View metadata, citation and similar papers at core.ac.uk
brought to you by CORE

On-line Supplement Tab. 1

Structure of input file (ASCII txt):
first number: the number of rows (sample quadrats)
second number: number of columns (species)
subsequent numbers: presence/absence or frequency of species

```
10
4
1 1 0 1
1 0 1 1
1 1 0 0
1 1 1 0
1 0 1 1
1 1 0 1
1 1 1 1
1 1 0 0
1 1 0 0
1 0 1 1
```

Source code of R-script:

```
## Moving Split Window procedure for detecting discontinuities
## with Squared Euclidean Distance
##(base) package required!!!
## László Körmöczi 2008

## INPUT PARAMETERS MUST BE SET AS APPROPRIATE BEFORE RUNNING!!!
## SET WORKING DIRECTORY:
          setwd ("i:/Databases/MSW")
## SET INPUT FILE NAME:
          rawdataname <- "mf.txt"
## SET OUTPUT FILE NAME:
          printfile <- "mf-rshift-sed.txt"
## SET MAXIMUM HALFWINDOW SIZE:
          maxwin <- 10
## SET NUMBER OF RANDOMIZATIONS
          iter<-99
## ----------------------------------------

nsor <- scan(rawdataname, n=1)
noszlop <- scan(rawdataname, skip=1, n=1)
kiir  <- c('Size of input matrix: ' ,nsor, ' quadrats and ',noszlop, '
species')
kiir
```

```
rawdata <- read.table(rawdataname, skip = 2)
slawe <- matrix(0, nsor, noszlop)
resu <- matrix(NA, nsor, maxwin)
z_score <- matrix(NA, nsor, (maxwin+1))
resu_rnd <- matrix(NA, (nsor*iter), maxwin)
resu_rndZ <- matrix(NA, (nsor*iter), maxwin)
blank<-""

write ('Moving split window analysis', file=printfile)
write ('Squared Euclidean distance; random shift', file=printfile,
append=TRUE)
write (kiir, file=printfile, append=TRUE, sep = "\t")
write ('Input file:', file=printfile, append=TRUE)
write (rawdataname, file=printfile, append=TRUE)
write ('Number of randomizations:', file=printfile, append=TRUE)
write (iter, file=printfile, append=TRUE)
write (blank, file=printfile, append=TRUE)
write ('SED values for increasing halfwindow sizes', file=printfile,
append=TRUE)

for(winsize in 1:maxwin){
for(i in winsize:nsor){
for (j in 1:noszlop)
slawe[i,j]<-mean(rawdata[(i-winsize+1):i ,j])}

## DISTANCE COMPUTATION
{    i<-1
for (i in winsize:(nsor-winsize)) {
     sed <- 0
     for (j in 1:noszlop) sed <- sed + (slawe[i,j] -
slawe[(i+winsize),j])^2
     resu[i,winsize]<-sed}
}
}

write (resu, file=printfile, ncolumns=nsor, sep = "\t", append=TRUE)
write (blank, file=printfile, append=TRUE)

## RANDOM REFERENCE

doboz <- read.table(rawdataname, skip = 2)
for (k in 1:iter) {
for (j in 1:noszlop)  { shift<-as.integer((runif(1))*nsor)

     for (i in 1:nsor)
          {if (i+shift < nsor)
          rawdata[i+shift,j]=doboz[i,j]
          else
     rawdata[i+shift-nsor,j]=doboz[i,j]
          }
     }
for(winsize in 1:maxwin){
```

```
for(i in winsize:nsor){
for (j in 1:noszlop)
slawe[i,j]<-mean(rawdata[(i-winsize+1):i ,j])}

{     i<-1
for (i in winsize:(nsor-winsize)) {
      sed <- 0
      for (j in 1:noszlop) sed <- sed + (slawe[i,j] -
slawe[(i+winsize),j])^2
      resu_rnd[(i+(k-1)*nsor),winsize]<-sed}
}
}
}

for (j in 1:winsize) {
for (i in 1:(nsor*iter)) {
resu_rndZ[i,j]<-(resu_rnd[i,j]-(mean (resu_rnd[ ,j],
na.rm=TRUE)))/(sd (resu_rnd[ ,j],na.rm=TRUE))
}
}

write (blank, file=printfile, append=TRUE)
head<-c("win_size","mean","SD")
statiszt<-matrix( ,3,maxwin)
write (blank, file=printfile, append=TRUE)
write (head, file=printfile, ncolumns=3, append=TRUE, sep = "\t")
for(i in 1:maxwin) {
statiszt[1,i]<-i
statiszt[2,i]<- (mean (resu_rnd[ ,i],na.rm=TRUE))
statiszt[3,i]<- (sd (resu_rnd[ ,i],na.rm=TRUE))
}
write (statiszt, file=printfile, ncolumns=3, append=TRUE, sep = "\t")
write (blank, file=printfile, append=TRUE)

for(j in 1:maxwin) {
for (i in 1:nsor) {
z_score[i,j]<-  (resu[i,j]-(mean  (resu_rnd[  ,j],na.rm=TRUE)))/(sd
(resu_rnd[ ,j],na.rm=TRUE))
}
}
for (i in 1:nsor) z_score[i,(maxwin+1)]<-(mean(z_score[i,(1:maxwin)],
na.rm=TRUE))

write ('Z-scores', file=printfile, append=TRUE)
write (z_score[ ,(1:maxwin)], file=printfile, ncolumns=nsor, sep = "\t",
append=TRUE)
write (' ', file=printfile, append=TRUE)
write ('Z-score averages', file=printfile, append=TRUE)
write (z_score[ ,(maxwin+1)], file=printfile, ncolumns=nsor, sep = "\t",
append=TRUE)

plot (z_score[ ,(maxwin+1)], type="l")
```