

Workflow for SI

Sort of pertubation theory

```
require(plotrix)
require(numDeriv)
require(plyr)
require(RColorBrewer)
```

First, let's source the function that executes simulations.

```
source('functions/coevoMutNet2Sites_array.R')
```

We start this workflow with a simple scenario: five species of each type, totaling 10 species, with a perfectly nested interaction structure:

```
n_row <- 5
n_col <- 5
n_sp <- n_row + n_col

mat <- matrix(0, n_row, n_col)
for(i in 1:n_row)
  for(j in 1:n_col)
    mat[i, j] <- ifelse(i >= j, 1, 0)

zeros_row <- matrix(0, n_row, n_row)
zeros_col <- matrix(0, n_col, n_col)

f <- rbind(cbind(zeros_row, mat), cbind(t(mat), zeros_col))

color2D.matplot(f, axes = FALSE, xlab = '', ylab = '', asp = 1)
```

An initial simulation scenario would be in the absence of either “abiotic” interactions and gene flow between communities. This scenario can be represented by setting the following parameters:

```
## environmental optimum, not used in this scenario
## but the function needs some values to run
theta_A <- rnorm(n_sp, 0, 1)
theta_B <- rnorm(n_sp, 0, 1)

h2 <- 0.1 # heritability

g = 0 # gene flow
```

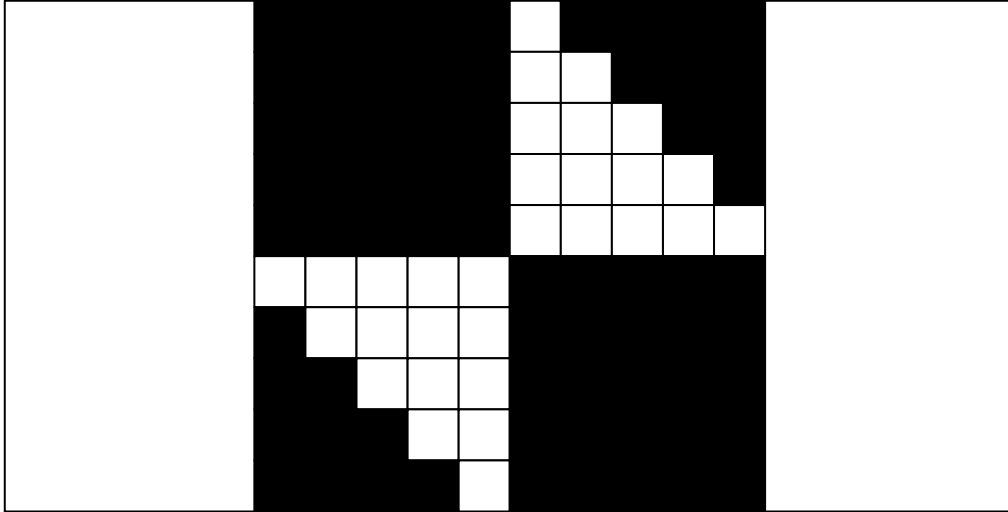


Figure 1: Simple adjacency matrix for five species with a perfectly nested structure. White cells represent ones, black cells represent zeroes.

```
p_A = 1 # hotness of A
p_B = 1 # hotness of B

alpha = 0.1
# in this scenario, changing alpha should not induce funky behavior

epsilon = 0.000001
t_max = 100000
```

In this scenario, we expect that every species should have the same value, corresponding to the mean of initial values at each community, weighted by the degree of each species.

```
diffA <- diffB <- c()

for(i in 1:1000)
{
  init_A <- rnorm(n_sp, 0, 1)
  init_B <- rnorm(n_sp, 0, 1)

  degree <- rowSums(f)

  wmeanA <- sum(init_A * degree) / sum(degree)
  wmeanB <- sum(init_B * degree) / sum(degree)

  Z <-
    CoevoMutNet2Sites_array(n_sp,
                             f = f,
```

```

g = rep(g, n_sp),
h = rep(h2, n_sp),
alpha = alpha,
theta_A, theta_B,
init_A, init_B,
m_A = rep(p_A, n_sp),
m_B = rep(p_B, n_sp),
epsilon = 1e-6, t_max)

final <- Z [dim(Z)[1], , ]

wm_fin_A <- sum(degree * final [, 'A']) / sum(degree)
wm_fin_B <- sum(degree * final [, 'B']) / sum(degree)

diffA[i] <- wm_fin_A - wmeanA
diffB[i] <- wm_fin_B - wmeanB
}

```

So, the distribution of differences between weighted means for initial and final values should be centered at zero:

```

par(mfrow = c(1, 2))
hist(diffA, main = '', xlab = 'Site A')
hist(diffB, main = '', xlab = 'Site B')

```

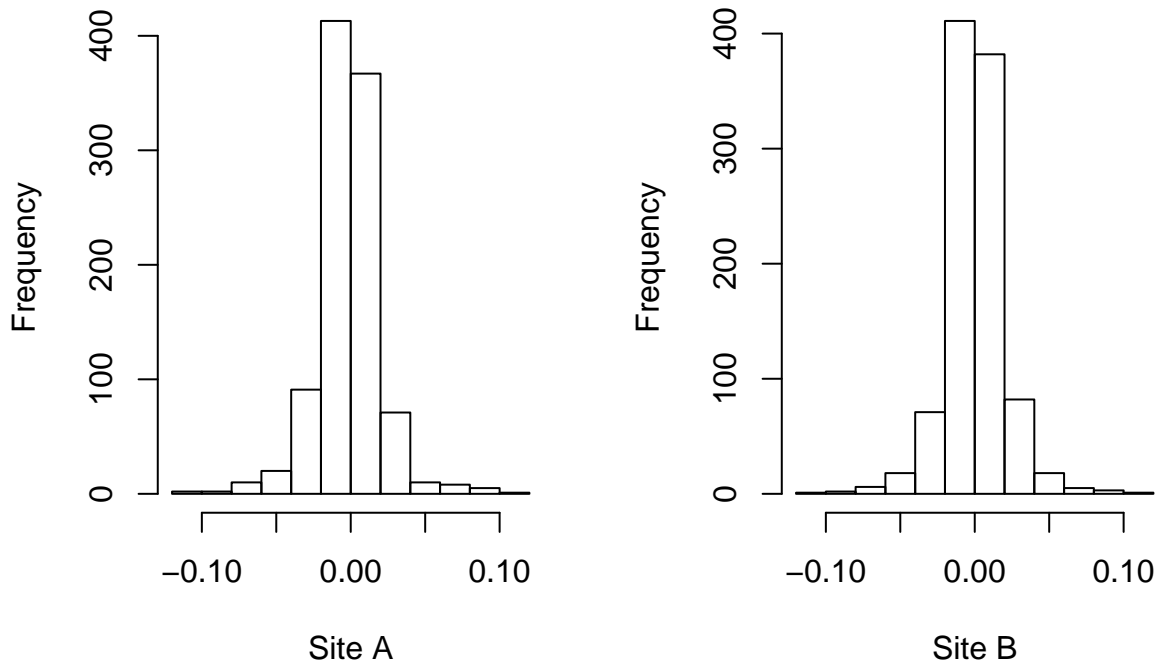


Figure 2: Distribution of differences between weighted averages for initial conditions and weighted averages for final values after simulation.

Cool.