Script for creating simulated community data, structured along one virtual ecological gradient

simul1.r

```
## Simulation of ecological data structured by one ecological gradient
## (based on the script of Jason Fridley (Fridley et al. 2007, Appendix S2),
modified by David Zelený (zeleny@sci.muni.cz)

# Basic setting:
totS<-300  #total species in simulation
Np<-500        #number of sample plots
gr.length <- 5000 # length of the gradient (arbitrary units)

# in case you want to add random species into the matrix, change following:
no.of.random.sp <- 0  # no of randomly added species to matrix
max.fr.random <- 0  # maximum frequency of randomly added species
# (random species are added with randomly generated frequency <0;
max.fr.random> into the matrix

#Creates coenoclines
#Creates survey designs (random, uniform, or biased exponentially along
gradient)
#Creates plot-spp matrix based on survey design
#Uses matrix in GS algorithm
#Outputs graphs
#Loops through "random" and "exponential (biased)" survey designs

#Choose niche distribution type: "normal", "skewed", or "random"
niche.type<-"random"

#This is beta function for generating niches
curve <- function(Ao,m,r,a,g)    {
    (Ao*((((x-m)/r)+(a/(a+g)))^a)*((1-(((x-
m)/r)+(a/(a+g))))^g))/(((a/(a+g))^a)*((1-(a/(a+g)))^g))
}
x <- seq(1,gr.length,by=1) # gradient (abscissa)

# species values for random niches
if(niche.type=="random") {
S<-totS                 #number of species
Ao<-rlnorm(S,2,1)            #amplitude vector (lognormal distribution)
m<-sample(seq(5:max(x)-5),S)      #location of optima
r<-runif(S,min=10,max=max(x))     #range along gradient (niche breadth)
a<-(runif(S,min=.1,max=4))       #shape parameter (alpha)
g<-(runif(S,min=.1,max=4))       #shape parameter (gamma)
}

# species values for normal niches
if(niche.type=="normal") {
S<-totS                 #number of species
Ao<-rlnorm(S,2,1)            #amplitude vector (lognormal distribution)
m<-sample(seq(5:max(x)-5),S)      #location of optima
```

```
r<-runif(S,min=10,max=max(x))     #range along gradient (niche breadth)
a<-rep(1.99,S)                #shape parameter (alpha)
g<-rep(1.99,S)                #shape parameter (gamma)
}

# species values for skewed niches
if(niche.type=="skewed") {
S<-totS                 #number of species
Ao<-rlnorm(S,2,1)            #amplitude vector (lognormal distribution)
m<-sample(seq(5:max(x)-5),S)      #location of optima
r<-runif(S,min=10,max=max(x))     #range along gradient (niche breadth)
#produce skews in either direction, randomly
    a1<-rep(1.99,S)
    g1<-rep(.25,S)
    samp<-sample(c(1:S),S/2,replace=FALSE)
    a1[samp]<-.25
    g1[samp]<-1.99
a<-a1               #shape parameter (alpha)
g<-g1               #shape parameter (gamma)
}

A <- matrix(0,nrow=gr.length,ncol=S) #response abundances
for(L in 1:S){
    A[,L] <- curve(Ao[L],m[L],r[L],a[L],g[L])
}

#Summary stats for each species
p.mat<-A
p.mat.NA<-is.na(p.mat)
p.mat[p.mat.NA]<-0
a.mat<-p.mat          #abundance matrix with zeros
p.mat[p.mat>0]<-1     #presence-absence matrix

ranges<-colSums(p.mat)        #total ranges (niche breadth) for each species
richness<-rowSums(p.mat)  #richness gradient


#------Summary plotting:
par(mfrow=c(2,2))

#Species distributions along gradient
#plot(x,A[,1],xlim=c(0,max(x)),ylim=c(0,max(Ao)),type="l",xlab="Gradient",ylab
="Abundance",cex.lab=1.7,cex.axis=1.5,lwd=2)
# for(L in 2:S) {
#     lines(x,A[,L])
# }
#lines(x,A[,51],lwd=3,col=2)
#lines(x,A[,52],lwd=3,col=3)

#Richness along gradient
#plot(x,richness)

#Rank-abundance of niche breadths
#plot(c(1:S),rev(sort(ranges)))
```

```r
##SAMPLING FROM GENERATED CURVES
#--------------------------------

sppmat <- data.frame(c(1:S),paste("S",c(1:S),sep=""))

#Equal sample intervals along gradient
#eq.sample.x <- trunc(seq(min(x),max(x)-1,length=Np))      #vector of sample
locations along gradient

#Random sample intervals along gradient
rand.sample.x <- sort(trunc(sample(c(2:max(x))-1,Np)))

#Biased samples at one end of gradient
    #based on exponential function; more samples at low end of gradient
# exp.x<-sort(rexp(Np,rate=20))
# exp.sample.x<-trunc(    exp.x/(max(exp.x))*(max(x)-1))
        #remove duplicate Np x values
#     while( length(unique(exp.sample.x))!=Np ) {
#     exp.sample.x[duplicated(exp.sample.x)]<-
exp.sample.x[duplicated(exp.sample.x)]+1
# }
# exp.sample.x<-rev(max(x)-exp.sample.x)    #switch to other side of gradient

#Distribution of # inds per local community along gradient
    #from beta distribution (see above)
density.curve<-curve(100,max(x)/2,max(x),.25,.25)

#Number of inds in each plot for each above sampling schemes
#draws.eq <- round(rnorm(Np,mean=density.curve[eq.sample.x],sd=1))
draws.rand <- round(rnorm(Np,mean=density.curve[rand.sample.x],sd=1))
#draws.exp <- round(rnorm(Np,mean=density.curve[exp.sample.x],sd=1))

#output data frames
#samp.out.eq <- matrix(0,nrow=Np,ncol=S)
samp.out.rand <- matrix(0,nrow=Np,ncol=S)
#samp.out.exp <- matrix(0,nrow=Np,ncol=S)

#Sampling for equal-sample-interval
#for(i in 1:Np) {
# samp.prob<-a.mat[eq.sample.x[i],]        #probabilities of sampling each
species in given location (based on rel abundance)
# tab.samp <- table(sample(c(1:S),size=draws.eq[i],prob=samp.prob,replace=T))
#tabulated vector of spp identities after choosing "draws" no. of individuals
# samp.out.eq[i,][as.numeric(names(tab.samp))] <- tab.samp
#}
#eq.pa <- samp.out.eq
#eq.pa[eq.pa>0] <- 1        #presence-absence version
#converted to 2-column list:
#spp.vec<-NULL
#plot.vec<-NULL
#for(i in 1:Np) {
# vec.true<-as.logical(eq.pa[i,])
# plot.vec<-c(plot.vec,rep(i,length=sum(eq.pa[i,])))
# spp.vec<-c(spp.vec,c(1:S)[vec.true])
#}
#out.eq <- data.frame(plot.vec,spp.vec)
```

```
#Sampling for random-sample-interval
for(i in 1:Np) {
    samp.prob<-a.mat[rand.sample.x[i],]        #probabilities of sampling each
species in given location (based on rel abundance)
    tab.samp <-
table(sample(c(1:S),size=draws.rand[i],prob=samp.prob,replace=T))
#tabulated vector of spp identities after choosing "draws" no. of individuals
    samp.out.rand[i,][as.numeric(names(tab.samp))] <- tab.samp
}
rand.pa <- samp.out.rand
colnames (rand.pa) <- paste ('spec_', 1:dim (rand.pa)[2], sep = '')
rand.pa[rand.pa>0] <- 1       #presence-absence version

if (no.of.random.sp > 0)
  {
  random.part <- matrix (0, ncol = no.of.random.sp, nrow = Np, dimnames = list
(NULL, paste ('rand_', 1:no.of.random.sp, sep = '')))
  for (i in seq (1, no.of.random.sp))  random.part[sample (c (1:Np), round
(runif (1)*max.fr.random)),i] <- 1
  simul.table <- cbind (rand.pa, random.part)
  } else simul.table <- rand.pa


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

## RESULTS
# m = vector of species optimas
# rand.pa = sample-species matrix
# rand.sample.x = position of sample locations
# simul.table = simulated data (if random species were added, they are
appended in the end of the table)
```

From:
https://anadat-r.davidzeleny.net/ - **Analysis of community ecology data in R**

Permanent link:
https://anadat-r.davidzeleny.net/doku.php/en:skripts:simul_1_gradient

Last update: **2017/10/11 20:36**