

Marcon, E., Hérault, B., Baraloto, C. and Lang, G. 2012. The decomposition of Shannon's entropy and a confidence interval for beta diversity. – *Oikos* 121: 516–522.

## Appendix A1

The R (R Development Core Team 2010) code we wrote and used.

```
#####
# DECOMPOSITION OF SHANNON'S INDEX OF DIVERSITY
# B. Hérault, bruno.herault@ecofog.gf
# E. Marcon, eric.marcon@ecofog.gf
# Last updated: April 15, 2011
#####

# Initialization
rm(list=ls(all=TRUE))

#####
##### FUNCTIONS #####
# Halpha_i and Hbeta_i return a vector: alpha diversity in each
plot, contribution of each plot to beta diversity
# Halpha and Hbeta return total alpha and beta diversity,
# as a weighted average of plots
# Hgamma returns gamma diversity
# Unbiased estimators of diversity are according to
# Chao and Shen (2003)
# Parameter: dataM is a matrix, lines are species, rows are plots,
# values are numbers of individuals

# Shannon's alpha diversity
Halpha_i<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # p_si
  freqM<-dataM%*%diag(1/Nrecords.plots)
  # p_si log p_si
  dataAlpha<--log(freqM)*freqM
  # force 0log0=0
  dataAlpha[dataM==0]<-0
  # return the vector Halpha_i
  apply(dataAlpha, 2, sum)
}
```

```

HalpHa<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # weighted sum of HalpHa_i
  HalpHa_i(dataM)%*%Nrecords.plots/Nrecords
}
# Unbiased Shannon's alpha diversity
UBEHalpHa_i<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # Sample Coverage C_i
  dataM.ObservedOnce<-numeric()
  dataM.ObservedOnce<-(dataM==1)
  Singletons.plots<-apply(dataM.ObservedOnce, 2, sum)
  SampleCoverage<-1-Singletons.plots/Nrecords.plots
  # C_i.p_si
  freqM<-dataM%*%diag(1/Nrecords.plots)%*%diag(SampleCoverage)
  # C_i.p_si log C_i.p_si
  Nspecies<-length(dataM[,1])
  Nplots<-length(dataM[1,])
  dataAlpha<--freqM*log(freqM)/(1-(1-
freqM)^t(matrix(Nrecords.plots,nrow=Nplots,ncol=Nspecies)))
  # force 0log0=0
  dataAlpha[dataM==0]<-0
  # return the vector HalpHa_i
  apply(dataAlpha, 2, sum)
}
UBEHalpHa<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # weighted sum of HalpHa_i
  UBEHalpHa_i(dataM)%*%Nrecords.plots/Nrecords
}
# Shannon's beta diversity
Hbeta_i<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_s+
  Nrecords.species<-apply(dataM, 1, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # p_si
  freqM<-dataM%*%diag(1/Nrecords.plots)
  # p_si log (p_si/(y_s+/y_++))
  dataBeta<-freqM*log(diag(1/Nrecords.species)%*%freqM*Nrecords)
  # force 0log0=0
}

```

```

    dataBeta[dataM==0]<-0
    # return the vector Hbeta_i
    apply(dataBeta, 2, sum)
  }
Hbeta<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # weighted sum of Hbeta_i
  Hbeta_i(dataM)%*%Nrecords.plots/Nrecords
}
# Unbiased Shannon's beta diversity
UBEHbeta_i<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_s+
  Nrecords.species<-apply(dataM, 1, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # Sample Coverage Plots: C_i
  dataM.ObservedOnce<-numeric()
  dataM.Observed<-numeric()
  dataM.ObservedOnce<-(dataM==1)
  dataM.Observed<-(dataM>0)
  Singletons.plots<-apply(dataM.ObservedOnce, 2, sum)
  SampleCoverage.Plots<-1-Singletons.plots/Nrecords.plots
  Species.plots<-apply(dataM.Observed, 2, sum)
  # Sample Coverage All: C
  dataM.ObservedOnce<-(Nrecords.species==1)
  dataM.Observed<-(Nrecords.species>0)
  Singletons<-sum(dataM.ObservedOnce)
  SampleCoverage<-1-Singletons/Nrecords
  Species<-sum(dataM.Observed)
  # C_i.p_si
  freqM<-dataM%%diag(1/Nrecords.plots)%*%diag(SampleCoverage.Plots)
  # C_i.p_si log (C_i.p_si/(C.y_s+/y_++))
  Nspecies<-length(dataM[,1])
  Nplots<-length(dataM[1,])
  dataBeta<-
freqM*log(diag(1/Nrecords.species)%*%freqM*Nrecords/SampleCoverage)/
(1-(1-freqM)^t(matrix(Nrecords.plots,nrow=Nplots,ncol=Nspecies)))
  # force 0log0=0
  dataBeta[dataM==0]<-0
  # return the vector Hbeta_i
  apply(dataBeta, 2, sum)
}
UBEHbeta<-function(dataM)
{
  # y_+i
  Nrecords.plots<-apply(dataM, 2, sum)
  # y_++
  Nrecords<-sum(Nrecords.plots)
  # weighted sum of Hbeta_i

```

```

    UBEHbeta_i(dataM)%*%Nrecords.plots/Nrecords
  }
# Shannon's gamma diversity
Hgamma<-function(dataM)
{
  # y_s+
  Nrecords.species<-apply(dataM, 1, sum)
  # y_++
  Nrecords<-sum(Nrecords.species)
  # p_s+
  freqS<-Nrecords.species/Nrecords
  # p_si log p_si
  dataGamma<--freqS*log(freqS)
  # force 0log0=0
  dataGamma[Nrecords.species==0]<-0
  # sum
  sum(dataGamma)
}
# Unbiased Shannon's alpha diversity
UBEHgamma<-function(dataM)
{
  # y_s+
  Nrecords.species<-apply(dataM, 1, sum)
  # y_++
  Nrecords<-sum(Nrecords.species)
  # Sample Coverage
  dataM.ObservedOnce<-numeric()
  dataM.ObservedOnce<-(Nrecords.species==1)
  Singletons<-sum(dataM.ObservedOnce)
  SampleCoverage<-1-Singletons/Nrecords
  # Cp_s+
  freqS<-Nrecords.species/Nrecords*SampleCoverage
  # Cp_si log Cp_si
  dataGamma<--freqS*log(freqS)/(1-(1-freqS)^Nrecords)
  # force 0log0=0
  dataGamma[Nrecords.species==0]<-0
  # sum
  sum(dataGamma)
}

#####
##### DATA IMPORTATION #####
# Load a .csv file with species in rows and plots in columns
# The name of the first column must be 'Species'
# Check that data file has no empty cells and no supplementary rows
# and/or columns
# If so, remove them
#####
dataD<-read.csv2(file="Data.csv") # Choose the name of the file
Nspecies<-length(dataD[,1])
Nplots<-length(dataD[1,])-1
dataM<-as.matrix(dataD[,2:(Nplots+1)])
dimnames(dataM)[[1]]<-dataD$Species

```

```

#####
## Diversity Decomposition#####
#####
# ALPHA DIVERSITY
(HAlpha_i<-Halpha_i(dataM))      # Per plot, biased
exp(HAlpha_i)                    # Hill numbers
(HAlpha<-Halpha(dataM))          # Total, biased
exp(HAlpha)                      # Hill number
(UBEHAlpha_i<-UBEHalpha_i(dataM)) # Per plot, unbiased
exp(UBEHAlpha_i)                # Hill numbers
(UBEHAlpha<-UBEHalpha(dataM))    # Total, unbiased
exp(UBEHAlpha)                  # Hill number
# GAMMA DIVERSITY
(HGamma<-Hgamma(dataM))          # Biased
exp(HGamma)                     # Hill number
(UBEHGamma<-UBEHgamma(dataM))    # Unbiased
exp(UBEHGamma)                  # Hill number
#####BETA DIVERSITY#####
(HBeta_i<-Hbeta_i(dataM))        # Per plot, biased
exp(HBeta_i)                    # Hill numbers
(HBeta<-Hbeta(dataM))           # Total, biased
exp(HBeta)                      # Hill number
(UBEHBeta_i<-UBEHbeta_i(dataM)) # Per plot, unbiased
exp(UBEHBeta_i)                # Hill numbers
(UBEHBeta<-UBEHbeta(dataM))     # Total, unbiased
exp(UBEHBeta)                  # Hill number
#####
## Test of Significance #####
## BETA DIVERSITY #####
#####
Nrecords.plots<-apply(dataM, 2, sum)
Nrecords.species<-apply(dataM, 1, sum)
Nrecords<-sum(Nrecords.plots)
PlotSize<-t(matrix(Nrecords.plots,nrow=Nplots,ncol=Nspecies))
GlobalSpeciesFrequency<-
matrix(Nrecords.species/Nrecords,nrow=Nspecies,ncol=Nplots)
LocalSpeciesFrequency<-dataM%*%diag(1/Nrecords.plots)
SIM<-1:10000                      # Choose the number of simulations
RiskLevel<-0.05                   # Choose the risk level

HBetaSimulation<-function(i)
{
  dataSIM<-
matrix(rbinom(Nplots*Nspecies,PlotSize,LocalSpeciesFrequency),nrow=N
species, ncol=Nplots)
  UBEHbeta(dataSIM)
}

RawHBetaDistribution<-sapply(SIM, HBetaSimulation)
# Recenter
HBetaDistribution<-RawHBetaDistribution+UBEHBeta-
mean(RawHBetaDistribution)

# Graphical parameters
oldpar <- par(no.readonly=TRUE)
par(mfrow=c(1,2))

```

```
# Plot HBetaDistribution
plot(density(HBetaDistribution), col="red", lwd=3, main="Red:
Expected ; Green: 95% CI ; Blue: Observed", xlab="Confidence
interval of H_beta" )
(HbetaCI<-quantile(HBetaDistribution, c(RiskLevel/2, 1-
RiskLevel/2)))
abline(v=HbetaCI, col="green", lwd=3)
abline(v=UBEHBeta, col="blue", lwd=3)
# Plot Hill numbers
BetaDivDistribution<-exp(HBetaDistribution)
plot(density(BetaDivDistribution), col="red", lwd=3, main="Red:
Expected ; Green: 95% CI ; Blue: Observed", xlab="Confidence
interval of Shannon diversity" )
(BetaDivCI<-quantile(BetaDivDistribution, c(RiskLevel/2, 1-
RiskLevel/2)))
abline(v=BetaDivCI, col="green", lwd=3)
abline(v=exp(UBEHBeta), col="blue", lwd=3)

# Restore graphical parameters
par(oldpar)
```