

Theoretical model

Intraspecific variability is often seen as an unstructured, intrinsic property of individuals, mostly genetic. Here, we rather investigate the effect of spatial variations of the environment on individuals. Our hypothesis is that the individuals can be clones, i.e. have no genetic differences, and still have different responses because the environment in which they strive varies. As the response of an individual is a result of all the environmental conditions it has encountered during its life, the individual level is a tool to describe the environment more acutely. This tool may incorporate some genetic differences, but is not able to discriminate the genetic effect from the effect of local environmental variation, or microsite effect. Moreover, the differences between individuals due to environmental variation does not imply a broader overlap of the species niches. Therefore, this type of intraspecific variability has no effect on species coexistence directly.

We designed and conducted a virtual experiment that aims at illustrating that intraspecific variability, or “individual effects”, can result from unobserved variation in some environmental variables [Clark et al. 2007], therefore accounting for multidimensional species differences which cannot be observed on a few niche axes.

To do so, we first consider a model that depicts the response of a process, e.g. growth, for individual clones of two different species to all the environmental variables that influence it. Henceforth denoted the “Perfect knowledge model”, it represents the perfect knowledge of the process as it occurs in the field and thus includes no residuals :

$$Y_{i,j,t} = \beta_{0,i} + \beta_{1,i} * X_{1,i,j,t} + \dots + \beta_{10,i} * X_{10,i,j,t} \quad \text{“Perfect knowledge model”}$$

where $1 \leq i \leq 2$ are the species ; $1 \leq j \leq 500$ are the individuals ; Y is a response variable, for instance growth ; X_1 to X_{10} are explanatory variables, for instance environmental variables, that can vary with time t ; and the $\beta_{k,i}$, $1 \leq k \leq 10$ depict the species-specific responses to these environmental variables. As we consider clones, individuals of the same species respond the same way to environmental variables, and variation in Y among individuals within species is due to differences in the environment where each individual thrives.

```
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE)
```

```
dim1 <- 500  #dimension of the grid
dim2 <- 500
nsp <- 2     #number of species
nind <- 500  #number of individuals per species
ntime <- 2   #number of observations per individual
nobs <- nsp * nind * ntime #number of observations
```

```
# 10 parameters of species 1 and 2 To modulate the overlap of
# the growth of both species, use the 9 random parameters :
# the closer to the second parameter, the more confused the
# statistical model will be. If many parameters are negative,
# it can even lead to a negative mean G ~ L relationship. In
# the current code, 5 variables have a positive effect on
# growth and 4 a negative effect.
```

```
param1 = c(0.25, 0.15, abs(runif(5, -0.05, 0.05)), -abs(runif(4,
```

```

-0.05, 0.05)))
param2 = c(0.2, 0.1, abs(runif(5, -0.05, 0.05)), -abs(runif(4,
-0.05, 0.05)))

par_sp <- cbind(param1, param2)

# Adding genetic IV
par_ind1 <- rbind(rnorm(nind, param1[1], abs(param1[1]/4)), rnorm(nind,
param1[2], abs(param1[2]/4)), rnorm(nind, param1[3], abs(param1[3]/4)),
rnorm(nind, param1[4], abs(param1[4]/4)), rnorm(nind, param1[5],
abs(param1[5]/4)), rnorm(nind, param1[6], abs(param1[6]/4)),
rnorm(nind, param1[7], abs(param1[7]/4)), rnorm(nind, param1[8],
abs(param1[8]/4)), rnorm(nind, param1[9], abs(param1[9]/4)),
rnorm(nind, param1[10], abs(param1[10]/4)), rnorm(nind, param1[11],
abs(param1[11]/4)))
par_ind2 <- rbind(rnorm(nind, param2[1], abs(param2[1]/4)), rnorm(nind,
param2[2], abs(param2[2]/4)), rnorm(nind, param2[3], abs(param2[3]/4)),
rnorm(nind, param2[4], abs(param2[4]/4)), rnorm(nind, param2[5],
abs(param2[5]/4)), rnorm(nind, param2[6], abs(param2[6]/4)),
rnorm(nind, param2[7], abs(param2[7]/4)), rnorm(nind, param2[8],
abs(param2[8]/4)), rnorm(nind, param2[9], abs(param2[9]/4)),
rnorm(nind, param2[10], abs(param2[10]/4)), rnorm(nind, param2[11],
abs(param2[11]/4)))
par_ind <- cbind(par_ind1, par_ind2)

```

We fix the species parameters of the “Perfect knowledge model” as follows:

Parameters of species 1 : $\beta_0 = 0.25$, $\beta_1 = 0.15$, β_2 to β_6 are chosen randomly between -0.05 and 0.05 and β_7 to β_{10} are chosen randomly between 0 and 0.05.

Parameters of species 2 : $\beta_0 = 0.2$, $\beta_1 = 0.1$, β_2 to β_{10} are chosen the same way as for species 1.

The difference between the species is imposed by those parameters. Here, the species 1 is more performant on average thanks to its higher intercept (β_0) and reaction to the first environmental variable (β_1). The first environmental variable (X_1) has a higher weight in the computation of the response variable, as would be light for growth for instance.

To account for genetic variability in our generated data, we add an intraspecific variability in species parameters by sampling each individual parameter in a normal distribution centered on the species mean parameter and with a standard deviation of a quarter of the species mean parameter.

To represent the spatialised environment of such a forest plot, we build a 2D matrix of dimension 500×500 for each environmental variable at a time t_0 , by randomly generating them with spatial autocorrelation. We here consider that all environmental variables are independent. Each variable is simulated by using the gstat package, enabling to create autocorrelated random fields. A spherical semivariogram model is used for each of the ten environmental variables, with a mean of 0 for each explanatory variable ($\beta = 0$), a sill of 1 ($\text{psill} = 1$) for each and a range of 50 ($\text{range} = 50$).

```

# Matrixes of variables Structure (coordinates)
xy <- expand.grid(1:dim1, 1:dim2)
names(xy) <- c("x", "y")

# Spatial models
mods <- list()
mods[[1]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",

```

```

    range = 50, nugget = 0), nmax = 20)
mods[[2]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[3]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[4]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[5]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[6]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[7]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[8]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[9]] <- gstat::gstat(formula = z ~ 1, locations = ~x + y,
    dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)
mods[[10]] <- gstat::gstat(formula = z ~ 1, locations = ~x +
    y, dummy = T, beta = 0, model = gstat::vgm(psill = 1, model = "Sph",
    range = 50, nugget = 0), nmax = 20)

```

```

# this chunk takes time and can be skipped (files will be
# loaded)

# Z predictions
Variables <- list()
Vars_t0 <- data.frame(matrix(nrow = dim1 * dim2, ncol = 12))
Vars_t0[, 1:2] <- xy
for (k in 1:10) {
    Variables[[k]] <- predict(mods[[k]], newdata = xy, nsim = 1)
    Vars_t0[, k + 2] <- Variables[[k]][, 3]
    gridded(Variables[[k]]) = ~x + y
    print(spplot(Variables[[k]]))
}
colnames(Vars_t0) <- c("x", "y", "Var1", "Var2", "Var3", "Var4",
    "Var5", "Var6", "Var7", "Var8", "Var9", "Var10")

save(Variables, file = here::here("outputs", "theoretical_model",
    "Variables.RData"))
save(Vars_t0, file = here::here("outputs", "theoretical_model",
    "Vars_t0.RData"))

```

We then consider that Y has been measured at two times, t_0 and t_1 , for each individual as it would be under periodic forest plot censuses for instance. To each of the 500 individuals is randomly assigned a pair of coordinates within this spatialised environment. We then consider that some of the environmental

variables (light, temperature, humidity, nutrient availability for instance) changed between t_0 and t_1 and others did not (slope, altitude for instance). For the first environmental variable which has the stronger impact on Y values (like light for growth for instance) and another randomly drawn environmental variable, we compute value at t_1 as $t_0 + \epsilon$, $\epsilon \sim \mathcal{N}(0, 0.1)$. For two other environmental variables randomly drawn, $X_{t_1} = X_{t_0} + |\epsilon|$, $\epsilon \sim \mathcal{N}(0, 0.1)$ (they increase) and for two other environmental variables, $X_{t_1} = X_{t_0} - |\epsilon|$, $\epsilon \sim \mathcal{N}(0, 0.1)$ (they decrease).

This leads to two repeated measurements of Y for each individual of each species, i.e. 2000 values of $Y_{i,j,t}$, with $i = [1; 2]$; $j = [1, 50]$; $t = [t_0; t_1]$.

```
# Second observation The variables have to vary in order to
# avoid identifiability problems + avoid a null residual
# variance
Vars_t1 <- Vars_t0
changing_variables <- sample(2:10, 5)
Vars_t1[, 3] <- Vars_t1[, 3] + abs(rnorm(1, mean = 0, sd = 0.1))
Vars_t1[, 2 + changing_variables[1]] <- Vars_t1[, 2 + changing_variables[1]] +
  abs(rnorm(1, mean = 0, sd = 0.1))
Vars_t1[, 2 + changing_variables[2]] <- Vars_t1[, 2 + changing_variables[2]] +
  abs(rnorm(1, mean = 0, sd = 0.1))
Vars_t1[, 2 + changing_variables[3]] <- Vars_t1[, 2 + changing_variables[3]] -
  abs(rnorm(1, mean = 0, sd = 0.1))
Vars_t1[, 2 + changing_variables[4]] <- Vars_t1[, 2 + changing_variables[4]] -
  abs(rnorm(1, mean = 0, sd = 0.1))
Vars_t1[, 2 + changing_variables[5]] <- Vars_t1[, 2 + changing_variables[5]] +
  rnorm(1, mean = 0, sd = 0.1)

# Sample the coordinates of the individuals in the matrices
# of variables for each species
Coord <- data.frame(matrix(nrow = 2, ncol = nind * nsp))
for (k in 1:ncol(Coord)) {
  Coord[, k] <- sample(1:dim1, 2, rep = TRUE)
}
# Making sure there is no duplicate in the coordinates
# couples (one individual only on a couple of coordinates)
while (any(duplicated.array(Coord, MARGIN = 2)) == TRUE) {
  Coord[, which(duplicated.array(Coord, MARGIN = 2) == TRUE)] <- sample(1:dim1,
    2, rep = TRUE)
}

Coord1_sp1 = Coord[1, 1:nind]
Coord2_sp1 = Coord[2, 1:nind]
Coord1_sp2 = Coord[1, (nind + 1):(2 * nind)]
Coord2_sp2 = Coord[2, (nind + 1):(2 * nind)]

# Species 1

# explanatory variables for species 1
VarExp_sp1_t0 <- data.frame(matrix(nrow = nind, ncol = 10))

for (ind in 1:nind) {
  VarExp_sp1_t0[ind, ] <- Vars_t0[Vars_t0$x == Coord1_sp1[,
    ind] & Vars_t0$y == Coord2_sp1[, ind], ][, 3:12]
}
```

```

VarExp_sp1_t1 <- data.frame(matrix(nrow = nind, ncol = 10))
for (ind in 1:nind) {
  VarExp_sp1_t1[ind, ] <- Vars_t1[Vars_t1$x == Coord1_sp1[,
    ind] & Vars_t1$y == Coord2_sp1[, ind], ], 3:12]
}

# Species 2
VarExp_sp2_t0 <- data.frame(matrix(nrow = nind, ncol = 10))

for (ind in 1:nind) {
  VarExp_sp2_t0[ind, ] <- Vars_t0[Vars_t0$x == Coord1_sp2[,
    ind] & Vars_t0$y == Coord2_sp2[, ind], ], 3:12]
}

VarExp_sp2_t1 <- data.frame(matrix(nrow = nind, ncol = 10))

for (ind in 1:nind) {
  VarExp_sp2_t1[ind, ] <- Vars_t1[Vars_t1$x == Coord1_sp2[,
    ind] & Vars_t1$y == Coord2_sp2[, ind], ], 3:12]
}

# Large explanatory variable matrix for all observations
# (including 1 for intercept)
VarExp <- as.matrix(rbind(VarExp_sp1_t0, VarExp_sp1_t1, VarExp_sp2_t0,
  VarExp_sp2_t1))
X <- cbind(rep(1, nobs), VarExp)

```

The main explanatory variable is switched to the natural logarithm.

These two virtual datasets (with and without genetic variability) is then analysed with a “Partial knowledge model”, which represents the process understanding ecologists can derive using these datasets and from their incomplete characterisation of the environment : only a few variables (here only one, e.g. light) are actually measured and taken into account.

$$Y_{i,j,t} = [\beta_{0,i'} + b_{i,j}] + \beta_{1,i'} * X1_{i,j,t} + \epsilon_{i,j,t} \quad \text{“Partial knowledge model”}$$

Priors :

$$\epsilon_{i,j,t} \sim \mathcal{N}(0, V)$$

$$\beta_{k,i'} \sim \mathcal{N}_2(0, 1), k = [0, 1], i = [1, 2]$$

$$b_{i,j} \sim \mathcal{N}(0, V_{b_i}), i = [1, 2], j = [1, 500]$$

Second level priors :

$$V \sim \mathcal{IG}(10^{-3}, 10^{-3})$$

$$V_{b_i} \sim \mathcal{IG}_2(10^{-3}, 10^{-3})$$

This model includes a random individual effect on the intercept ($b_{i,j}$) allowing to account for any variability among individuals within species regarding this parameter.

We ran this model twice, with the datasets generated with and without intraspecific genetic variability. We then acquire the species parameters and the intraspecific variability generated with genetic intraspecific variability and get only intraspecific variability from the model with clones.

These models were fitted with bayesian inference thanks to Stan, with the brms package in Rstudio.

```

# Growth of each individual

Growth <- data.frame(Species = rep(c(1, 2), each = nind * ntime),
  Ind = c(rep(c(1:nind), ntime), rep((nind + 1):(nind * 2),
    ntime)), Time = rep(c(1, 2, 1, 2), each = nind), Growth = double(nobs),
  Growth_sp = double(nobs))

Xloglight <- X

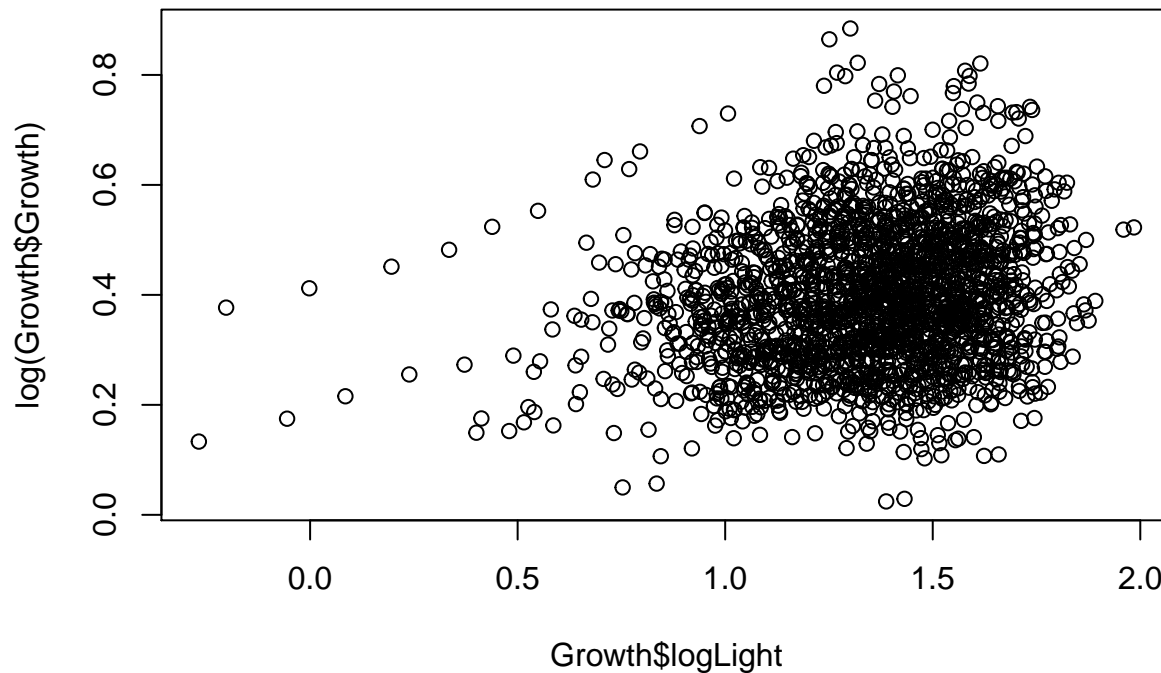
# constant to avoid NaNs while logging
const_log <- 4
Xloglight[, 2] <- log(Xloglight[, 2] + const_log)

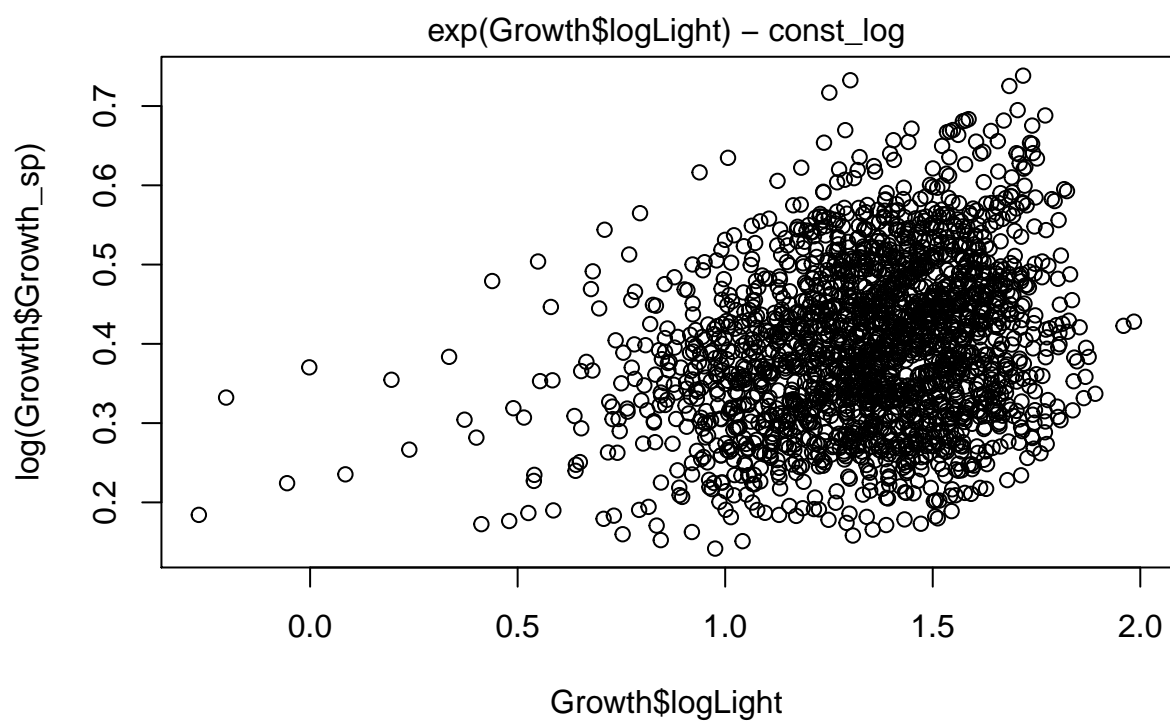
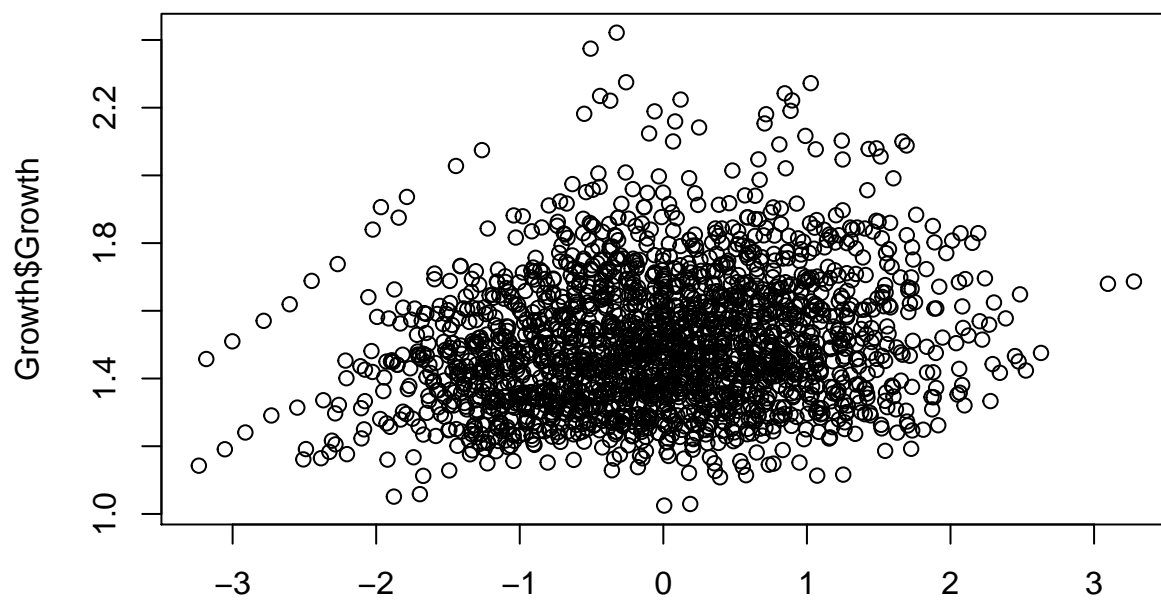
# Process model : log(G) =
# beta1+beta2*log(Light)+...+beta10*X10
Growth$Growth <- exp(rowSums(Xloglight * t(par_ind[, Growth$Ind])))
# Without individual parameters
Growth$Growth_sp <- exp(rowSums(Xloglight * t(par_sp[, Growth$Species])))

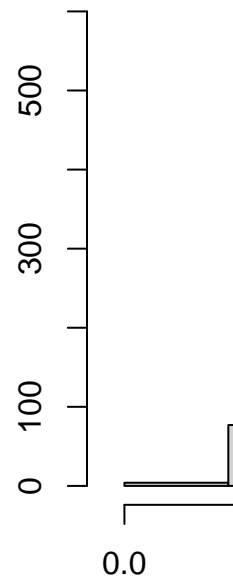
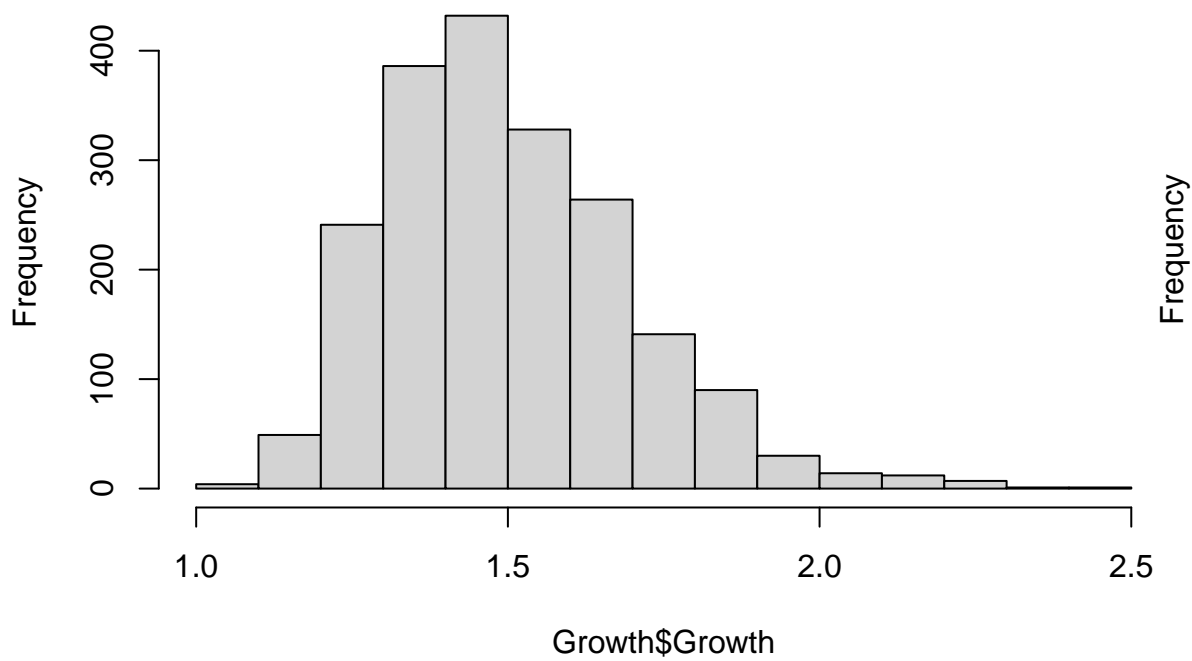
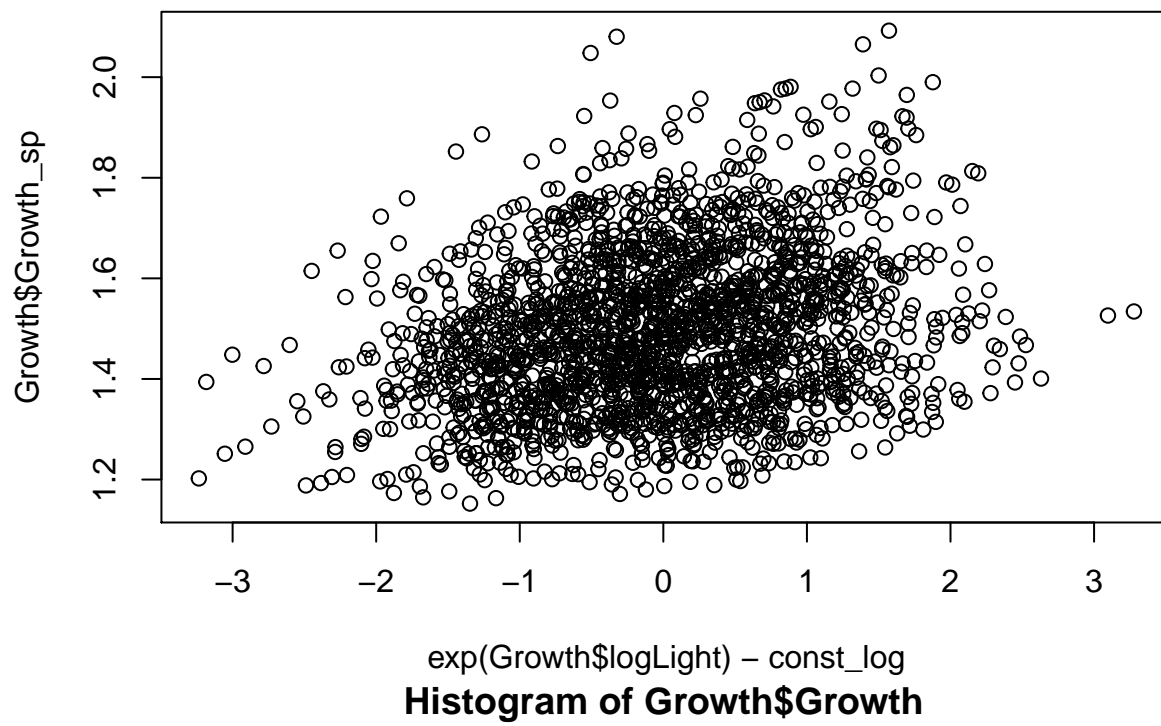
Growth$Species <- as.factor(Growth$Species)
Growth$Ind <- as.factor(Growth$Ind)
# Only one variable is selected to build the coming
# statistical model
Growth$logLight <- Xloglight[, 2]

```

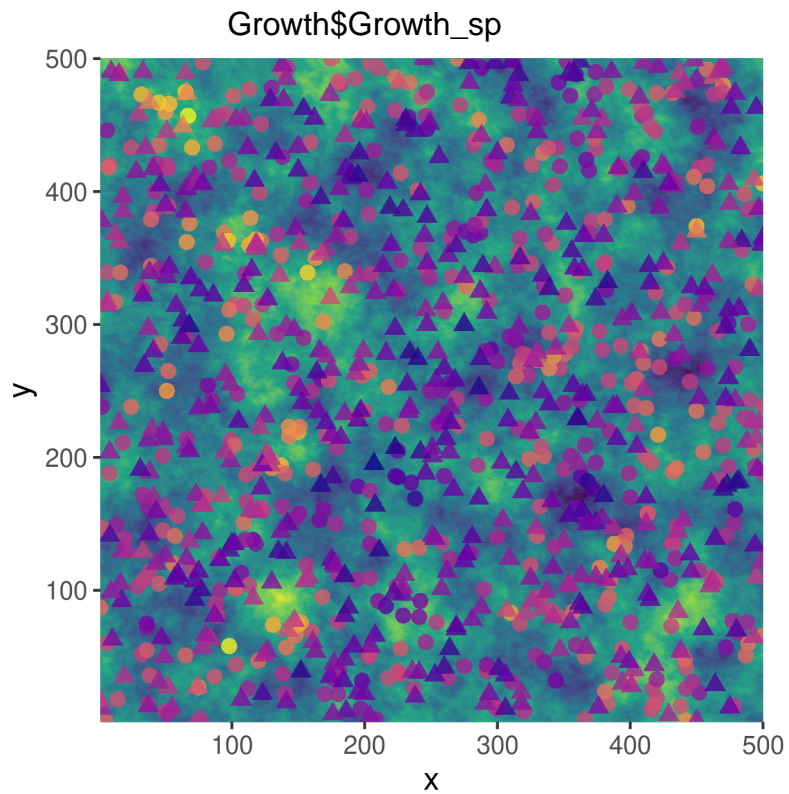
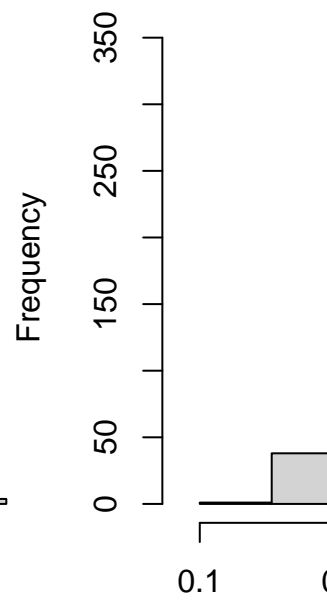
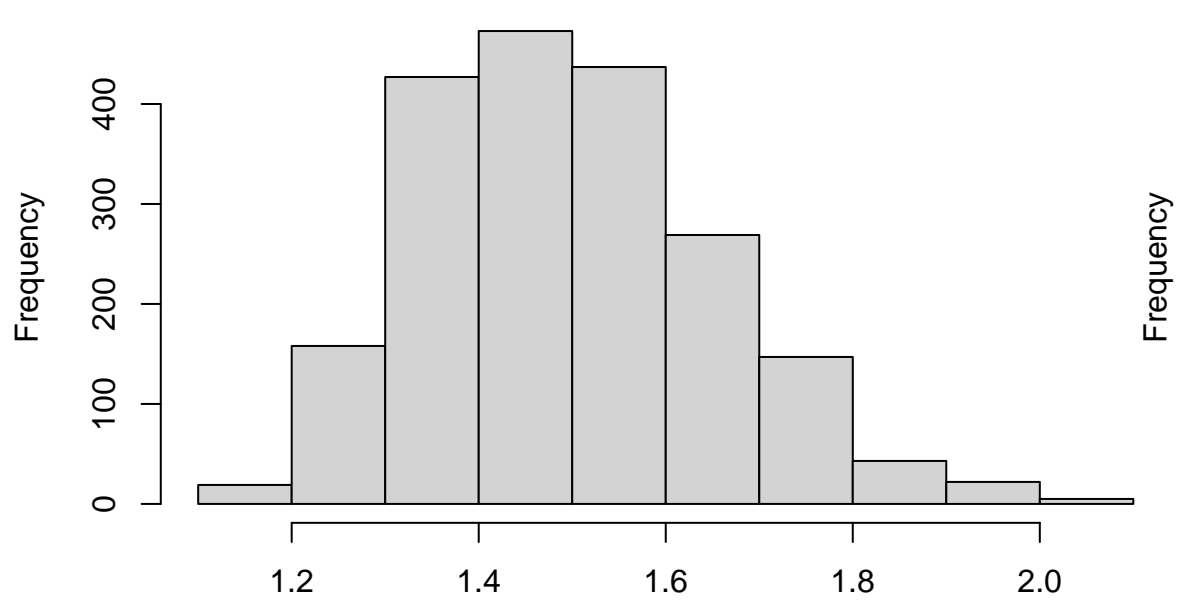
Following are some plots to visualise the data, and a plot helping to visualise the link between the virtual landscape and the growth values (without genetic variability).



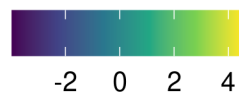




Histogram of Growth\$Growth_sp



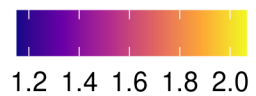
Environmental variable X1



Species



Response Y



On this figure, we can already see that microsite effects, which is the effect of the local multidimensional environment on the observed variable, can result in local reversals of the competitive hierarchy between species, i.e. the local outcome of competition can be opposite to the mean hierarchy : at one point of the space-time, an individual of Species 1 can overcome an individual of Species 2, whilst Species 1 is, on average, the fittest of the two species. Doing this, microsite effects foster the coexistence of Species 1 and Species 2.

```
Growth$logGrowth <- log(Growth$Growth)
Growth$logGrowth_sp <- log(Growth$Growth_sp)

# Statistical model with only one variable X

Data_brms_sp_1 <- data.frame(Y = Growth[which(Growth$Species ==
  1), ]$logGrowth_sp, X1 = Growth[which(Growth$Species == 1),
  ]$logLight, tree = Growth[which(Growth$Species == 1), ]$Ind)
# dplyr::mutate(Y=scale(Y), X1=scale(X1))

Data_brms_sp_2 <- data.frame(Y = Growth[which(Growth$Species ==
  2), ]$logGrowth_sp, X1 = Growth[which(Growth$Species == 2),
  ]$logLight, tree = Growth[which(Growth$Species == 2), ]$Ind)
# dplyr::mutate(Y=scale(Y), X1=scale(X1))

Data_brms_sp_1_IV <- data.frame(Y = Growth[which(Growth$Species ==
  1), ]$logGrowth, X1 = Growth[which(Growth$Species == 1),
  ]$logLight, tree = Growth[which(Growth$Species == 1), ]$Ind)
# dplyr::mutate(Y=scale(Y), X1=scale(X1))

Data_brms_sp_2_IV <- data.frame(Y = Growth[which(Growth$Species ==
  2), ]$logGrowth, X1 = Growth[which(Growth$Species == 2),
  ]$logLight, tree = Growth[which(Growth$Species == 2), ]$Ind)
# dplyr::mutate(Y=scale(Y), X1=scale(X1))

options(m.cores = 2)

### Priors

prior_brms <- c(brms::prior(normal(0, 1), class = "Intercept"),
  brms::prior(normal(0, 1), class = "b"), brms::prior(inv_gamma(10^-3,
  10^-3), class = "sd"))

### Models

brms_mod_sp_1 <- brms::brm(formula = Y ~ 1 + X1 + (1 | tree),
  data = Data_brms_sp_1, prior = prior_brms, iter = 1e+05,
  warmup = 50000, thin = 50, chains = 2, cores = 2)

save(brms_mod_sp_1, file = here::here("outputs", "theoretical_model",
  "brms_mod_sp_1.RData"))

brms_mod_sp_2 <- brms::brm(formula = Y ~ 1 + X1 + (1 | tree),
  data = Data_brms_sp_2, prior = prior_brms, iter = 1e+05,
  warmup = 50000, thin = 50, chains = 2, cores = 2)

save(brms_mod_sp_2, file = here::here("outputs", "theoretical_model",
```

```

"brms_mod_sp_2.RData"))

brms_mod_sp_1_IV <- brms::brm(formula = Y ~ 1 + X1 + (1 | tree),
  data = Data_brms_sp_1_IV, prior = prior_brms, iter = 1e+05,
  warmup = 50000, thin = 50, chains = 2, cores = 2)

save(brms_mod_sp_1_IV, file = here::here("outputs", "theoretical_model",
  "brms_mod_sp_1_IV.RData"))

brms_mod_sp_2_IV <- brms::brm(formula = Y ~ 1 + X1 + (1 | tree),
  data = Data_brms_sp_2_IV, prior = prior_brms, iter = 1e+05,
  warmup = 50000, thin = 50, chains = 2, cores = 2)

save(brms_mod_sp_2_IV, file = here::here("outputs", "theoretical_model",
  "brms_mod_sp_2_IV.RData"))

```

We visualise the convergence and the results of the models thanks to trace and density plots and the summary of the models.

Warning: Parts of the model have not converged (some Rhats are > 1.05). Be careful when
analysing the results! We recommend running more iterations and/or setting stronger priors.

	β_0	β_1	V_b	V
Species 1				
Estimate	6.4e-02	2.9e-01	9.5e-02	1.9e-03
Estimation error	4.7e-03	2.4e-03	3e-03	6.1e-05
Species 2				
Estimate	1.3e-01	1.5e-01	7.6e-02	5.2e-04
Estimation error	3.3e-03	6.3e-04	2.4e-03	1.7e-05

We infer a high intraspecific variability even in the absence of genetic intraspecific variability. Therefore, observed intraspecific variability does not necessarily reveal intrinsic (mainly genetic) intraspecific variability, but can also reveal hidden dimensions of the environment.

The mean and quantiles of the results of the models are used to visualise the inferred link between Y and $X1$. To do so, we create a sequence of explanatory variable values and compute the response variable with the parameters inferred with the models.

```

# Create a sequence of 100 values of X (which is light for
# instance)
npred <- 100
logLight_seq <- seq(min(Growth$logLight), max(Growth$logLight),
  length.out = npred)
Sd_b_IV <- mean(c(MCMC_brms_sp_1_IV[[1]][, 3], MCMC_brms_sp_1_IV[[2]][,
  3]))
Sd_b_sp <- mean(c(MCMC_brms_sp_1[[1]][, 3], MCMC_brms_sp_1[[2]][,
  3]))

# Without genetic variation#
Param_1_sp <- data.frame(b1.1 = numeric(2000), b1.2 = numeric(2000),
  ind = numeric(2000))

```

```

Param_1_sp[, 1] <- c(MCMC_brms_sp_1[[1]][, 1], MCMC_brms_sp_1[[2]][,
1])
Param_1_sp[, 2] <- c(MCMC_brms_sp_1[[1]][, 2], MCMC_brms_sp_1[[2]][,
2])
Param_1_sp[, 3] <- rnorm(2000, 0, Sd_b_sp)
Log_growth_pred_sp1_sp <- matrix(nrow = 2000, ncol = npred)
for (n in 1:npred) {
  Log_growth_pred_sp1_sp[, n] <- Param_1_sp$b1.1 + Param_1_sp$b1.2 *
    logLight_seq[n] + Param_1_sp$ind
}

Param_2_sp <- data.frame(b2.1 = numeric(2000), b2.2 = numeric(2000),
ind = numeric(2000))
Param_2_sp[, 1] <- c(MCMC_brms_sp_2[[1]][, 1], MCMC_brms_sp_2[[2]][,
1])
Param_2_sp[, 2] <- c(MCMC_brms_sp_2[[1]][, 2], MCMC_brms_sp_2[[2]][,
2])
Param_2_sp[, 3] <- rnorm(2000, 0, Sd_b_sp)
Log_growth_pred_sp2_sp <- matrix(nrow = 2000, ncol = npred)
for (n in 1:npred) {
  Log_growth_pred_sp2_sp[, n] <- Param_2_sp$b2.1 + Param_2_sp$b2.2 *
    logLight_seq[n] + Param_2_sp$ind
}

Mean_1_sp <- apply(Log_growth_pred_sp1_sp, 2, mean)
Quant_1_sp <- apply(Log_growth_pred_sp1_sp, 2, quantile, c(0.025,
0.975))
Mean_2_sp <- apply(Log_growth_pred_sp2_sp, 2, mean)
Quant_2_sp <- apply(Log_growth_pred_sp2_sp, 2, quantile, c(0.025,
0.975))

Marg_1_sp <- data.frame(logLight_seq, Mean_sp = Mean_1_sp, Q025_sp = Quant_1_sp[1,
], Q975_sp = Quant_1_sp[2, ])
Marg_2_sp <- data.frame(logLight_seq, Mean_sp = Mean_2_sp, Q025_sp = Quant_2_sp[1,
], Q975_sp = Quant_2_sp[2, ])

Marg_sp <- rbind(Marg_1_sp, Marg_2_sp)
names(Marg_sp) <- c("logLight", "logGrowth_sp", "logQ025_sp",
"logQ975_sp")
Marg_sp$Species <- rep(c(1, 2), each = npred)
Marg_sp$Species <- as.factor(Marg_sp$Species)

# Use exponential to draw a logarithmical curve
Marg_sp$Light <- exp(Marg_sp$logLight) - const_log
Marg_sp$Growth_sp <- exp(Marg_sp$logGrowth_sp)
Marg_sp$Q025_sp <- exp(Marg_sp$logQ025_sp)
Marg_sp$Q975_sp <- exp(Marg_sp$logQ975_sp)

Growth$Light <- exp(Growth$logLight) - const_log

B <- ggplot2::ggplot(data = Growth[which(Growth$Time == 1), ],
  ggplot2::aes(Light, Growth_sp)) + ggplot2::geom_point(ggplot2::aes(colour = Species)) +
  ggplot2::geom_line(data = Marg_sp, size = 1, ggplot2::aes(colour = Species)) +

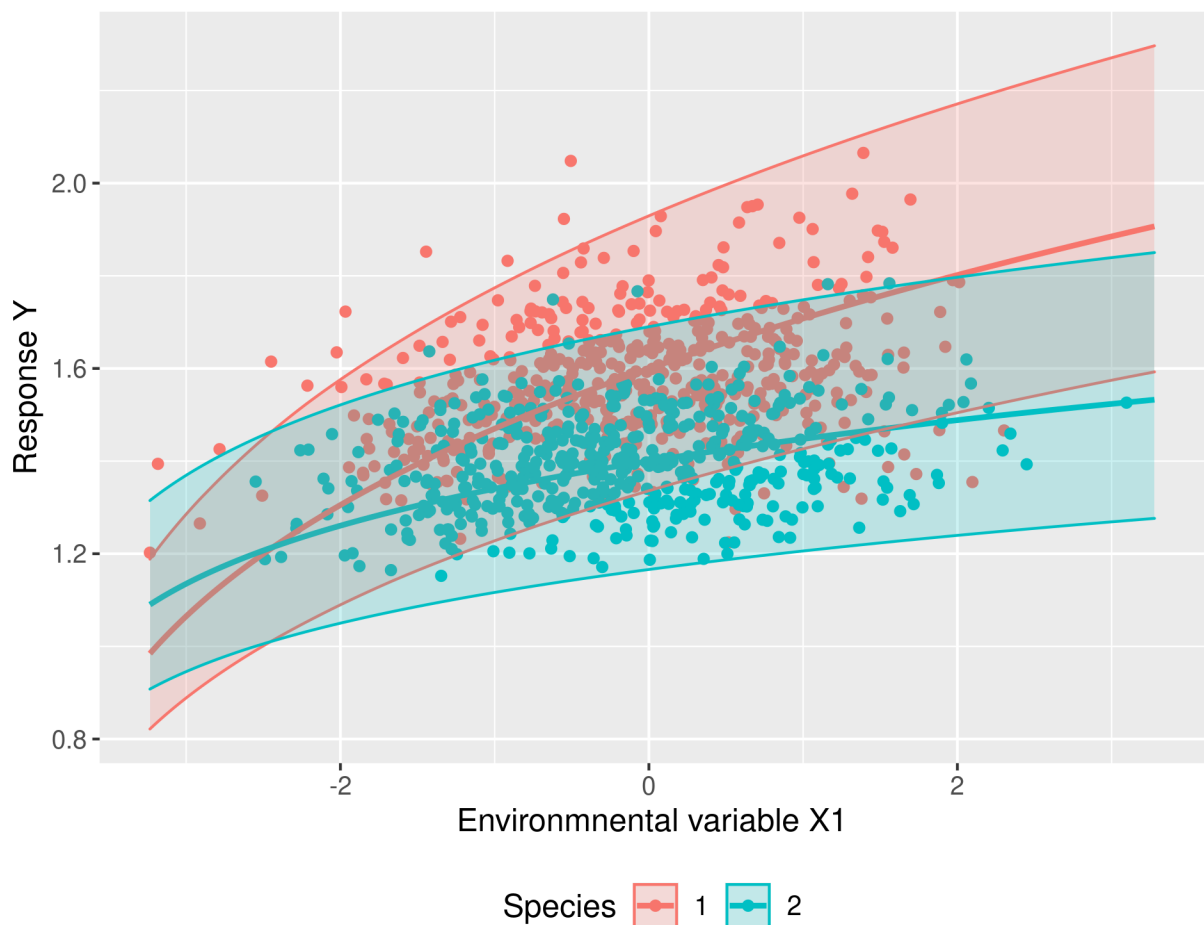
```

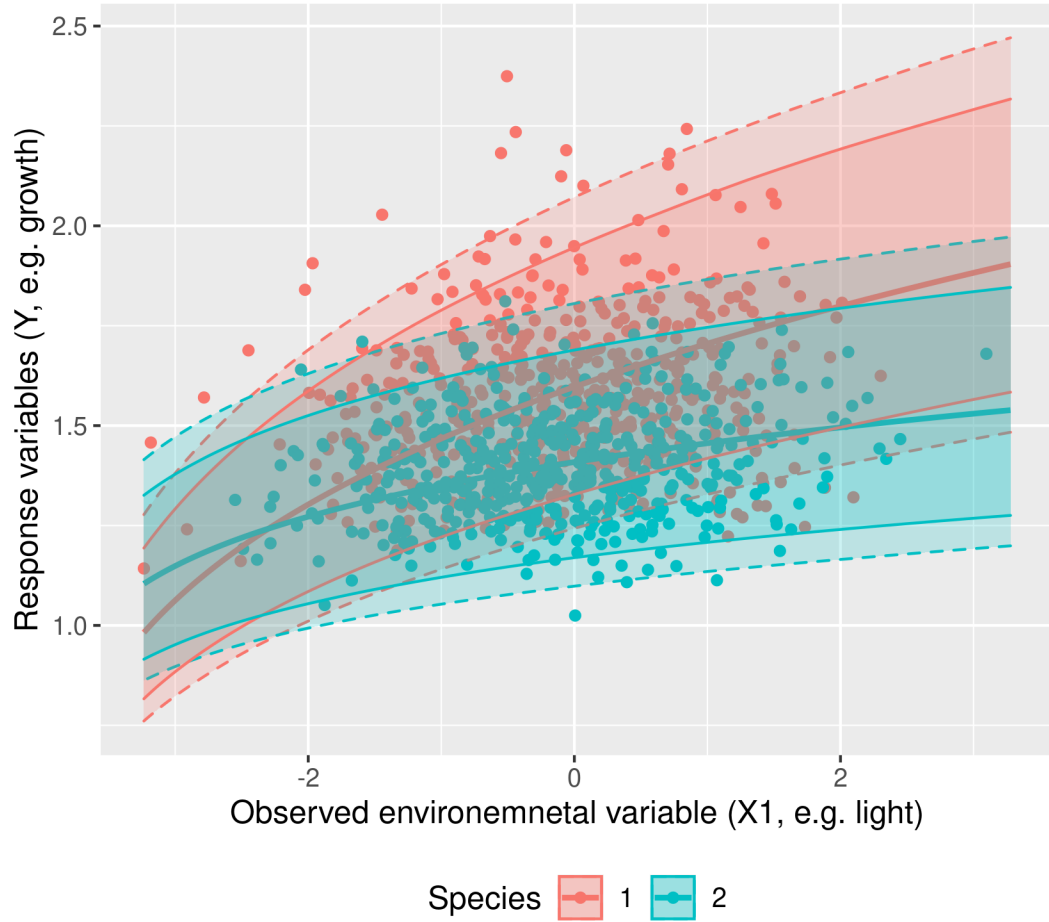
```

ggplot2::geom_ribbon(data = Marg_sp, alpha = 0.2, ggplot2::aes(Light,
  ymin = Q025_sp, ymax = Q975_sp, colour = Species, fill = Species)) +
ggplot2::coord_fixed(ratio = 3) + ggplot2::xlab("Environmental variable X1") +
ggplot2::ylab("Response Y") + ggplot2::guides(fill = ggplot2::guide_legend(title.position = "top",
  label.position = "bottom")) + ggplot2::theme(legend.position = "bottom",
  legend.title = ggplot2::element_text(size = 12), legend.text = ggplot2::element_text(size = 10),
  text = ggplot2::element_text(size = 12))

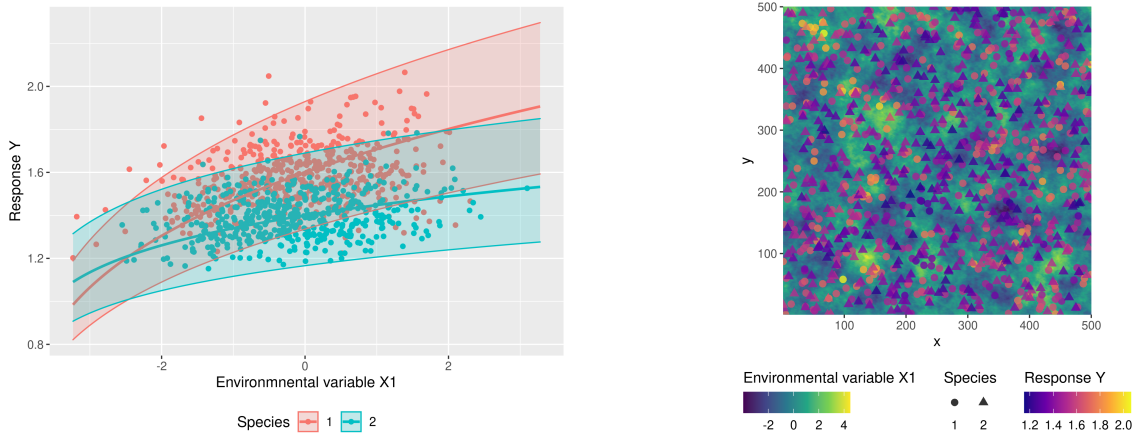
ggplot2::ggsave(filename = "Partial_knowledge_model.png", plot = B,
  path = here::here("outputs", "theoretical_model", "figures"),
  device = "png", width = 7, height = 5)
knitr::include_graphics(here::here("outputs", "theoretical_model",
  "figures", "Partial_knowledge_model.png"))

```





In these figures, the solid and bold lines represent the mean growth rate of Species 1 (red) and Species 2 (blue) as computed with the parameters retrieved from the model with or without genetic variability respectively. The plain lines represent the 95% interval of the posteriors from the model without genetic variability and the dotted lines show the 95% confidence interval of the posteriors from the model with genetic variability.



In the model without genetic intraspecific variability, the unobserved variation in the environment results in an important “individual effect”. The performances of the two species can intersect, which means that the competitive hierarchy among the two species can be locally reversed depending on the micro-environmental variation, offering opportunities for the coexistence of the two species in a variable environment. In that

sense, incorporating the individual level in statistical models can help to explain the coexistence of species through a better integration of niche multidimensionality in models.

We then analyse the spatial structure of individual growth.

To do so, we compute Moran's I test, and we plot the semivariograms of all individuals and of individuals of each species separately.

```
Growth_norep <- Growth[which(Growth$Time == 1), ] #we study the growth at time t0
Growth_1 <- Growth_norep[which(Growth_norep$Species == 1), ]
Growth_2 <- Growth_norep[which(Growth_norep$Species == 2), ]

# Growth_sp : we keep the growth without genetic IV
Growth_vario_all <- Growth_norep %>%
  dplyr::select(Growth_sp) %>%
  dplyr::rename(Growth = Growth_sp) %>%
  dplyr::mutate(X = as.numeric(c(Coord1_sp1, Coord1_sp2)),
    Y = as.numeric(c(Coord2_sp1, Coord2_sp2)))

Growth_vario_1 <- Growth_1 %>%
  dplyr::select(Growth_sp) %>%
  dplyr::rename(Growth = Growth_sp) %>%
  dplyr::mutate(X = as.numeric(Coord1_sp1), Y = as.numeric(Coord2_sp1))

Growth_vario_2 <- Growth_2 %>%
  dplyr::select(Growth_sp) %>%
  dplyr::rename(Growth = Growth_sp) %>%
  dplyr::mutate(X = as.numeric(Coord1_sp2), Y = as.numeric(Coord2_sp2))

# Compute distance matrices for Moran's I test
Growth_vario_dists_all <- as.matrix(dist(cbind(Growth_vario_all$X,
  Growth_vario_all$Y)))
Growth_vario_dists_all_inv <- 1/Growth_vario_dists_all
diag(Growth_vario_dists_all_inv) <- 0
Mor_all <- ape::Moran.I(Growth_vario_all$Growth, Growth_vario_dists_all_inv)

Growth_vario_dists_sp1 <- as.matrix(dist(cbind(Growth_vario_1$X,
  Growth_vario_1$Y)))
Growth_vario_dists_sp1_inv <- 1/Growth_vario_dists_sp1
diag(Growth_vario_dists_sp1_inv) <- 0
Mor_sp1 <- ape::Moran.I(Growth_vario_1$Growth, Growth_vario_dists_sp1_inv)

Growth_vario_dists_sp2 <- as.matrix(dist(cbind(Growth_vario_2$X,
  Growth_vario_2$Y)))
Growth_vario_dists_sp2_inv <- 1/Growth_vario_dists_sp2
diag(Growth_vario_dists_sp2_inv) <- 0
Mor_sp2 <- ape::Moran.I(Growth_vario_2$Growth, Growth_vario_dists_sp2_inv)
```

	Moran's I index	P-value
Species 1	6.7e-02	0e+00
Species 2	6.9e-02	0e+00
All individuals	4.3e-02	0e+00

These results show that individual growth is largely spatially autocorrelated. This is due to the spatial

autocorrelation in the environmental variables. In this simple, completely controlled experiment, this is natural. However, in a less controlled environment, detecting spatial autocorrelation in a response variable could be the sign of the spatial structure of the underlying environmental variables.

However, if genetics are spatially structured, this could also translate into spatial autocorrelation in growth.

Finally, we can visualise this spatial autocorrelation with semivariograms, but also control if the individual growth is more similar within conspecifics than heterospecifics.

```
# Prepare data for variogram computation
sp::coordinates(Growth_vario_all) = ~X + Y
sp::coordinates(Growth_vario_1) = ~X + Y
sp::coordinates(Growth_vario_2) = ~X + Y

Vario_all <- gstat::variogram(Growth ~ X + Y, data = Growth_vario_all,
  width = 0.5, cutoff = 500)
Vario_1 <- gstat::variogram(Growth ~ X + Y, data = Growth_vario_1,
  width = 0.5, cutoff = 500)
Vario_2 <- gstat::variogram(Growth ~ X + Y, data = Growth_vario_2,
  width = 0.5, cutoff = 500)

# Fit spherical variogram models
Vario_all.fit = gstat::fit.variogram(Vario_all, gstat::vgm("Sph"))
Vario1.fit = gstat::fit.variogram(Vario_1, gstat::vgm("Sph"))
Vario2.fit = gstat::fit.variogram(Vario_2, gstat::vgm("Sph"))

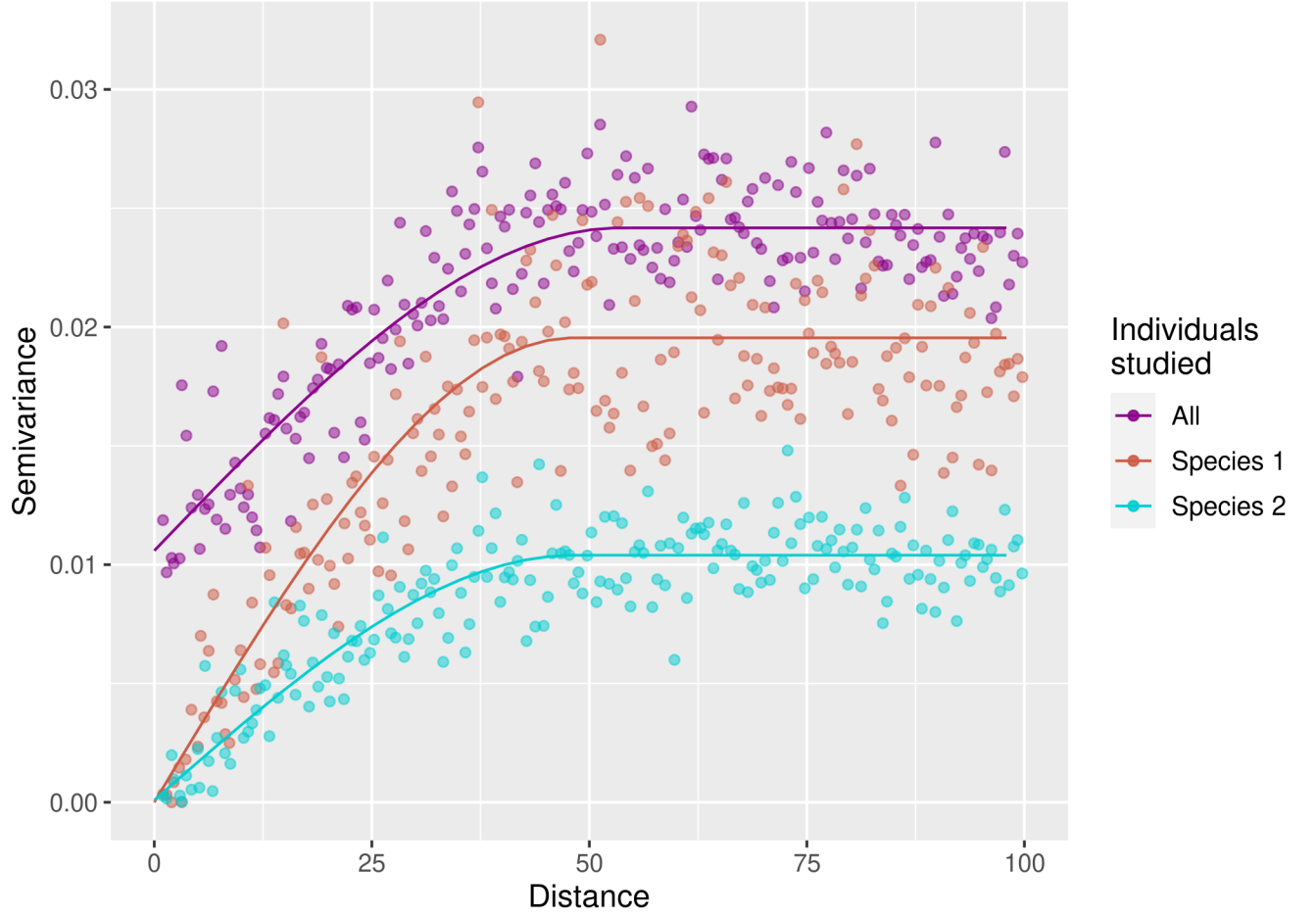
vgLine <- rbind(cbind(gstat::variogramLine(Vario_all.fit, maxdist = max(Vario_all$dist)),
  id = "All"), cbind(gstat::variogramLine(Vario1.fit, maxdist = max(Vario_1$dist)),
  id = "Species 1"), cbind(gstat::variogramLine(Vario2.fit,
  maxdist = max(Vario_2$dist)), id = "Species 2"))
```

```
## Warning: Removed 800 rows containing missing values (geom_point).
```

```
## Warning: Removed 800 rows containing missing values (geom_point).
```

```
## Warning: Removed 800 rows containing missing values (geom_point).
```

```
## Warning: Removed 480 row(s) containing missing values (geom_path).
```

As the semivariance between individuals of both species is higher than the semivariance between conspecifics, individual growth is more similar between conspecifics than heterospecifics. Considering growth as a proxy of fitness, and linking fitness to competition, we argue that in a Lotka-Volterra model, this would translate into $\alpha_{i,i} > \alpha_{i,j}$, the main condition for stable coexistence.

Therefore, stable coexistence is possible even with high intraspecific variability, especially when this variability is not intrinsic but due to environmental heterogeneity that is structured in space.