

Growth, Survival, Diversity and Concentration

Eric Marcon

Florence Puech

Stuart Sweeney

June 5, 2023

Abstract

Exploratory model of growth and survival depending on Diversity and concentration.

1 Purpose

This exploratory study is a proof of concept for modeling plant growth and survival with respect to relative concentration and diversity.

A set of plants is simulated in a square area with R (R Core Team, 2023). Each plant belongs to an economic sector and has a size. Each sector is drawn separately. Location is generated according to a classical point process, and plant sizes according to a random distribution.

The growth model follows Audretsch and Dohse (2007). Exogenous variables are the plant environment, summarized here by its X coordinate: a positive gradient of growth conditions exists from west to east. The local geographic concentration of the sector around each plant (Lang et al., 2020) and the local diversity (Marcon et al., 2014) of sectors in the neighborhood of each plant are the variables of interest.

The survival model is similar. Exogenous variables are summarized by the Y coordinate.

2 Data generation

2.1 Point set

Plants are simulated in a square window by the *Spatdiv* package.

```

library("SpatDiv")
# Geometry of the window
window_size <- 2000 # Small during development
unit_name <- c("meter", "meters")
# Density
plants_n_per_area <- 100/1E6
# Number of sectors
sectors_n <- 5
# Spatial concentration of plants
# Cluster size
thomas_scale <- window_size / 3
# Number of plants per cluster
thomas_mu <- 100

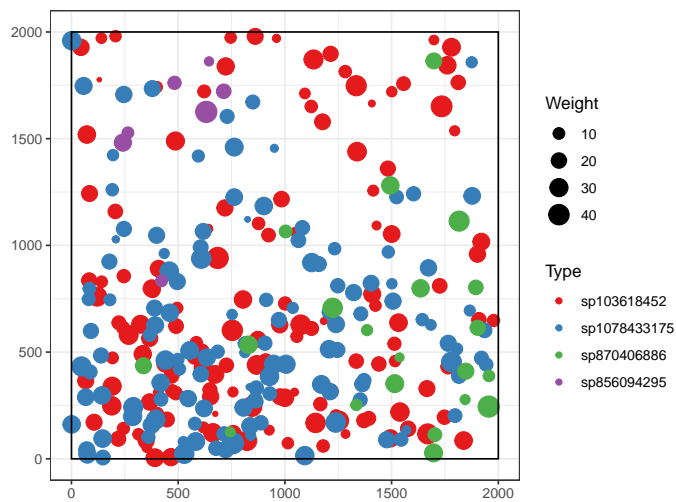
```

The community is simulated: first, a set of clustered sector.

```

library("spatstat")
rSpCommunity(
  # A single community
  n = 1,
  # Number of plants
  size = window_size^2 * plants_n_per_area,
  # Number of sectors
  S = sectors_n,
  # Spatial structure
  Spatial = "Thomas", scale = thomas_scale, mu = thomas_mu,
  # Distribution of plant sizes
  Sizes = "Weibull",
  # Window
  win = square(r = window_size, unitname = unit_name)
) -> spCommunity
# Map
autoplot(spCommunity)

```



Then, a sector with no relative concentration, i.e distributed as a Poisson process with density equal to that of existing points.

```

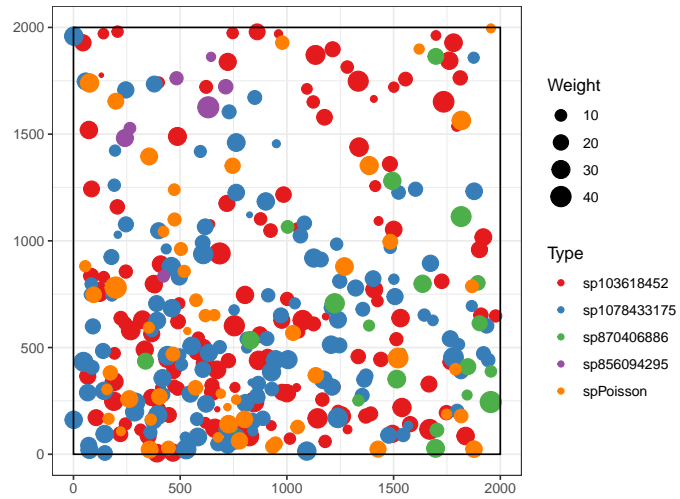
# Simulate the point set.
# ToDo: implement Spatial="Poisson" in SpatDiv::rspCommunity()

```

```

spPoisson <- rpoispp(density(spCommunity)/5)
marks(spPoisson) <- data.frame(
  PointWeight = rweibull(spPoisson$n, shape = 2, scale = 20),
  PointType = "spPoisson"
)
# Merge
spCommunity <- superimpose(spCommunity, spPoisson)
autoplot(spCommunity)

```



```

# Number of plants
spCommunity$n

```

```
## [1] 355
```

```

# Per sector
summary(spCommunity$marks$PointType)

```

```

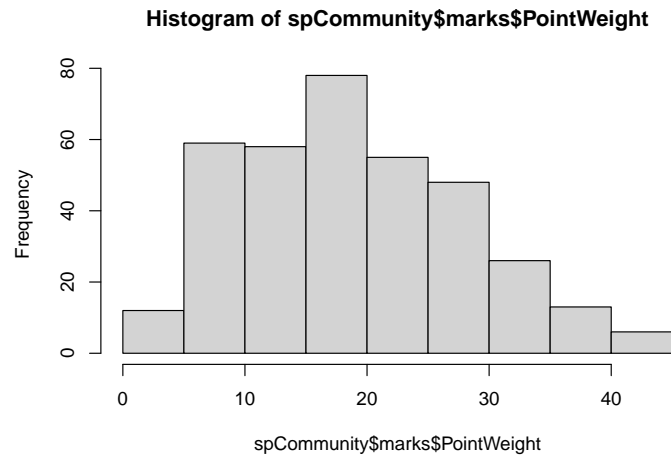
## sp103618452 sp1078433175 sp870406886
##      143          129           21
## sp856094295    spPoisson
##           7           55

```

```

# Sizes
hist(spCommunity$marks$PointWeight)

```



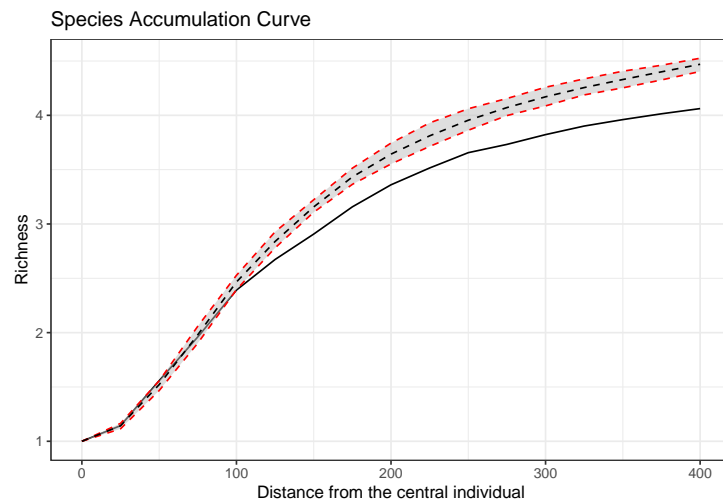
2.2 Parameters

```
# Number of simulations to compute confidence envelopes
n_simulations <- 10 # Small during development
```

3 Diversity

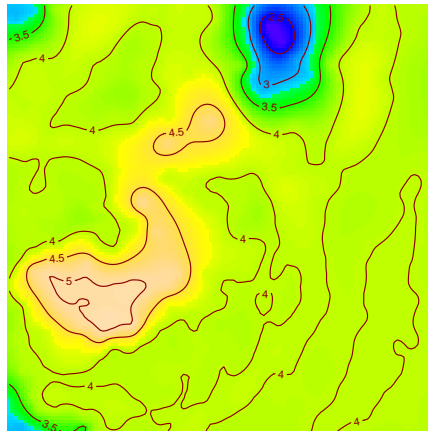
Accumulation.

```
# Average distance between plants
dist_neighbor <- 1 / sqrt(plants_n_per_area)
# Accumulation of diversity with confidence interval (not run)
accum <- DivAccum(
  spCommunity,
  r.seq = c(
    0,
    seq(
      from = dist_neighbor / 4,
      to = dist_neighbor * 4,
      by = dist_neighbor / 4
    )
  ),
  q.seq = 0:2,
  H0 = "RandomLocation",
  NumberOfSimulations = n_simulations,
  Individual = TRUE
)
autoplot(accum)
```

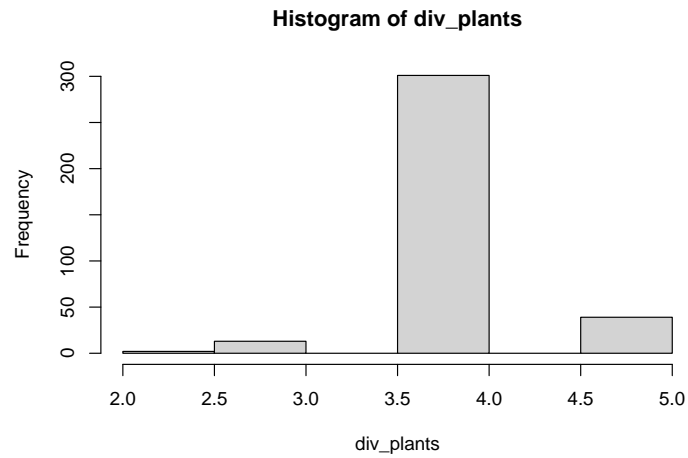


```
# Map the local diversity (richness at 4 x dist to neighbor)
MapPlot(accum, Order = 0, NeighborHood = dist_neighbor * 4)
```

```
## [using ordinary kriging]
```



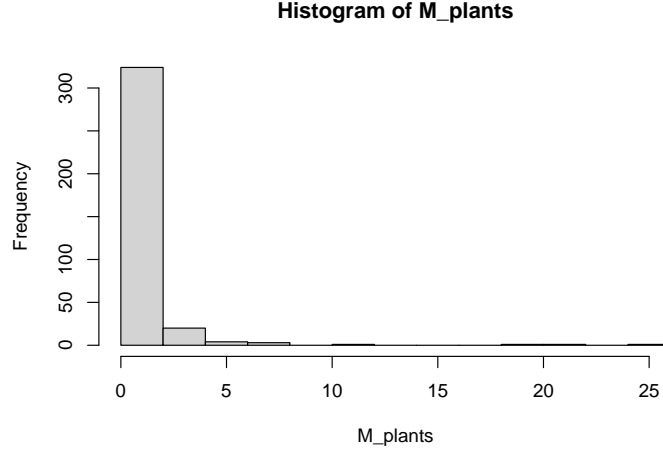
```
# Diversity at 4 times the average distance to neighbor
accum_5 <- DivAccum(
  spCommunity,
  r.seq = c(0, dist_neighbor * 4),
  q.seq = 0:2,
  Individual = TRUE
)
# Extract the data [order, distance, points]
div_plants <- accum_5$Neighborhoods[1, 2, ]
# Distribution of richness
hist(div_plants)
```



4 Concentration

Spatial concentration of sectors.

```
# Compute individual M
library("dbmss")
# Compute concentration of each sector
M_plants <- numeric(spCommunity$n)
for (sector in levels(spCommunity$marks$PointType)) {
  M_sector <- Mhat(
    spCommunity,
    r = c(0, window_size / 10),
    ReferenceType = sector,
    Individual = TRUE
  )
  point_numbers <- as.integer(substring(names(M_sector)[-(1:3)], 3))
  M_plants[point_numbers] <- as.numeric(as.data.frame(M_sector)[2, -(1:3)])
}
hist(M_plants)
```



Correlation

Higher concentration implies lower diversity, which may be an issue to disentangle their effects. Yet, it appears that the correlation remains very low at realistic levels of concentration.

```
cor(M_plants, div_plants)
```

```
## [1] -0.06886661
```

5 Growth model

The model defines growth as

$$\ln(Size_{i,t+1}) = \ln(Size_{i,t}) + \alpha_g x + \beta_g Conc_{i,t} + \gamma_g Div_{i,t} + \epsilon_{i,t}^g.$$

5.1 Parameters

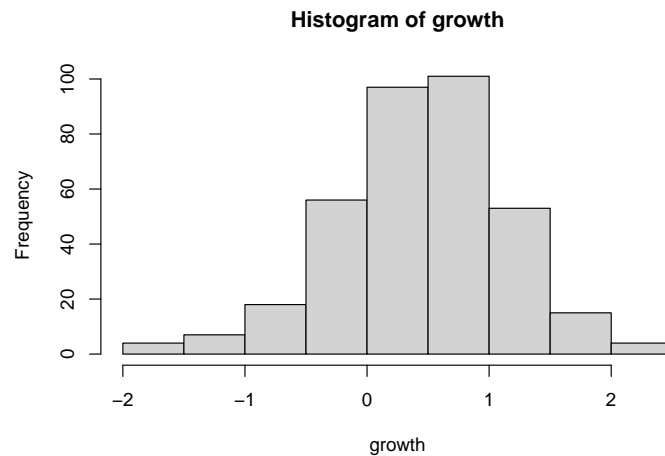
Parameters are chosen to obtain similar effects of all factors. Concentration and diversity are standardized.

```
# x effect between 0 and 1: gradient from west to east
alpha_g <- 1 / max(spCommunity$x)
# Standardize concentration
conc <- scale(log(M_plants+1))
# Concentration effect between -1 and +1 for 1 sd dispersion
beta_g <- 1 / max(conc)
# Standardize diversity
div <- scale(div_plants)
# Diversity effect similar to that of concentration
gamma_g <- 1 / max(div)
# Vector of coefficients for future use
coefficients_g_actual <- c(0, alpha_g, beta_g, gamma_g)
# Error term
epsilon_g <- rnorm(spCommunity$n, 0, .5)
```

5.2 Simulation

Simulate growth.

```
growth <- alpha_g * spCommunity$x + beta_g * conc + gamma_g * div + epsilon_g
# Growth if the log ratio of sizes
hist(growth)
```



```
# Calculate Sizes at t+1
size_t1 <- spCommunity$marks$PointWeight * exp(growth)
```

5.3 Inference

Check that parameters can be inferred from the data.

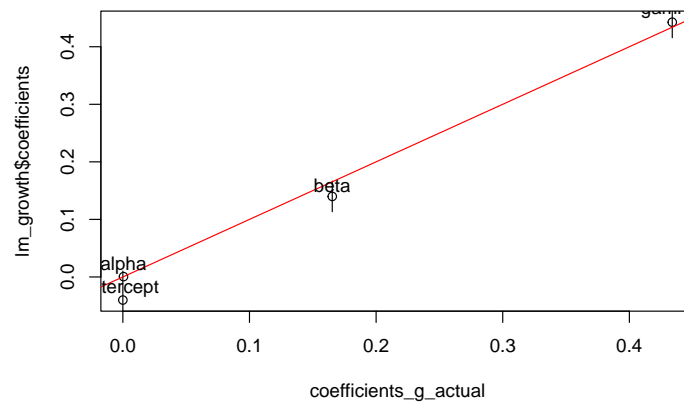
```
# Observed growth
growth_obs <- log(size_t1 / spCommunity$marks$PointWeight)
# Estimate the model
lm_growth <- lm(growth_obs ~ spCommunity$x + conc + div)
summary(lm_growth)
```

```
##
## Call:
## lm(formula = growth_obs ~ spCommunity$x + conc + div)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49141 -0.35129  0.02947  0.32585  1.67617
##
## Coefficients:
##              Estimate Std. Error t value
## (Intercept)  -0.0401209  0.0504126  -0.796
## spCommunity$x  0.0005593  0.0000488  11.462
## conc         0.1399475  0.0266911   5.243
## div          0.4426321  0.0269761  16.408
##
##      Pr(>|t|)
## (Intercept)    0.427
## spCommunity$x < 2e-16 ***
```



```
## conc          2.73e-07 ***
## div           < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.497 on 351 degrees of freedom
## Multiple R-squared:  0.5012, Adjusted R-squared:  0.497
## F-statistic: 117.6 on 3 and 351 DF,  p-value: < 2.2e-16
```

```
# Compare actual and estimated coefficients
plot(x = coefficients_g_actual, y = lm_growth$coefficients)
segments(
  x0 = coefficients_g_actual,
  y0 = lm_growth$coefficients - summary(lm_growth)$coefficients[, 2],
  y1 = lm_growth$coefficients + summary(lm_growth)$coefficients[, 2]
)
abline(a = 0, b = 1, col = "red")
text(
  x = coefficients_g_actual,
  y = lm_growth$coefficients + .02,
  labels = c("intercept", "alpha", "beta", "gamma")
)
```



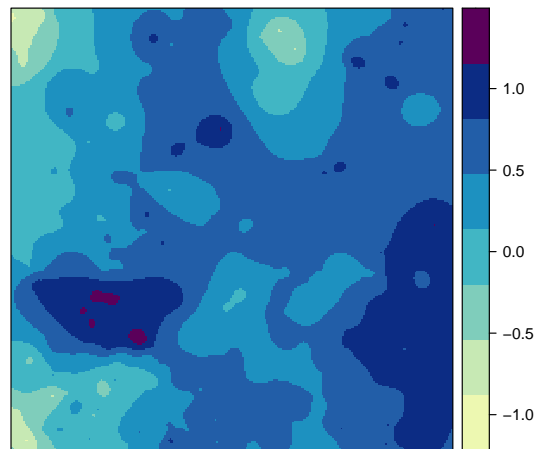
5.4 Map growth

```
library("sp")
# Create a SpatialPointsDataFrame with x, y and growth
growth_sdf <- SpatialPointsDataFrame(
  coords = data.frame(
    x = spCommunity$x,
    y = spCommunity$y,
    data = data.frame(Growth = growth_obs)
  )
)
# Grid for kriging
xy <- gridcentres(spCommunity, 256, 256)
# Format
grid_xy <- SpatialPoints(cbind(xy$x, xy$y))
gridded(grid_xy) <- TRUE
```

```
# Map it
library("automap")
growth_kriged <- autoKrige(
  formula = Growth ~ 1,
  input_data = growth_sdf,
  new_data = grid_xy
)
```

```
## [using ordinary kriging]
```

```
automapPlot(growth_kriged$krige_output, "var1.pred")
```



Compare with the map of diversity.

6 Survival model

A model similar to that of growth makes the log-odds of survival

$$\text{logit}(p_{i,t+1}) = \alpha_s y + \beta_s \text{Conc}_{i,t} + \gamma_s \text{Div}_{i,t} + \epsilon_{i,t}^s.$$

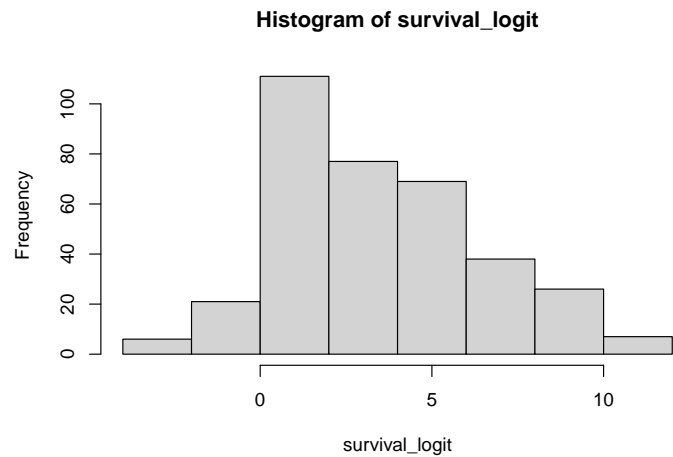
6.1 Parameters

```
# Gradient of survival from south to north
alpha_s <- 10 / max(spCommunity$y)
# Negative effect of concentration, e.g. due to competition
beta_s <- -1
# Positive effect of diversity
gamma_s <- 1
# Vector of parameters
coefficients_s_actual <- c(0, alpha_s, beta_s, gamma_s)
# Error term
epsilon_s <- rnorm(spCommunity$n, 0, .5)
```

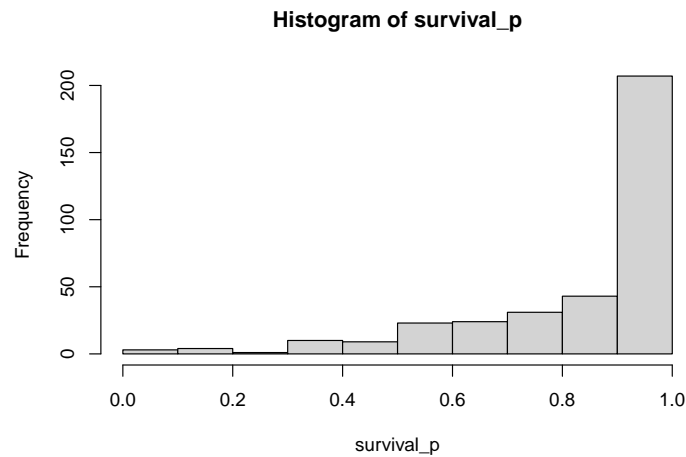
6.2 Simulation

Simulate survival

```
# Log-odds of survival
survival_logit <- alpha_s * spCommunity$y + beta_s * conc + gamma_s * div + epsilon_s
hist(survival_logit)
```



```
# Inv-logit function: survival probability
survival_p <- 1 / (1 + exp(-survival_logit))
hist(survival_p)
```



```
# Draw survival
survival_obs <- rbinom(spCommunity$n, 1, survival_p)
# Observed mortality
1 - sum(survival_obs) / spCommunity$n

## [1] 0.1408451
```

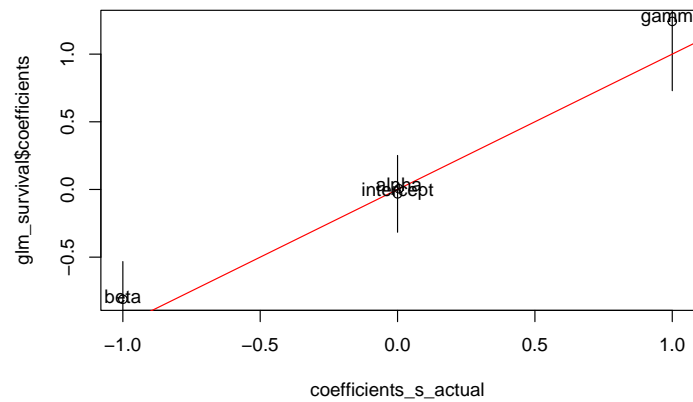
6.3 Inference

Check that parameters can be inferred from the data.

```
# GLM to estimate the logit model
glm_survival <- glm(survival_obs ~ spCommunity$y + conc + div, family = "binomial")
summary(glm_survival)
```

```
##
## Call:
## glm(formula = survival_obs ~ spCommunity$y + conc + div, family = "binomial")
##
## Coefficients:
##             Estimate Std. Error z value
## (Intercept) -0.0323082  0.2836900  -0.114
## spCommunity$y  0.0054870  0.0009541   5.751
## conc         -0.8111739  0.2785259  -2.912
## div           1.2416165  0.5118466   2.426
##             Pr(>|z|)
## (Intercept)  0.90933
## spCommunity$y 8.86e-09 ***
## conc         0.00359 **
## div          0.01528 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 288.61  on 354  degrees of freedom
## Residual deviance: 186.69  on 351  degrees of freedom
## AIC: 194.69
##
## Number of Fisher Scoring iterations: 7
```

```
# Compare actual and estimated coefficients
plot(x = coefficients_s_actual, y = glm_survival$coefficients)
segments(
  x0 = coefficients_s_actual,
  y0 = glm_survival$coefficients - summary(glm_survival)$coefficients[, 2],
  y1 = glm_survival$coefficients + summary(glm_survival)$coefficients[, 2]
)
abline(a = 0, b = 1, col = "red")
text(
  x = coefficients_s_actual,
  y = glm_survival$coefficients + .02,
  labels = c("intercept", "alpha", "beta", "gamma")
)
```

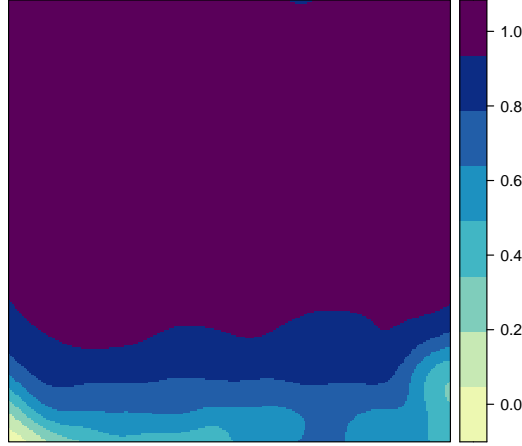


6.4 Map survival

```
# Create a SpatialPointsDataFrame with x, y and survival
survival_sdf <- SpatialPointsDataFrame(
  coords = data.frame(
    x = spCommunity$x,
    y = spCommunity$y,
    data = data.frame(Survival = glm_survival$fitted.values)
  )
# Map it
survival_kriged <- autoKrige(
  formula = Survival ~ 1,
  input_data = survival_sdf,
  new_data = grid_xy
)
```

```
## [using ordinary kriging]
```

```
automapPlot(survival_kriged$krige_output, "var1.pred")
```



Survival is mainly driven by the northern positive gradient.

7 Conclusion

These simulations show that both local concentration and diversity may influence plant growth and survival. We provide methods to compute their values and estimate their effect.

The application of this proof of concept to the real world requires characterizing pertinent exogenous variables (here summarized by the X and Y coordinates) and neighborhood sizes (here arbitrarily set to 1/10 of the window's size for concentration and 4 times the average distance between plants for diversity).

References

- Audretsch, D. B. and D. Dohse (2007, April). Location: A Neglected Determinant of Firm Growth. *Review of World Economics* 143(1), 79–107.
- Lang, G., E. Marcon, and F. Puech (2020). Distance-based measures of spatial concentration: Introducing a relative density function. *The Annals of Regional Science* 64, 243–265.
- Marcon, E., I. Scotti, B. Hérault, V. Rossi, and G. Lang (2014). Generalization of the partitioning of Shannon diversity. *Plos One* 9(3), e90289.
- R Core Team (2023). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.