ELECTRONIC APPENDIX

This is the Electronic Appendix to the article

Experimental evidence rejects pairwise modelling approach to coexistence in plant communities

by

Carsten F. Dormann & Stephen H. Roxburgh

Proc. R. Soc. B (doi:10.1098/rspb.2005.3066)

Electronic supplement:

R-code for the analysis presented in this study

```
#Using a LV equation, fit an ODE to the data using information on initial dry weight, final
dry weight (in single density) and
#carrying capacity estimate (the maximum, rather than the average, pot biomass in single
density).
#Step size for the ODE is chosen as a day, i.e. 365 steps from initial to final.

#1. fit a logistic growth model to the monoculture data, assuming that N final = carrying
capacity K

## load relevant packages
library(odesolve)

## functions used during this procedure:
LVsingle <- function(t,x,parms){# function for monoculture biomass, i.e. logistic growth
   N <- x[1]  # initial population size
   K <- parms[2]
   r <- parms[1]
   dN <- r*N*(1- N/K)
   res <- c(dN)
   list(res)
}

LVtwospecies <- function(t,x,parms){ # function for two-species competition
   N1 <- x[1]  # initial population size
   N2 <- x[2]
   K1 <- parms[1]
   K2 <- parms[2]
   r1 <- parms[3]
   r2 <- parms[4]
   alpha12 <- parms[5]
   alpha21 <- parms[6]
   dN1 <- r1*N1*(K1 - N1 - alpha12*N2)
   dN2 <- r2*N2*(K2 - N2 - alpha21*N1)
   res <- c(dN1, dN2)
   list(res)
}

LVthreespecies <- function(t,x,parms){
   N1 <- x[1]  # initial population size
   N2 <- x[2]
   N3 <- x[3]
   K1 <- parms[1]
   K2 <- parms[2]
   K3 <- parms[3]
   r1 <- parms[4]
   r2 <- parms[5]
   r3 <- parms[6]
   alpha12 <- parms[7]
   alpha21 <- parms[8]
   alpha13 <- parms[9]
   alpha31 <- parms[10]
   alpha23 <- parms[11]
   alpha32 <- parms[12]
   dN1 <- r1*N1*(K1 - N1 - alpha12*N2 - alpha13*N3)
   dN2 <- r2*N2*(K2 - N2 - alpha21*N1 - alpha23*N3)
   dN3 <- r3*N3*(K3 - N3 - alpha31*N1 - alpha32*N2)
   res <- c(dN1, dN2, dN3)
   list(res)
}

# minimising functions
rmin.fun<-function(x){ #internal function for finding the minimum r that agrees with Nfinal
(monocultures)
   min(x[which(lsoda(N0, times, LVsingle, c(x, K=K))[length(times),2]-Nfinal>=0)])  }

rmin2.fun<-function(a){ # internal function for the two species competition analysis
   i <- a[1]
   j <- a[2]
   sum((lsoda(c(Nini1,Nini2), times, LVtwospecies, c(K1=K1, K2=K2, r1=r1, r2=r2,
alpha12=i, alpha21=j))[length(times),2:3]-c(N1final, N2final))^2)
}

# set step size (although calculation are independent of these; they are just to tell the model
when to output data
times  <- 0:52  # vector of timesteps


## GET THE VARIOUS DATA IN
rrd <- read.table("ROX_initial_final_monoculture_biomass.txt", head=T) #rrd=rox real data
attach(rrd)
names(rrd)
#rrd.N0<-rrd[species=="Ac", "initial"]
#rrd.Nfinal<-rrd[species=="Ac", "final"]
species.names <- sort(unique(rrd$species))

# initial biomass matrix (monocultures)
ini.weights <- matrix(ncol=7, nrow=10)
colnames(ini.weights) <- species.names
for (i in 1:7){
   ini.weights[,i] <- rrd[(species==species.names[i] & SD=="S"), "initial"]    }
# final biomass matrix (single-density monocultures)
fin.weights <- matrix(ncol=7, nrow=10)
colnames(fin.weights) <- species.names
for (i in 1:7){
   fin.weights[,i] <- rrd[(species==species.names[i] & SD=="S"), "final"]  }
# carrying capacity vector (double-density monocultures)
K.mat <- matrix(ncol=7, nrow=10)
colnames(K.mat) <- species.names
```

```
for (j in 1:7){
   K.mat[,j] <- if(mean(rrd[(species==species.names[j] & SD=="D"), "final"]) >
mean(rrd[(species==species.names[j] & SD=="S"), "final"]))
                (rrd[(species==species.names[j] & SD=="D"), "final"])
               else (rrd[(species==species.names[j] & SD=="S"), "final"])
#K.mat[,j] <- rrd$final[species==species.names[j]]
}
detach(rrd)

## load data on pairwise-competition experiments
pairwise <- read.table("RoxburghData.txt", head=T)

# initial biomass for three-species competition
triple <- read.table("ROXtripledata.txt", head=T)


## STEP 1.
#find smallest r given N0 (initial biomass) and K values

r.vec <- 1:7
names(r.vec)<-species.names
for (k in 1:7){ # calculates r for each monoculture (single density)
   K <- colMeans(fin.weights)[k]#colMeans(K.mat)[k]
   N0 <- colMeans(ini.weights)[k]
   Nfinal <- colMeans(fin.weights)[k]
   r.vec[k] <- suppressWarnings(optimise(rmin.fun, c(0,1))$minimum)
}



## STEP 2.
## fit non-linear least-square model according to equation (3):


bigfun<-function(x){
   alpha.coef.mat <- matrix(ncol=3, nrow=3) #comp. effect of species "col" on species "row",
i.e. species 1 on 2 would be in cell [2,1]
   colnames(alpha.coef.mat) <- c("species", "alphaA.BC", "se.alpha")
   triple.names <- x
   m<-1;n<-2;o<-3
   KA <- K.mat[,triple.names[m]]
   KB <- K.mat[,triple.names[n]]
   KC <- K.mat[,triple.names[o]]
   wAB <- pairwise$biomass[pairwise$species==triple.names[m] &
pairwise$competitor==triple.names[n]]
   wBA <- pairwise$biomass[pairwise$species==triple.names[n] &
pairwise$competitor==triple.names[m]]
   wAC <- pairwise$biomass[pairwise$species==triple.names[m] &
pairwise$competitor==triple.names[o]]
   wCA <- pairwise$biomass[pairwise$species==triple.names[o] &
pairwise$competitor==triple.names[m]]
```

```
   wBC <- pairwise$biomass[pairwise$species==triple.names[n] &
pairwise$competitor==triple.names[o]]
   wCB <- pairwise$biomass[pairwise$species==triple.names[o] &
pairwise$competitor==triple.names[n]]
   wA.BC <- triple$final[triple$species1==triple.names[m] &
triple$species2==triple.names[n] & triple$species3==triple.names[o]]
   wB.AC <- triple$final[triple$species1==triple.names[n] &
triple$species2==triple.names[m] & triple$species3==triple.names[o]]
   wC.AB <- triple$final[triple$species1==triple.names[o] &
triple$species2==triple.names[m] & triple$species3==triple.names[n]]
     #nls(wBA~(KA-wAB)/alphaAB, start=list(alphaAB=1))
   alpha.coef.mat[1,2]<-summary(nls(wA.BC ~ KA- alphaA.BC * ( (wB.AC/wBA)*(KA-
wAB) + (wC.AB/wCA)*(KA-wAC) ), start=list(alphaA.BC=1) ))$parameters[1]
      alpha.coef.mat[1,3]<-summary(nls(wA.BC ~ KA- alphaA.BC * ( (wB.AC/wBA)*(KA-
wAB) + (wC.AB/wCA)*(KA-wAC) ), start=list(alphaA.BC=1) ))$parameters[2]
      alpha.coef.mat[1,1]<-triple.names[m]
   alpha.coef.mat[2,2]<-summary(nls(wB.AC ~ KB- alphaB.AC * ( (wA.BC/wAB)*(KB-
wBA) + (wC.AB/wCB)*(KB-wBC) ), start=list(alphaB.AC=1) ))$parameters[1]
      alpha.coef.mat[2,3]<-summary(nls(wB.AC ~ KB- alphaB.AC * ( (wA.BC/wAB)*(KB-
wBA) + (wC.AB/wCB)*(KB-wBC) ), start=list(alphaB.AC=1) ))$parameters[2]
      alpha.coef.mat[2,1]<-triple.names[n]
   alpha.coef.mat[3,2]<-summary(nls(wC.AB ~ KC- alphaC.AB * ( (wA.BC/wAC)*(KC-
wCA) + (wB.AC/wBC)*(KC-wCB) ), start=list(alphaC.AB=1) ))$parameters[1]
      alpha.coef.mat[3,3]<-summary(nls(wC.AB ~ KC- alphaC.AB * ( (wA.BC/wAC)*(KC-
wCA) + (wB.AC/wBC)*(KC-wCB) ), start=list(alphaC.AB=1) ))$parameters[2]
      alpha.coef.mat[3,1]<-triple.names[o]
   return(alpha.coef.mat)
}



set1 <- bigfun(c("Fr", "Pv", "Tr"))
set2 <- bigfun(c("Ac", "Hh", "Hl"))
set3 <- bigfun(c("Ac", "Fr", "Hh"))
set4 <- bigfun(c("Hl", "Pv", "Rr"))
set5 <- bigfun(c("Ac", "Pv", "Tr"))
set6 <- bigfun(c("Fr", "Hh", "Rr"))

allsets<- rbind(set1, set2, set3, set4, set5, set6)
rownames(allsets)<-paste("set",as.character(rep(1:6, each=3)))

#t test on significant deviation of parameters from 1:
my.t.test <-function(x,a){
   res<-pt((as.real(x[a,2])-1)/as.real(x[a,3]), df=9)
#   paste("p < ", round(ifelse(res>0.5, 1-res, res),5), sep="")
   as.real(round(ifelse(res>0.5, 1-res, res),5))
}

my.t.test(allsets, 1)
sort(my.t.test(allsets, 1:18))
cbind(allsets,my.t.test(allsets, 1:18))
```

```
#----------------------------------------------------------------------------------------------------------
#--------------- now this is about stability analysis of the 3-species sets --------------------------
#----------------------------------------------------------------------------------------------------------

# I. Calculate predicted biomass on the basis of LV-approach
# what we need: comp.coef.matrix; r.vec; K.vec; Nini.pairs.mat
# calculate competition coefficients for all 49 species combinations ("1" for intraspecific)
# alphaAB <- (KA - biomassA)/biomassB (Istock 1977: American Naturalist 111, 279-287;
# cited from Schowalter 2000: Insect Ecology)

comp.coef.mat <- matrix(ncol=7, nrow=7) #comp. effect of species "col" on species "row",
i.e. species 1 on 2 would be in cell [2,1]
colnames(comp.coef.mat) <- rownames(comp.coef.mat) <- species.names
for (m in 1:7){
   for (n in 1:7){
   K1 <- colMeans(K.mat)[m]
   K2 <- colMeans(K.mat)[n]
   N1final <- mean(pairwise$biomass2[pairwise$species==species.names[m] &
pairwise$competitor==species.names[n]])
   N2final <- mean(pairwise$biomass2[pairwise$species==species.names[n] &
pairwise$competitor==species.names[m]])
      Nintra <- mean(pairwise$biomass[pairwise$species==species.names[m] &
pairwise$competitor==species.names[m]])/2
   comp.coef.mat[m,n] <- (K1-N1final)/N2final
   comp.coef.mat[n,m] <- (K2-N2final)/N1final
      comp.coef.mat[m,m]<- (K1-Nintra)/Nintra
}}


## make a table with N.equil and eigenvalues for LV and non-add

stabil.fun <- function(x){
   triple.names <- x[,1]
   alpha <- comp.coef.mat[triple.names, triple.names]
   diag(alpha)<-1
   r<-r.vec[triple.names]
   K <- colMeans(K.mat)[triple.names]
   N.equilLV <- solve(alpha, K)
   # construct the community (=Jacobian) matrix for LV model. diagonal entries are r/K; off-
diagonal are r/K*alpha
   comm.mat <- matrix(-c(r[1]/K[1], r[1]/K[1]*alpha[1,2], r[1]/K[1]*alpha[1,3],
           r[2]/K[2] *alpha[2,1], r[2]/K[2], r[2]/K[2]*alpha[2,3],
           r[3]/K[3]*alpha[3,1], r[3]/K[3]*alpha[3,2], r[3]/K[3]), ncol=3, byrow=T)
   S <- comm.mat * as.vector(N.equilLV)
   # now the non-add LV:
   alpha2 <- alpha*as.real(x[,2])
   diag(alpha2) <- 1
   N.equilNA <- solve(alpha2, K)
   comm.mat2 <- comm.mat*as.real(x[,2])
   diag(comm.mat2)<-diag(comm.mat)
   S2 <- comm.mat2 * as.vector(N.equilNA)
   print(as.real(round(c(N.equilLV, sort(Re(eigen(S)$values), decreasing=T), N.equilNA,
sort(Re(eigen(S2)$values), decreasing=T)),4)))
}

res <- sapply(list(set1, set2, set3, set4, set5, set6), stabil.fun)
stabil <- cbind(1:6, t(res))
colnames(stabil) <- c("set", "LV*A", "LV*B", "LV*C", "evLV1", "evLV2", "evLV3",
"NA*A", "NA*B", "NA*C", "evNA1", "evNA2", "evNA3")
stabil

# I. Calculate predicted biomass on the basis of LV-approach
# what we need: comp.coef.matrix; r.vec; K.vec; Nini.pairs.mat
# calculate competition coefficients for all 49 species combinations ("1" for intraspecific)
# alphaAB <- (KA - biomassA)/biomassB (Istock 1977: American Naturalist 111, 279-287;
# cited from Schowalter 2000: Insect Ecology)

comp.coef.mat <- matrix(ncol=7, nrow=7)
#comp. effect of species "col" on species "row", i.e. species 1 on 2 would be in cell [2,1]
colnames(comp.coef.mat) <- rownames(comp.coef.mat) <- species.names
for (m in 1:7){
   for (n in 1:7){
   K1 <- colMeans(K.mat)[m]
   K2 <- colMeans(K.mat)[n]
   N1final <- mean(pairwise$biomass2[pairwise$species==species.names[m] &
pairwise$competitor==species.names[n]])
   N2final <- mean(pairwise$biomass2[pairwise$species==species.names[n] &
pairwise$competitor==species.names[m]])
      Nintra <- mean(pairwise$biomass[pairwise$species==species.names[m] &
pairwise$competitor==species.names[m]])/2
   comp.coef.mat[m,n] <- (K1-N1final)/N2final
   comp.coef.mat[n,m] <- (K2-N2final)/N1final
      comp.coef.mat[m,m]<- (K1-Nintra)/Nintra
}}


## compare observed and predicted (according to referee's comments):
# r2 = sum((observed-expected)^2)/sum((observed-mean(observed))^2),
# aka Nash-Sutcliffe efficiency.

expectLV <- stabil[,2:4]
expectnonadd <- stabil[,8:10]
observed <- final.pred #set up the right type of matrix
final.fun <- function(x){
      triple.names <- x
      a1 <- mean(triple$final[triple$species1==triple.names[1] &
triple$species2==triple.names[2] & triple$species3==triple.names[3]])
      a2 <- mean(triple$final[triple$species1==triple.names[2] &
triple$species2==triple.names[1] & triple$species3==triple.names[3]])
```

```
       a3 <- mean(triple$final[triple$species1==triple.names[3] &
triple$species2==triple.names[1] & triple$species3==triple.names[2]])
       c(a1, a2, a3)
    }
observed[1,] <- final.fun(c("Fr", "Pv", "Tr"))
observed[2,] <- final.fun(c("Ac", "Hh", "Hl"))
observed[3,] <- final.fun(c("Ac", "Fr", "Hh"))
observed[4,] <- final.fun(c("Hl", "Pv", "Rr"))
observed[5,] <- final.fun(c("Ac", "Pv", "Tr"))
observed[6,] <- final.fun(c("Fr", "Hh", "Rr"))
# I know this is awkward, but I could be bothered to do it more elegantly, sorry.


r2LV <- 1-sum((observed-expectLV)^2)/sum((observed-mean(observed))^2) ## shows that
LV is extremely poor!
r2nonadd <- 1-sum((observed-expectnonadd)^2)/sum((observed-mean(observed))^2) ##
shows that nonadd version is very nice.



#-----------------------------------------------------------------------------------------------------------
#------------------- now this is about stability analysis of the 7-species sets          -----------------
#-----------------------------------------------------------------------------------------------------------

seven <- read.table("ROXsevendata.txt", head=T)
attach(seven)
seven.final.mat <- matrix(ncol=7, nrow=10)
   colnames(seven.final.mat) <- species.names
for (i in 1:7) seven.final.mat[,i]<-final[species==species.names[i]]
detach(seven)
seven.final.mat <- as.data.frame(seven.final.mat)
attach(seven.final.mat)
seven.coef.res<-matrix(ncol=3, nrow=7)
   colnames(seven.coef.res) <- c("coef", "SE", "p")
   rownames(seven.coef.res) <- species.names
seven.coef.res[1,]<-summary(nls(Ac ~ K.mat[,1]- betaA.BC*(alpha[1,2]*Fr + alpha[1,3]*Hh
+ alpha[1,4]*Hl + alpha[1,5]*Pv +
       alpha[1,6]*Rr+ alpha[1,7]*Tr) , start=list(betaA.BC=1) ))$parameter[c(1,2,4)]
seven.coef.res[2,]<-summary(nls(Fr ~ K.mat[,1]- betaA.BC*(alpha[2,1]*Ac+alpha[2,3]*Hh
+alpha[2,4]*Hl +alpha[2,5]*Pv +alpha[2,6]*Rr+
       alpha[2,7]*Tr), start=list(betaA.BC=1) ))$parameter[c(1,2,4)]
seven.coef.res[3,]<-summary(nls(Hh ~ K.mat[,1]- betaA.BC*(alpha[3,1]*Ac+alpha[3,2]*Fr
+alpha[3,4]*Hl +alpha[3,5]*Pv +alpha[3,6]*Rr+
       alpha[3,7]*Tr), start=list(betaA.BC=1) ))$parameter[c(1,2,4)]
seven.coef.res[4,]<-summary(nls(Hl ~ K.mat[,1]- betaA.BC*(alpha[4,1]*Ac+alpha[4,2]*Fr
+alpha[4,3]*Hh +alpha[4,5]*Pv +alpha[4,6]*Rr+
       alpha[4,7]*Tr), start=list(betaA.BC=1) ))$parameter[c(1,2,4)]
seven.coef.res[5,]<-summary(nls(Pv ~ K.mat[,1]- betaA.BC*(alpha[5,1]*Ac+alpha[5,2]*Fr
+alpha[5,3]*Hh +alpha[5,4]*Hl +alpha[5,6]*Rr+
       alpha[5,7]*Tr), start=list(betaA.BC=1) ))$parameter[c(1,2,4)]
seven.coef.res[6,]<-summary(nls(Rr ~ K.mat[,1]- betaA.BC*(alpha[6,1]*Ac+alpha[6,2]*Fr
+alpha[6,3]*Hh +alpha[6,4]*Hl +alpha[6,6]*Pv+
        alpha[6,7]*Tr), start=list(betaA.BC=1) ))$parameter[c(1,2,4)]
seven.coef.res[7,]<-summary(nls(Tr ~ K.mat[,1]- betaA.BC*(alpha[7,1]*Ac+alpha[7,2]*Fr
+alpha[7,3]*Hh +alpha[7,4]*Hl +alpha[7,5]*Pv+
        alpha[7,6]*Rr), start=list(betaA.BC=1) ))$parameter[c(1,2,4)]


K <- colMeans(K.mat)
r<-r.vec
alpha <- comp.coef.mat
alpha2 <- alpha*as.real(seven.coef.res[,1])#the non-add competition matrix
diag(alpha2) <- diag(alpha)

N.equilLV7 <- solve(alpha, colMeans(K.mat))
N.equilNA7 <- solve(alpha2, colMeans(K.mat))

comm.mat7 <- matrix(-c(#the community matrix for LV
            r[1]/K[1], r[1]/K[1]*alpha[1,2], r[1]/K[1]*alpha[1,3], r[1]/K[1]*alpha[1,4],
r[1]/K[1]*alpha[1,5], r[1]/K[1]*alpha[1,6], r[1]/K[1]*alpha[1,7],
            r[2]/K[2] *alpha[2,1], r[2]/K[2], r[2]/K[2]*alpha[2,3], r[2]/K[2]*alpha[2,4],
r[2]/K[2]*alpha[2,5], r[2]/K[2]*alpha[2,3], r[2]/K[2]*alpha[2,7],
            r[3]/K[3]*alpha[3,1], r[3]/K[3]*alpha[3,2], r[3]/K[3], r[3]/K[3]*alpha[3,4],
r[3]/K[3]*alpha[3,5],r[3]/K[3]*alpha[3,6], r[3]/K[3]*alpha[3,7],
            r[4]/K[4]*alpha[4,1], r[4]/K[4]*alpha[4,2], r[4]/K[4]*alpha[4,3],
r[4]/K[4],r[4]/K[4]*alpha[4,5],r[4]/K[4]*alpha[4,6], r[4]/K[4]*alpha[4,7],
            r[5]/K[5]*alpha[5,1], r[5]/K[5]*alpha[5,2], r[5]/K[5]*alpha[5,3],
r[5]/K[5]*alpha[5,4],r[5]/K[5], r[5]/K[5]*alpha[5,6], r[5]/K[5]*alpha[5,7],
            r[6]/K[6]*alpha[6,1], r[6]/K[6]*alpha[6,2], r[6]/K[6]*alpha[6,3],
r[6]/K[6]*alpha[6,4],r[6]/K[6]*alpha[6,5], r[6]/K[6], r[6]/K[6]*alpha[6,7],
            r[7]/K[7]*alpha[7,1], r[7]/K[7]*alpha[7,2], r[7]/K[7]*alpha[7,3],
r[7]/K[7]*alpha[7,4],r[7]/K[7]*alpha[7,5], r[7]/K[7]*alpha[7,6], r[7]/K[7],
            ), ncol=7, byrow=T)

comm.mat7.2 <- comm.mat7 * seven.coef.res[,1]
diag(comm.mat7.2) <- diag(comm.mat7)

S7 <- comm.mat7 * as.vector(N.equilLV7)
S7.2 <- comm.mat7.2 * as.vector(N.equilNA7)

sort(Re(eigen(S7)$values), decreasing=T)
sort(Re(eigen(S7.2)$values), decreasing=T)


## compare observed and predicted (according to referee's comments):
# r2 = sum((observed-expected)^2)/sum((observed-mean(observed))^2),
# aka Nash-Sutcliffe efficiency.

expectLV7 <- N.equilLV7
expectnonadd7 <- N.equilNA7
obs7 <- c(2.626, .378, .118, 6.639, .729, 2.746, 7.696)
```

```
r2LV7 <- 1-sum((obs7-expectLV7)^2)/sum((obs7-mean(obs7))^2) ## shows that LV is
extremely poor!
r2nonadd7 <- 1-sum((obs7-expectnonadd7)^2)/sum((obs7-mean(obs7))^2) ## shows that
nonadd version is even worse.


#----------------------------------------------------------------------------------------------------------
#                              modelling the stability -species numbers boundary
#----------------------------------------------------------------------------------------------------------


eigen.fun <- function(x,y){#calculates eigenvalues for given number of species (x) and mean
comp intensity (y)
   set.seed(2)
   N <- x
   res <- as.data.frame(matrix(ncol=4, nrow=10))
      for (i in 1:10){
         alpha <- matrix(runif(N^2, y-.1, y+.1), ncol=N, nrow=N)
         alpha.mean<-mean(alpha)
         diag(alpha) <- 1
         res[i,2] <- min(Re(eigen(alpha)$values))
         res[i,1] <- N
         res[i,3] <- ifelse(res[i,2]>0, "black", "grey")
         res[i,4] <- alpha.mean #mean(alpha)#
      }
      mean(res[,2])
}

numbers <- 2:20
comp.means <- seq(0.1, 1.9, 0.1)
lambda <- matrix(nrow=length(numbers), ncol=length(comp.means))
for (j in 1:length(comp.means)){
   for (i in 1:length(numbers)){
         lambda[i,j]<- -eigen.fun(numbers[i], comp.means[j])
   }
}
contour(numbers, comp.means, lambda, xlab="number of species", ylab="mean competition
coefficient (off-diagonal)", cex.lab=1.5, labcex=1.5)
arrows(3, 1.1, 3, .935, lwd=3, length=0.2)
arrows(7, 1, 7, 1.3, lwd=3, length=0.2)
text(3, 1.05, "3", pos=4, cex=2)
text(7, 1.05, "7", pos=4, cex=2)


## use values for alphas drawn from the same distribution as in the experiment
eigen.fun <- function(x){#calculates eigenvalues for given number of species (x) and mean
comp intensity (y)
   set.seed(2)
   N <- x
   res <- as.data.frame(matrix(ncol=4, nrow=10))
      for (i in 1:10){
            alpha <- matrix(sample(alpha, N^2,replace=T), ncol=N, nrow=N)
            alpha.mean<-mean(alpha)
            diag(alpha) <- 1
            res[i,2] <- min(Re(eigen(alpha)$values))
            res[i,1] <- N
            res[i,3] <- ifelse(res[i,2]>0, "black", "grey")
            res[i,4] <- alpha.mean #mean(alpha)#
        }
        mean(res[,2])
}

numbers <- 1:20
lambda <- matrix(nrow=length(numbers), ncol=length(comp.means))
    for (i in 1:length(numbers)){
            lambda[i,j]<- eigen.fun(numbers[i])
    }

contour(numbers, comp.means, lambda, xlab="number of species", ylab="mean competition
coefficient (off-diagonal)", cex.lab=1.5, labcex=1.5)
```