Appendix S-B

R code for simulations

The procedure for simulations and the code are illustrated in what follows.

Input needed:

1- tab.txt (the community matrix with values -1, 0, +1 has to be created in .txt format and called tab.txt).
2- names2.txt (this is a row vector that contains labels for the variables. They have to be the same used in the community matrix, namely the column heads, and in the same order as they appear along the rows or columns of the community matrix)

Functions:

Once in R space launch:
> source ("com_mat2.R")

Script: com_mat2.R

```
#marcus<-matrix(c(-1,1,-1,0),nrow=2,byrow=TRUE)
#marcus<-matrix(c(-1,0,-1,0,0,-1,-1,-1,1,1,0,-1,0,1,0,0),nrow=4,byrow=TRUE)
#marcus<-matrix(c(-1,1,0,0,0,0,-1,1,0,-1,0,0,-1,1,0,1,0,0,-1,0,0,0,0,1,-1), nrow=5, byrow=T)

names<-scan("names2.txt", what=character(), sep="," )

marcus<-read.table("tab.txt", col.names=names, row.names=names, sep="," )

#colnames(marcus)<-names
#rownames(marcus)<-names

print("community matrix")
print(marcus)

ll<-length(marcus)
print("ll")
print(ll)

#library("Rgraphviz")
#rEG<-new("graphNEL", nodes=c(names), edgemode="directed")
# plot(rEG)

#for (i in 1:ll)
#  for (j in 1:ll)
#  { if (i==j) (rEG<-addEdge(names(marcus)[i], names(marcus)[i], rEG, 1))
#  else if ((i!=j) & (marcus[i,j]==1)) (rEG<-addEdge(names(marcus)[i], names(marcus)[j], rEG, 1))
#  else if ((i!=j) & (marcus[i,j]==-1)) (rEG<-addEdge(names(marcus)[i], names(marcus)[j], rEG, 1))
#  }
#  plot(rEG, recipEdges = "distinct")

library("LoopAnalyst")
mat<-scan("tab.txt", what=character(), sep=",")
mat_v<-as.vector(mat)
mat_c<-matrix(c(mat_v), nrow=ll, byrow=T)
```

```
mat_tc<-t(mat_c)
colnames(mat_tc)<-names
rownames(mat_tc)<-names
graph.cm(mat_tc, file="mat_tc.dot")


#library("network")
#g<-network(marcus, direct=T, hyper=F, loops=T)
#plot(g, usearrow=T, arrowhead.cex=2, loop.cex=3, vertex.cex=1, edge.col=4,
#plot.network(g, usearrow=T, arrowhead.cex=2, loop.cex=3, vertex.cex=1, edge.col=4,
#   vertex.col=1, label=network.vertex.names(g), displaylabels=T, boxed.labels=F, label.lwd=3, label.pos=0)
```

Output is the community matrix

Launch:
>source ("func_LOOP_sugg2.R")

Script: func_LOOP_sugg2.R

```
## Community matrix in Levins' notation###

library("MASS")
Loop <- function(marcus) {

        ### initializing count###
#       print ("WARNING!!! MASS PACKAGE NEEDED")

        print("community matrix")
        print(marcus)

        names<-scan("names2.txt", what=character(), sep="," )

        k<-1
        m<-0
    h<-0
        ##########################################
        #community matrix as sign matrix: lev #
        ##########################################
        ####MATRIX IS: a11, a21, a31,...ect. where
        # aij=dfi/dxj change in the growth function of i due to j!#############
        #1°es. Predator-Prey

                lev<-t(marcus)
                Det_m<-det(lev)

          print("Determinant")
          print(Det_m)

        dl<-sqrt(length(lev))
        dl2<-dl^2
          print("dl")
           print(dl)
```

```
        #  print(dl^2)

        nacc<-as.vector(k, mode="integer")
        nacc[1]=0

        n_p<-as.vector(m, mode="integer")
        n_m<-as.vector(m, mode="integer")
        n_oo<-as.vector(m, mode="integer")

        for (m in 1:dl2) {(n_p[m]=0) & (n_m[m]=0) & (n_oo[m]=0)}

        n_plus<-as.vector(n_p, mode="integer")
        n_min<-as.vector(n_m, mode="integer")
        n_o<-as.vector(n_oo, mode="integer")

######### NUMBER OF RUNS
        # ntent<-(length(lev)*100)
        # ntent<-(length(lev)*500)
        ntent<-(length(lev)*1000)
        # ntent<-(length(lev)*5000)
        # ntent<-(length(lev)*10000)

      print(ntent)
#########

        for (k in 1:ntent){

        ##########################
        #random matrix: casuale #
        ##########################

        casuale<-matrix(rep(0,dl2),nrow=dl)

###### RANDOM MATRIX GENERATION  (NAME IS: casuale) in [1e-6,1]
        for (i in 1:dl)
             for(j in 1:dl) casuale[i,j]<-runif(n=1,min=1e-6,max=1)

        #  print(casuale)

        ######################################################
        #weighted matrix (on degree tot for each variable) #
        ######################################################

        num=0

        ww<-matrix(rep(NA,dl2),nrow=dl)

          for(i in 1:dl)
            for(j in 1:dl){ww[i,j]<-(lev[i,j]*casuale[i,j])
                          }

        det_ww<-round(det(ww), digits=6)


        eig_vre<-round(Re(eigen(ww)$values), digits=6)
        eig_vim<-round(Im(eigen(ww)$values), digits=6)
```

```
        for (y in 1:dl) {if (eig_vre[y]<0) (num=num+1)}

        ##########A MINUS SIGN IS INSERTED DURING MATRIX INVERSION TO TAKE INTO
        ACCOUNT THE SIGN OF COEFF b=-dfi/dc  #########

        if ((num-dl)==0) {(nacc[k]=nacc[k]+1) & (inv_ww<-(-round(ginv(ww), digits=6))) & (vector<-
        as.vector(inv_ww))
                        for (m in 1:dl2)
                          {if (vector[m]>0) (n_plus[m]=n_plus[m]+1)
                                  else if (vector[m]<0) (n_min[m]=n_min[m]+1)
                                          else if (vector[m]==0) (n_o[m]=n_o[m]+1)}
                        }

        round(ginv(ww), digits=6)

        per_p<-round(matrix(c((n_plus*100)/nacc[k]), nrow=dl, byrow=T), digits=5)
        per_m<-round(matrix(c((n_min*100)/nacc[k]), nrow=dl, byrow=T), digits=5)
        per_o<-round(matrix(c((n_o*100)/nacc[k]), nrow=dl, byrow=T), digits=5)

        v_p<-as.vector(per_p)
        v_m<-as.vector(per_m)
        v_o<-as.vector(per_o)

        nacc[k+1]=nacc[k]


        ##########################################

          }


        ntot=nacc[k]

OUT <- as.list(rep(NA,5))


OUT[[1]]<-ntot
OUT[[2]]<-k
OUT[[3]]<-per_p
OUT[[4]]<-per_m
OUT[[5]]<-per_o

print("n° ACCEPTED MOVES")
print(OUT[[1]])
print("n° loops")
print(OUT[[2]])
print(" (%) + ")
colnames(OUT[[3]])<-names
rownames(OUT[[3]])<-names
print(OUT[[3]])
print(" (%) - ")
colnames(OUT[[4]])<-names
rownames(OUT[[4]])<-names
print(OUT[[4]])
print(" (%) 0 ")
```

```
  colnames(OUT[[5]])<-names
  rownames(OUT[[5]])<-names
  print(OUT[[5]])

tab<-as.vector(h, mode="any")

  for (h in 1:dl2) {
    if (v_o[h]==100) (tab[h]<-0)
     else if (v_p[h]>=75) (tab[h]<-"+")
       else if (v_p[h]<=25 & v_p[h]>=0) (tab[h]<-"-")
            else if (abs(v_p[h]-v_m[h])<=20) (tab[h]<-"0*")
             else if (25<v_p[h] & v_p[h]<40) (tab[h]<-"?-")
         else if (60<v_p[h] & v_p[h]<75) (tab[h]<-"?+")
               }


tab_m<-matrix(c(tab), nrow=dl, byrow=T)
tabella_predizioni<-t(tab_m)

### insert as external#########
colnames(tabella_predizioni)<-names
rownames(tabella_predizioni)<-names

 print("tabella_predizioni")
 print.noquote(tabella_predizioni)


  }
```

Launch: >Loop(marcus)

Output is: community matrix, determinant, accepted moves (i.e. stable matrices), matrices of +, - and 0 sign and percentages for each predictions, table of predictions.