Workflow for SI

Sort of pertubation theory

```
require(doMC)
registerDoMC(cores = 3)
require(plotrix)
require(dplyr)
require(magrittr)
require(ggplot2)
require(numDeriv)
require(plyr)
require(RColorBrewer)
```

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

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

Simple Scenario

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:

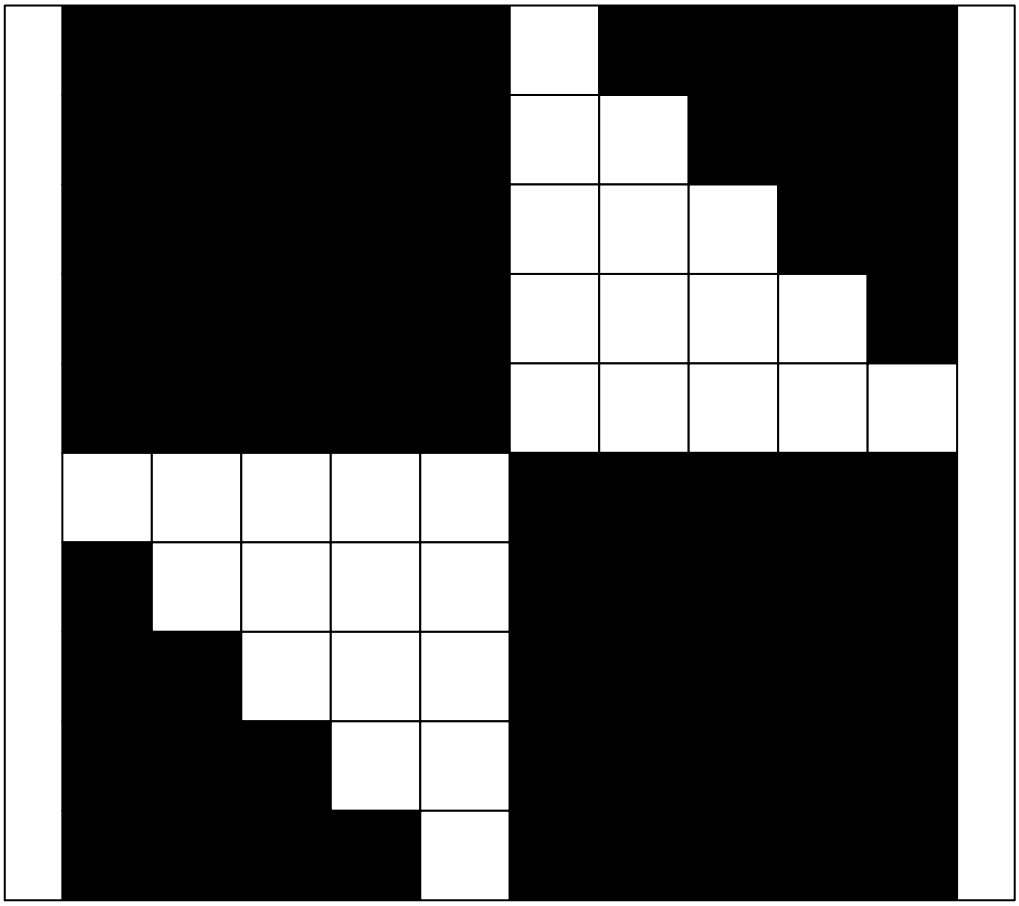

Figure 1: Simple adjacency matrix for five species with a perfectly nested structure. White cells represent ones, black cells represent zeroes.

```
## 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
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()

## check if there is no variance in final values
varA <- varB <- 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], , ]

        varA[i] <- var(final [, 'A'])
        varB[i] <- var(final [, 'B'])

        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')
```

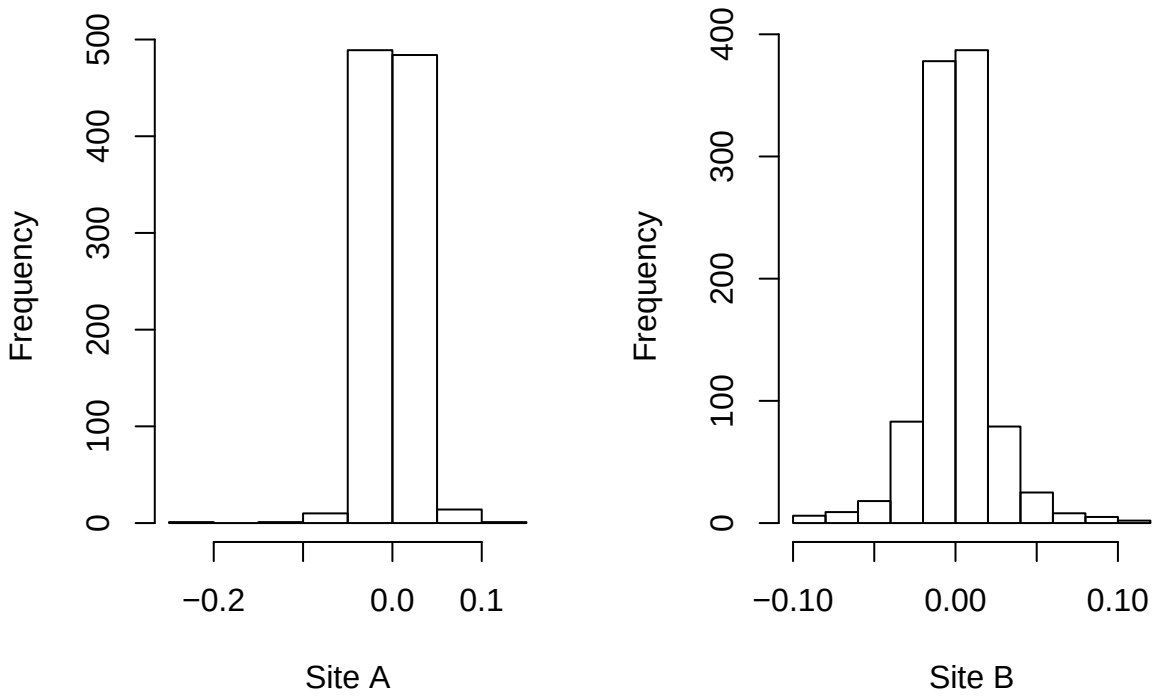

Figure 2: Distribution of differences between weighted averages for initial conditions and weighted averages for final values after simulation.

And all values of species in either sites should be the same, thus having zero variance:

```
summary(varA)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.440e-14 8.729e-11 4.957e-10 3.339e-10 5.270e-10 5.855e-10
```

```
summary(varB)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 4.970e-14 8.480e-11 4.984e-10 3.350e-10 5.241e-10 5.877e-10
```

Looks zero to me.

Perturbations

In this simple setting, it is clear that local populations would converge in their phenotypes, and such convergence would depend both on initial conditions and the underlying interaction network. Initial conditions of central species (hm, so maybe supergeneralists invading some community would strongly shift phenotypes of composing populations?). Local communities would thus exhibit perfect trait matching (or maybe not so perfect, but perfect *on average*).

Now, we focus on perturbations onto this initial scenario. First we introduce gene flow, and observing its effect on the same statistics obtained in the first scenario: the difference between weighted means for initial and final values in either communities, and variances within each community.

(gotta look at the spectrum)

```
g <- seq(0.001, 0.1, by = 0.001)

set.seed(1037)

scen2 <-
    aaply(1:100, 1, function(i)
    {
        aaply(1:length(g), 1, function(j)
        {

            init_A <- rnorm(n_sp, 10, sqrt(10))
            init_B <- rnorm(n_sp, 20, sqrt(10))

            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[j], 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], , ]

            varA <- var(final [, 'A'])
            varB <- var(final [, 'B'])

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

            diffA <- wm_fin_A - wmeanA
            diffB <- wm_fin_B - wmeanB

           c(diffA, diffB, varA, varB)
    })}, .parallel = TRUE)
```

```
names(dimnames(scen2)) [2] <- 'geneflow'
dimnames(scen2) [[2]] <- as.character(g)
dimnames(scen2) [[3]] <- c('diffA', 'diffB', 'varA', 'varB')

adply(scen2, 2) %>%
    ggplot(.) +
    geom_violin(aes(x = geneflow, y = (diffA + diffB))) +
    theme(axis.text.x = element_text(angle = 90, size = 2))
```

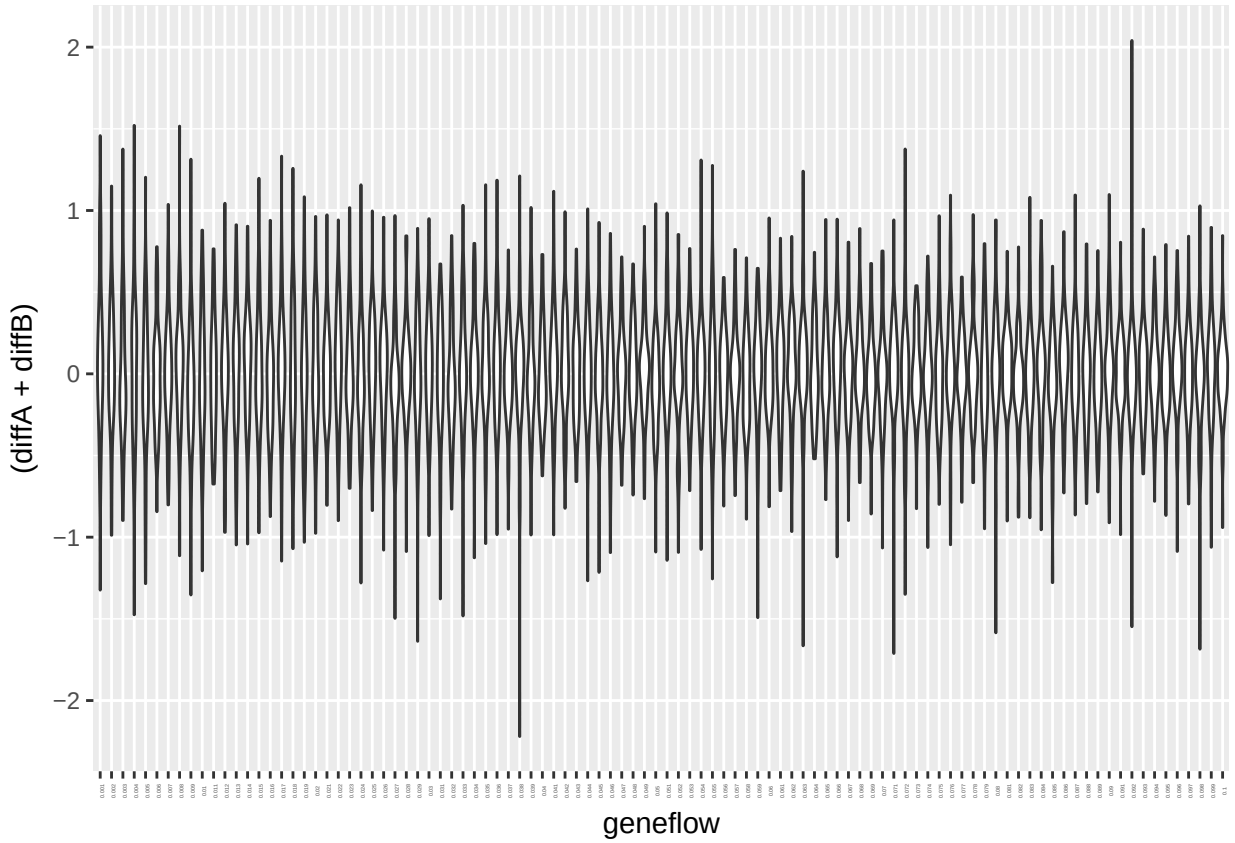

Figure 3: Distribution of the sum of differences between weighted means for initial and final values for both sites, for values of gene flow between 0 and 0.1