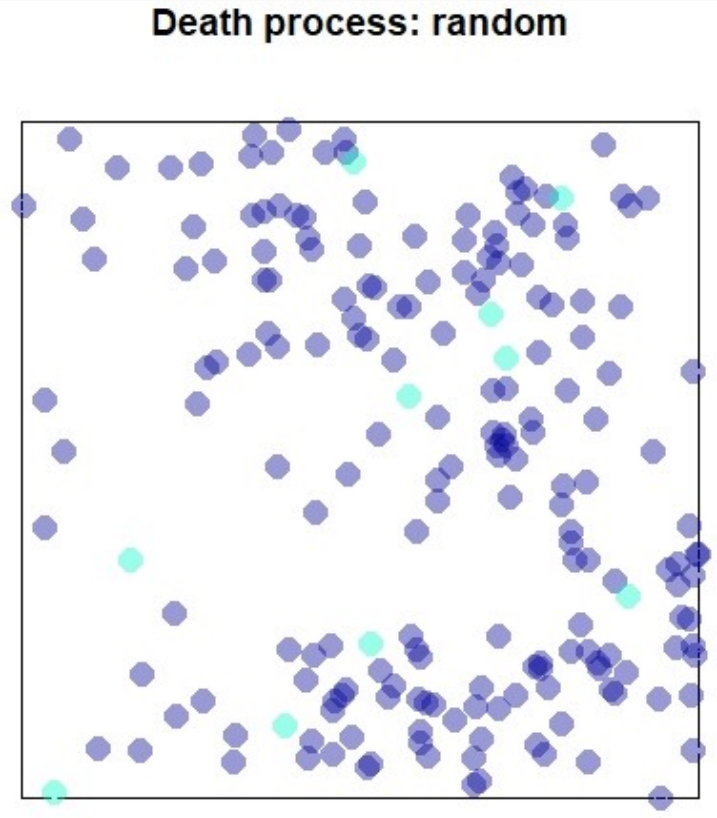
Death

```
Year1 = PO3
prob_death = 0.05
died = sample(1:PO3$n, round(rnorm(1, prob_death, 0.01)*PO3$n))
PO3_yr2 = PO3[-died]
```



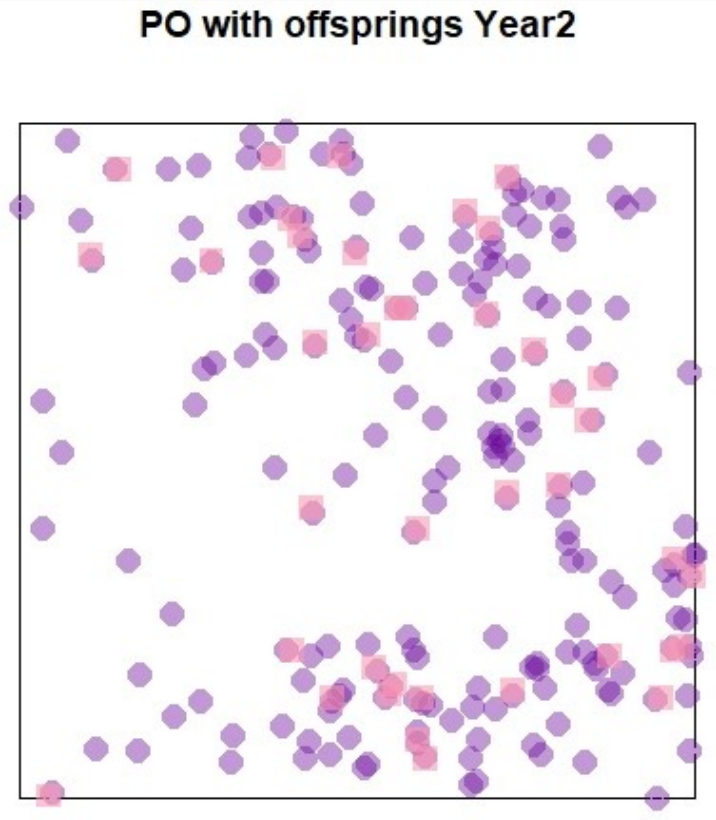
```
deathpoints(simpoRes=test2, prob_death=0.05, sd=0.01)

deathpoints(simpoRes=test2, prob_death=0.05, sd=0.01, random = TRUE)
```

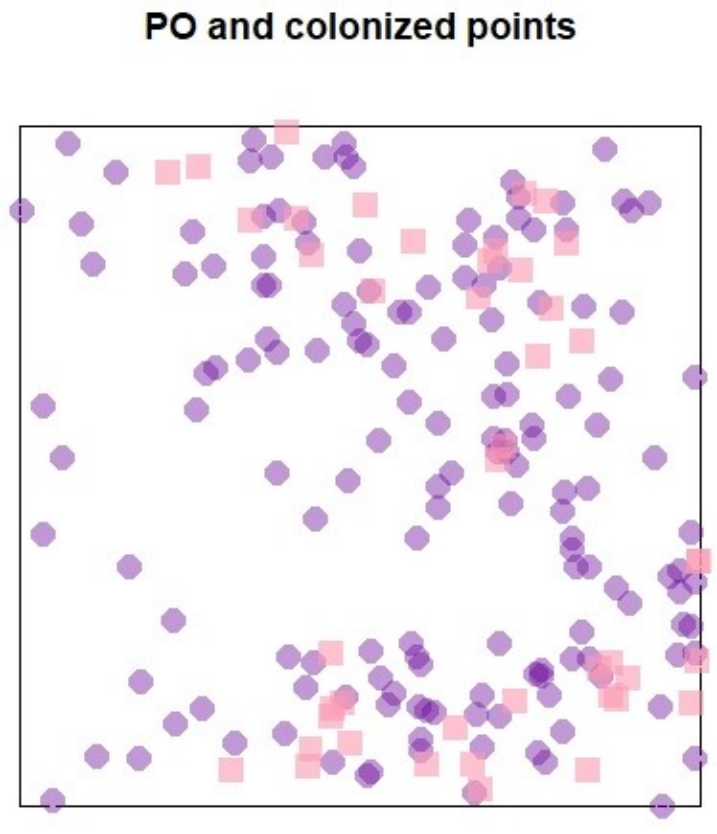




Birth and colonization

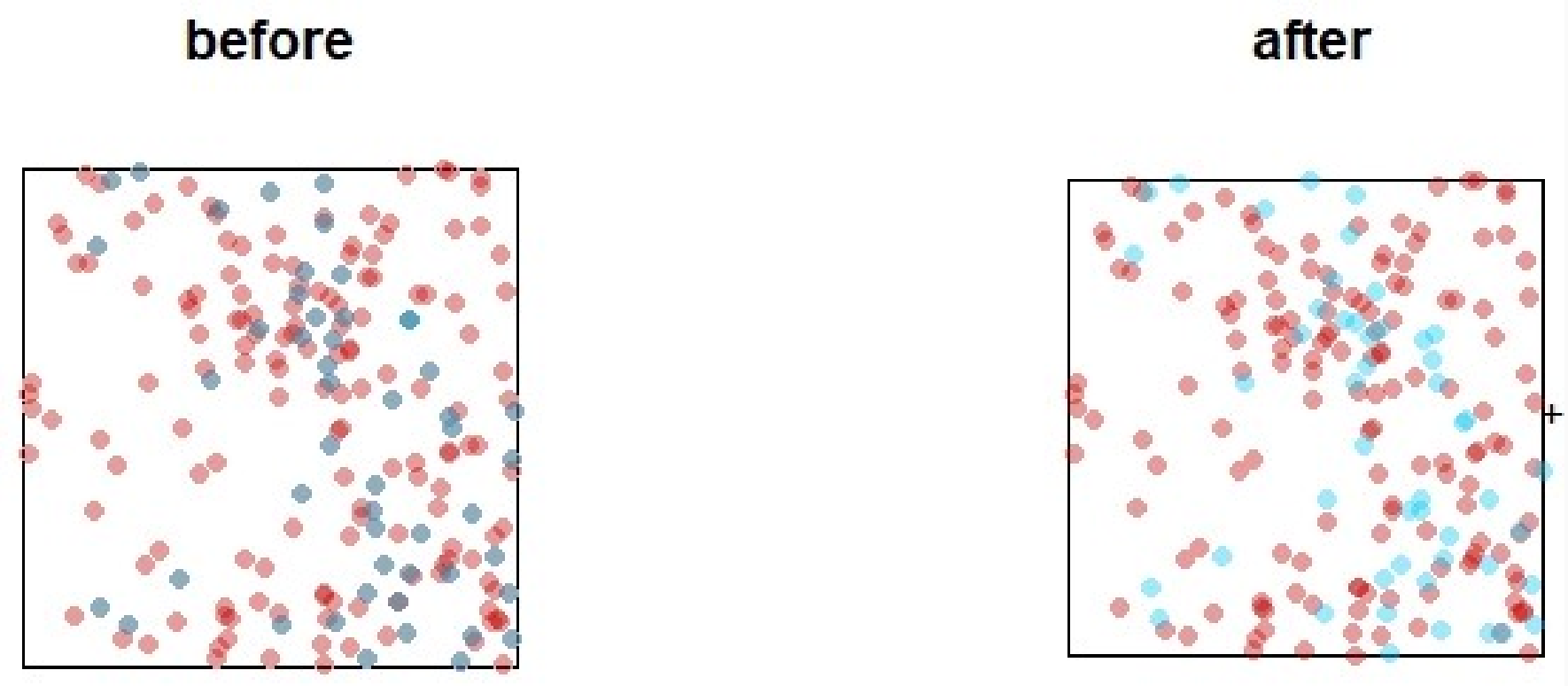
```
Test10 = birthpoints(simpoRes=test2, prop_offsp=0.2, radius=1, birth=TRUE)

Test11 = birthpoints(simpoRes=test2, prop_offsp=0.2, birth=FALSE)
```





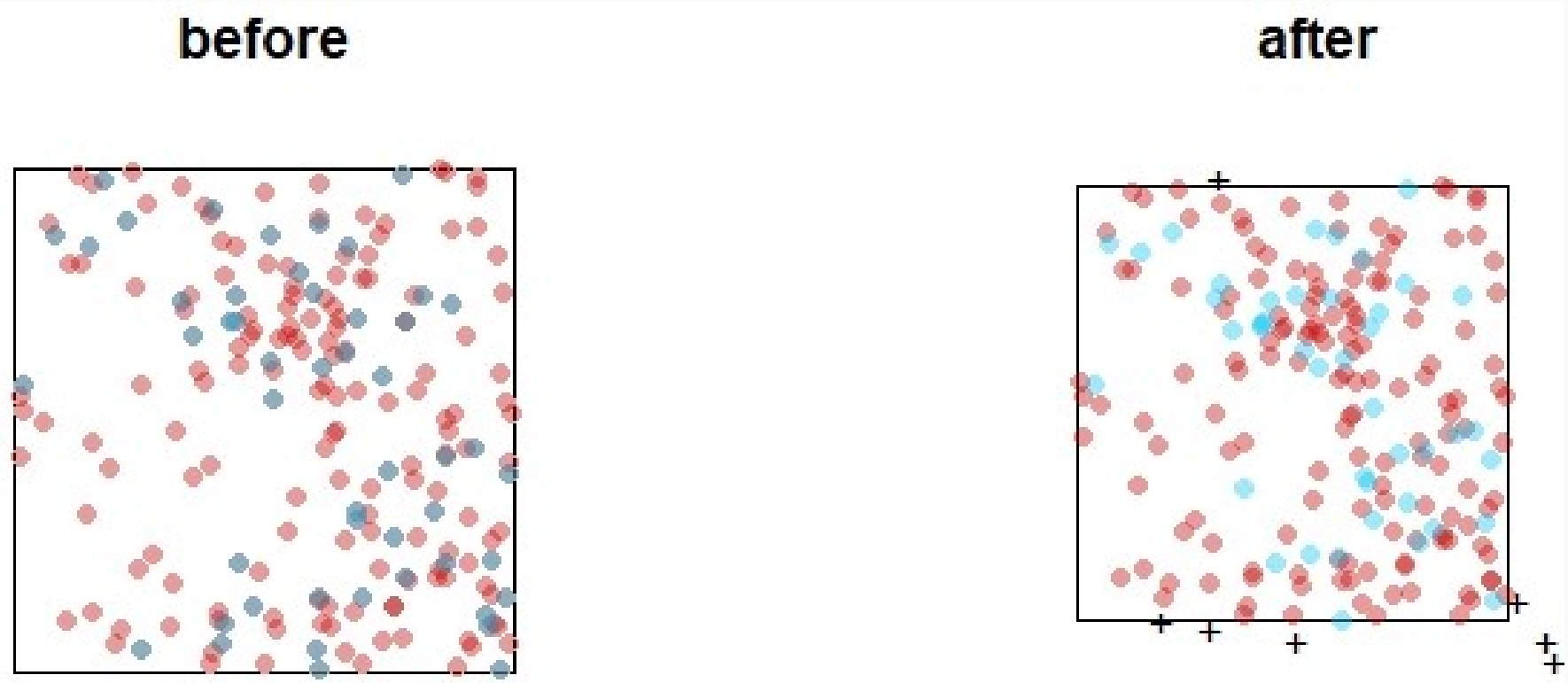
Random movement

```
Test12 = movepoints(simpoRes=test2, mov_type = "random", mean=0, sd=2, Allpt = TRUE,
        p_scale=NULL, n_var=NULL, p_move=NULL)

Test12b = movepoints(simpoRes=test2, mov_type = "random", mean=0, sd=2, Allpt = FALSE,
        p_move = 0.25, p_scale=NULL, n_var=NULL)
```



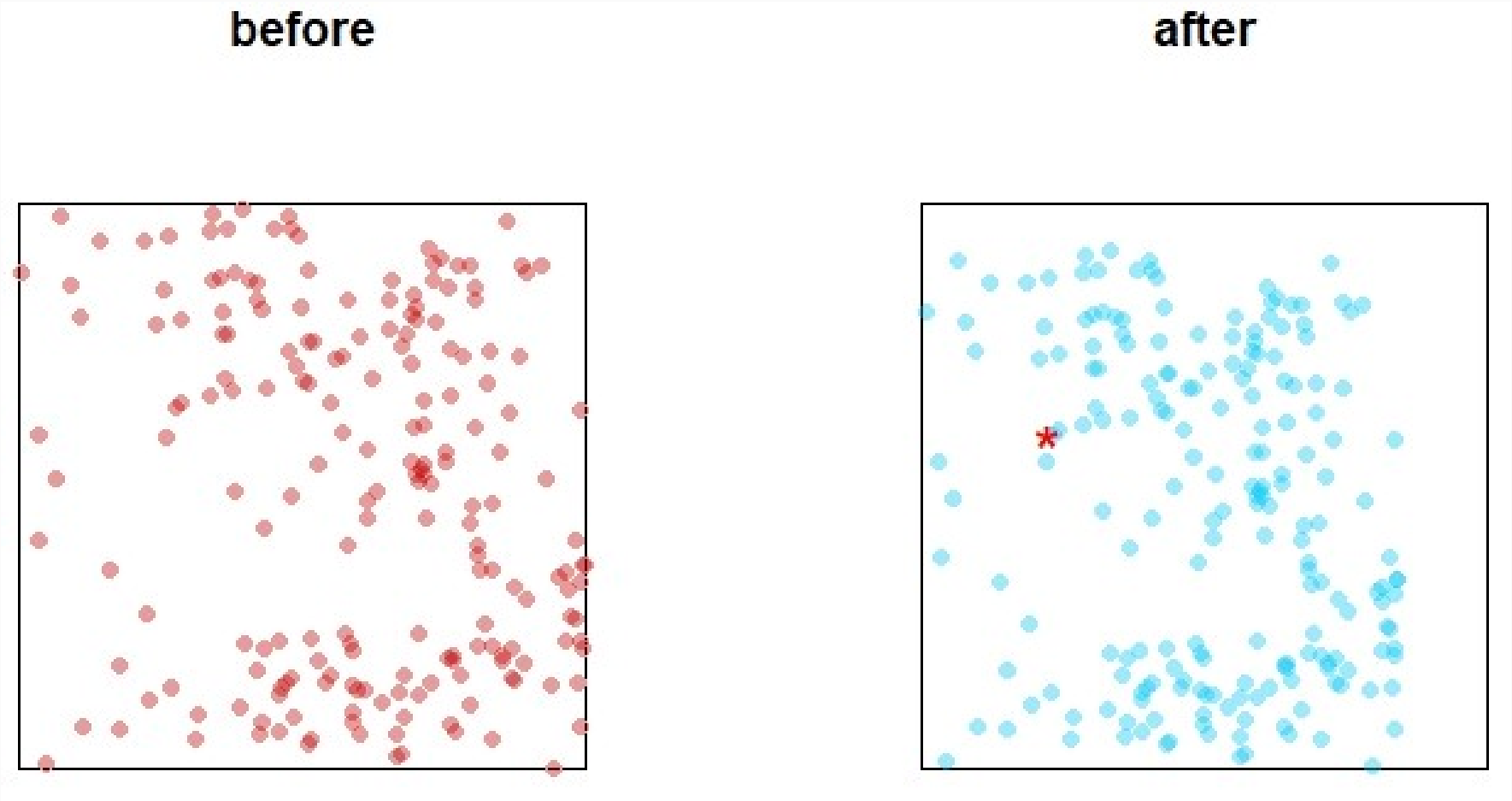
Movement according to habitat suitability

```
Test13 = movepoints(simpoRes=test2, mov_type = "intensity", mean=0, sd=2, Allpt = TRUE,
        p_scale=NULL, n_var=NULL, p_move=NULL)

Test13b = movepoints(simpoRes=test2, mov_type = "intensity", mean=0, sd=2, Allpt = FALSE,
        p_move = 0.25, p_scale=NULL, n_var=NULL)
```



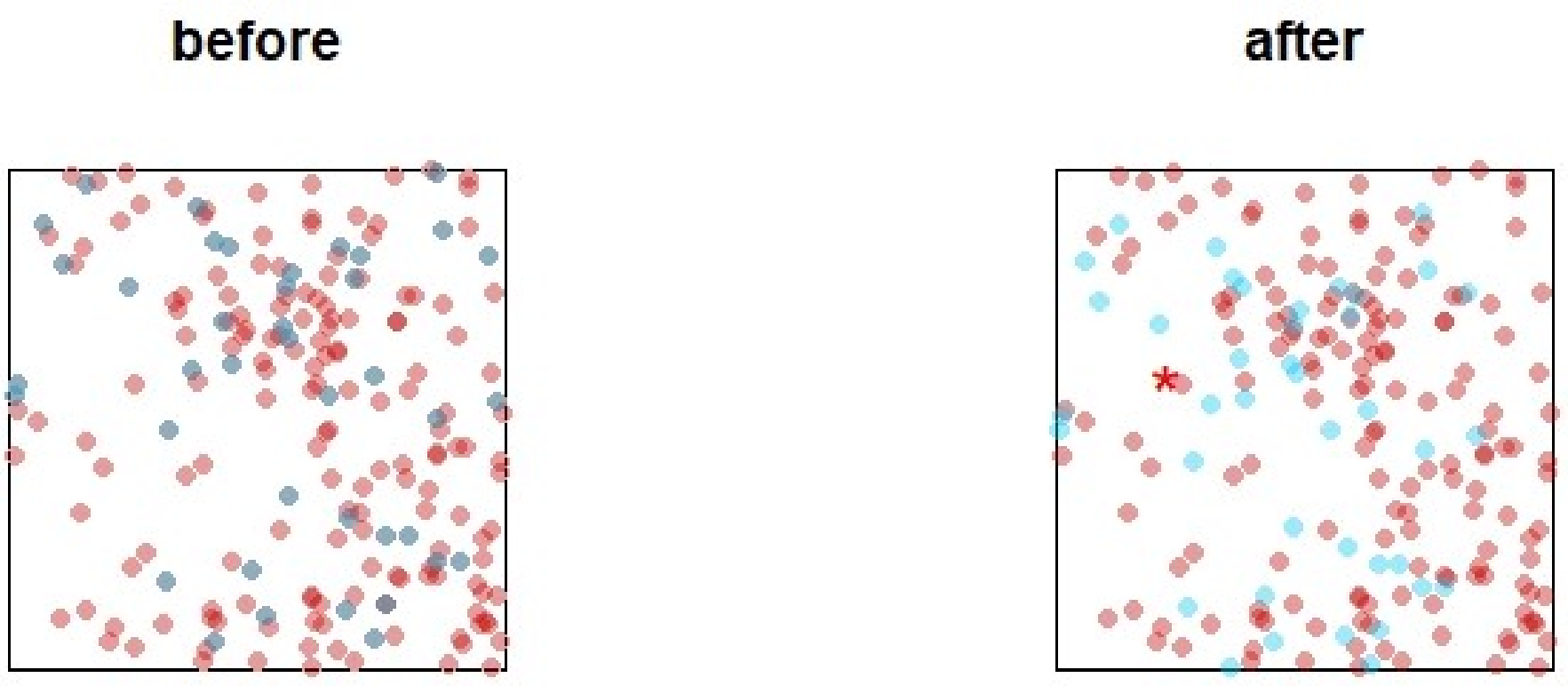
Movement according to habitat suitability

```
Test14 = movepoints(simpoRes=test2, mov_type = "center", mean=0, sd=2, Allpt = T,
        n_var=3, p_move=NULL, p_scale=0.2)
```



Movement according to habitat suitability

```
Test14b = movepoints(simpoRes=test2, mov_type = "center", mean=0, sd=2, Allpt = F,
        n_var=3, p_move=0.2, p_scale=0.2)
```



Thank you for your attention!

Emy.Guilbault@uon.edu.au