Moth

Murilo Miranda

11 Sep 2014

Load Data and Codes

- Read dataset with id, site, period, species, abund

```
data <- read.csv("data.csv", h = T)
head(data)
```

```
##   ID Site Period                   species Nr
## 1  1   42      1            Noctua pronuba  8
## 2  2   42      1      Atlantarctia tigrina  1
## 3  3   42      1           Anarta myrtilli  1
## 4  4   42      1      Lycophotia molothina  7
## 5  5   42      1        Harpyia milhauseri  1
## 6  6   42      1 Pseudoterpna coronillaria  1
```

```
library(xtable)
library(boot)
source('chao.R')
source('distanc.R')
```

Distances

The distances between pairs were calculated using QGIS for IPCC and IGeoE:

- Hayford Gauss IPCC:

```
dist <- read.csv("dist_mat_ipcc.csv", h = T)
dist <- dist[-1]
DIST1 <- distanc(dist)
head(DIST1)
```

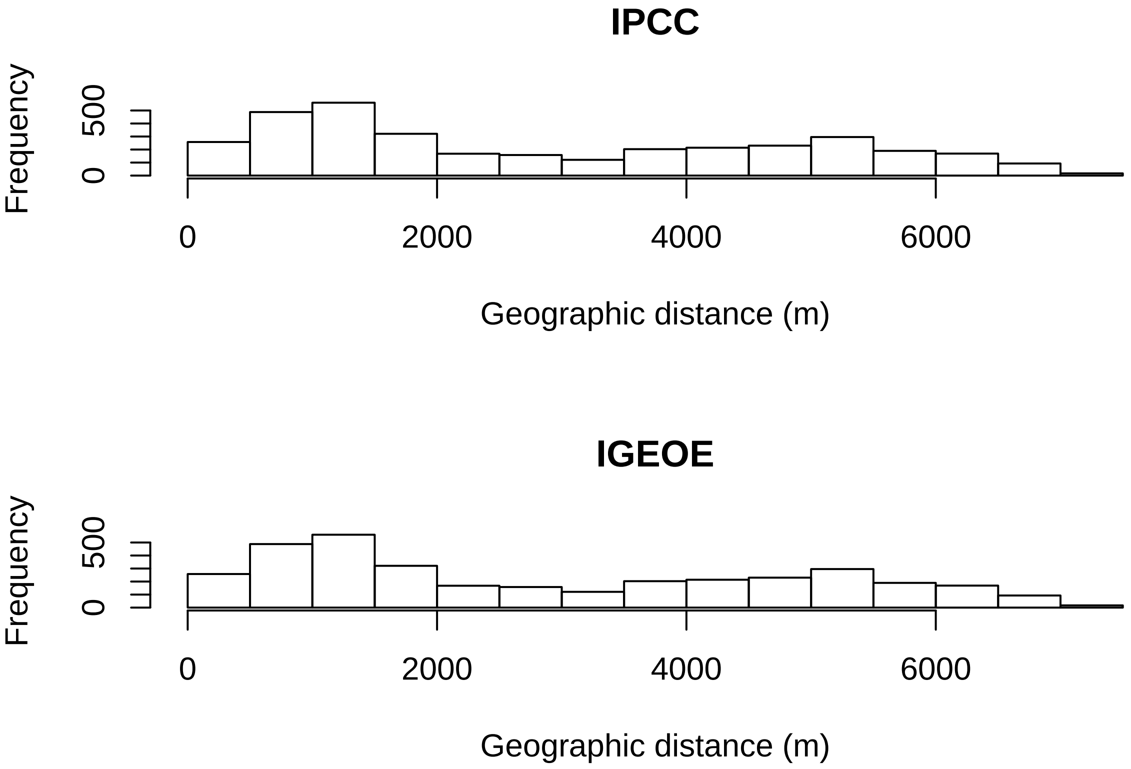
```
##      [,1] [,2]   [,3]
## [1,]    1    2  60.00
## [2,]    1    3  60.00
## [3,]    1    4  84.85
## [4,]    1    5 247.39
## [5,]    1    6 247.40
## [6,]    1    7 339.41
```

- Hayford Gauss IGeoE

```
dist <- read.csv("dist_mat_igeoe.csv", h = T)
dist <- dist[-1]
DIST2 <- distanc(dist)
head(DIST2)
```

```
##      [,1] [,2]   [,3]
## [1,]    1    2  60.00
## [2,]    1    3  60.00
## [3,]    1    4  84.85
## [4,]    1    5 247.39
## [5,]    1    6 247.40
## [6,]    1    7 339.41
```

```
par(mfrow=c(2, 1))
hist(DIST1[, 3], main = "IPCC", xlab = "Geographic distance (m)", ylab = "Frequency")
hist(DIST2[, 3], main = "IGEOE", xlab = "Geographic distance (m)", ylab = "Frequency")
```



The histograms have the same behavior, then I chose the first vector of distances (IPCC) for analysing the data.

Data

Sorensen and Jaccard Estimators were estimated using unseen species (Chao et al 2005). The data-set was subseted by sample. Variables created from the chao-function have the following columns:

```
sample1 <- chao(1, 1, "sample1")
sample2 <- chao(2, 2, "sample2")
sample3 <- chao(3, 3, "sample3")
sample4 <- chao(4, 4, "sample4")
sample5 <- chao(5, 5, "sample5")
sample6 <- chao(6, 6, "sample6")
```

- p: sampling period.

- s1: the code of sample 1.
- s2: the code of sample 2.
- n: the number of observed species in the sample 1.
- nsp1: the number of species in the sample 1.
- m: the number of observed species in the sample 2.
- nsp2: the number of species in the sample 2.
- nsh: the number of observed shared species in the two samples.
- fm1: the observed number of shared species that occur once in sample 2.
- fm2: the observed number of shared species that occur twice in sample 2.
- f1m: the observed number of shared species that occur once in sample 1.
- f2m: the observed number of shared species that occur twice in sample 1.
- u_pt1: abundance sum of shared species divided by species number for the sample 1.
- v_pt1: abundance sum of shared species divided by species number for the sample 2.
- Junad: Jaccard unadjusted estimator.
- Lunad: Sorensen unadjusted estimator.
- U: u_pt1 plus unseen species for the sample 2.
- V: v_pt1 plus unseen species for the sample 1.
- Jabd: Jaccard adjusted estimator.
- Labd: Sorensen adjusted estimator.

```r
head(sample1)
```

```
##      p s1 s2  n nsp1   m nsp2 nsh fm1 fm2 f1m f2m   u_pt1  v_pt1  Junad
## [1,] 1  1  2 44   17  72   26  13   4   2   5   4 0.88636 0.6667 0.6142
## [2,] 1  1  3 44   17  73   33  13   5   1   5   5 0.86364 0.6301 0.5731
## [3,] 1  1  4 44   17   4    4   1   1   1   0   1 0.04545 0.2500 0.0400
## [4,] 1  1  5 44   17 114   33  12   2   1   4   4 0.86364 0.6754 0.6103
## [5,] 1  1  6 44   17  67   25  10   4   2   4   2 0.77273 0.4328 0.3840
## [6,] 1  1  7 44   17  97   29  12   3   1   5   4 0.81818 0.7216 0.6219
##        Lunad      U      V    Jabd   Labd
## [1,] 0.76098 1.0000 0.7854 0.78543 0.8798
## [2,] 0.72864 1.0000 0.7172 0.71715 0.8353
## [3,] 0.07692 0.0625 0.2500 0.05263 0.1000
## [4,] 0.75803 0.9312 0.7440 0.70526 0.8272
## [5,] 0.55487 1.0000 0.5495 0.54953 0.7093
## [6,] 0.76689 1.0000 0.9231 0.92315 0.9600
```

Exploratory Analysis

These functions were used to synthesize the samples for each year. Moth samplings were conducted for two years (2011 and 2012), each year had three sampling period.

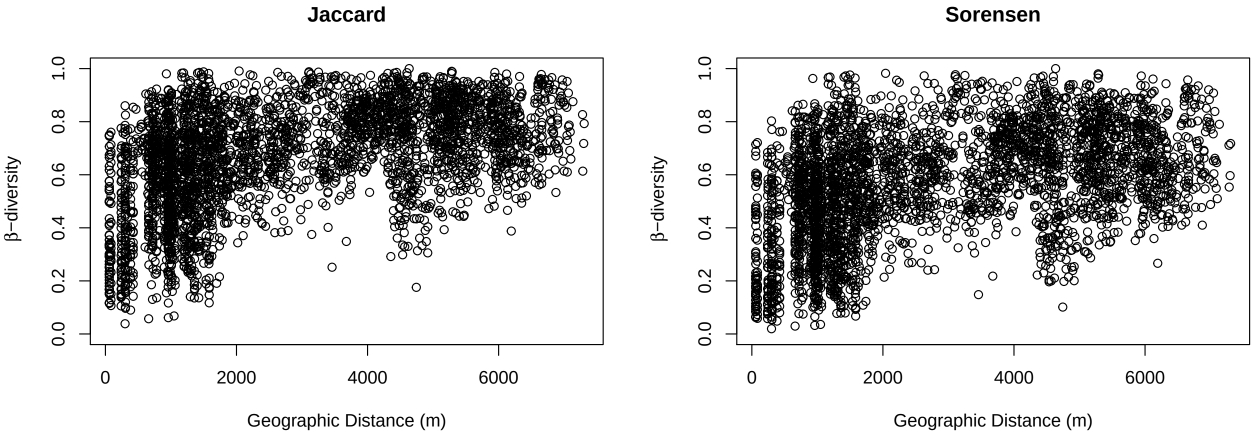
```r
trisamples <- function(base1, base2, base3, vr){
  base <- matrix(0, nrow = dim(base1)[1])
  for(i in 1:dim(base1)[1]){
    base[i] <- (base1[i, vr] + base2[i, vr] + base3[i, vr])/3
  }
  base
}

bisamples <- function(base1, base2){
  base <- matrix(0, nrow = length(base1))
  for(i in 1:length(base1)){
    base[i] <- (base1[i] + base2[i])/2
  }
  base
}
```

- Calculated the dissimilarity average per site for first year

```
Jperiod1 <- trisamples(sample1, sample2, sample3, 19)
Lperiod1 <- trisamples(sample1, sample2, sample3, 20)
```

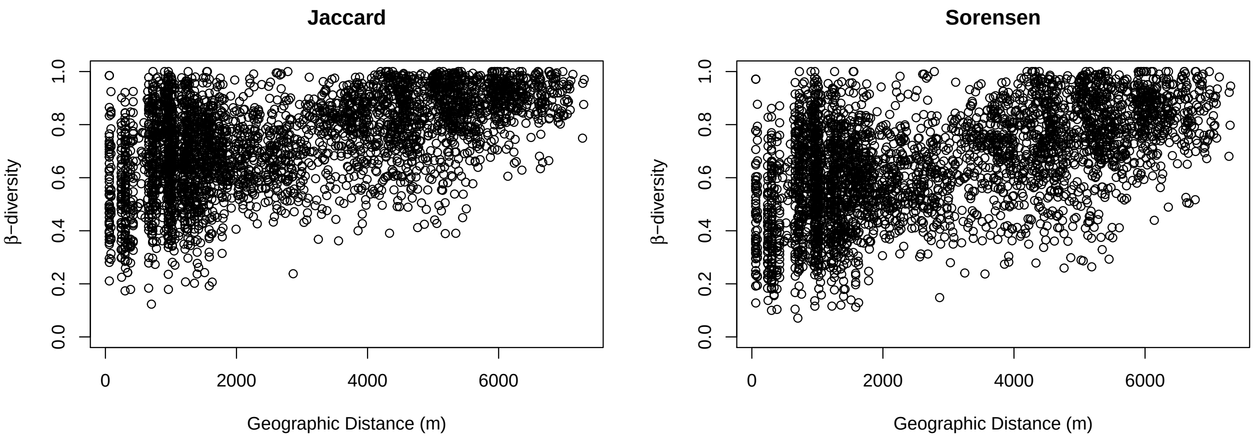
```
par(mfrow=c(1, 2))
plot(DIST1[, 3], (1 - Jperiod1), ylim = c(0, 1), xlab = "Geographic Distance (m)", ylab =
       expression(~beta*"-diversity"), main = "Jaccard")
plot(DIST1[, 3], (1 - Lperiod1), ylim = c(0, 1), xlab = "Geographic Distance (m)", ylab =
       expression(~beta*"-diversity"), main = "Sorensen")
```



- Calculated the dissimilarity average per site for second year

```
Jperiod2 <- trisamples(sample4, sample5, sample6, 19)
Lperiod2 <- trisamples(sample4, sample5, sample6, 20)
```

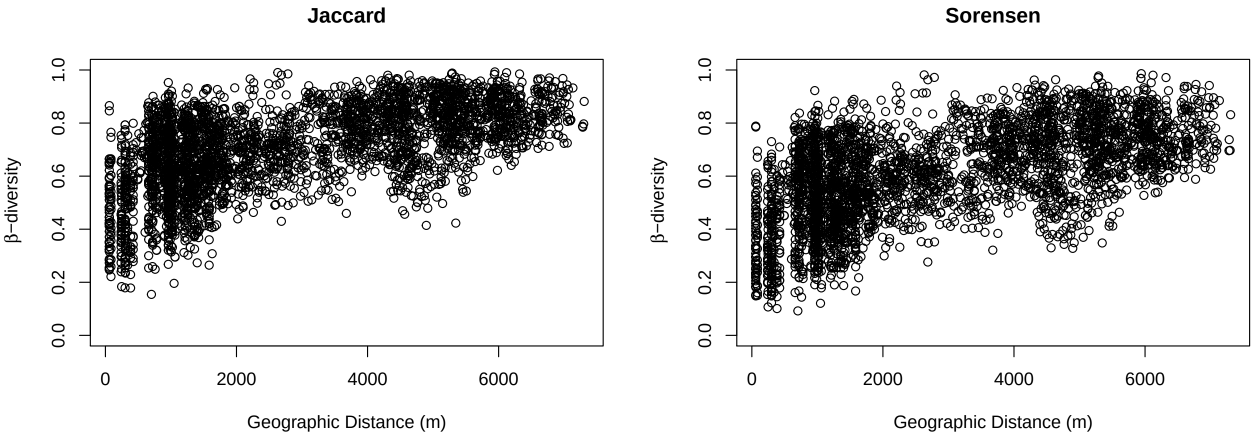
```
par(mfrow=c(1, 2))
plot(DIST1[, 3], (1 - Jperiod2), ylim = c(0, 1), xlab = "Geographic Distance (m)", ylab =
       expression(~beta*"-diversity"), main = "Jaccard")
plot(DIST1[, 3], (1 - Lperiod2), ylim = c(0, 1), xlab = "Geographic Distance (m)", ylab =
       expression(~beta*"-diversity"), main = "Sorensen")
```



- Calculated the dissimilarity average per site for both years

```
Jperiod <- bisamples(Jperiod1, Jperiod2)
Lperiod <- bisamples(Lperiod1, Lperiod2)
```

```
par(mfrow=c(1, 2))
plot(DIST1[, 3], (1 - Jperiod), ylim = c(0, 1), xlab = "Geographic Distance (m)", ylab =
       expression(~beta*"-diversity"), main = "Jaccard")
plot(DIST1[, 3], (1 - Lperiod), ylim = c(0, 1), xlab = "Geographic Distance (m)", ylab =
       expression(~beta*"-diversity"), main = "Sorensen")
```



DATASET: I created a new database with six columns:

- p: no functionality, I created just to standardize with other functions previously created.
- s1: site 1
- s2: site 2
- Jabd: Jaccard adjusted estimator
- Labd: Sorensen adjusted estimator
- ipcc: distance between site 1 and site 2 (Hayford Gauss IPCC)

```
b.div.mean <- cbind(sample1[, 1], sample1[, 2], sample1[, 3], Jperiod, Lperiod, DIST1[, 3])
colnames(b.div.mean) <- c("p", "s1", "s2", "Jabd", "Labd", "ipcc")

head(b.div.mean)
```

```
##      p s1 s2   Jabd   Labd   ipcc
## [1,] 1  1  2 0.6520 0.7724  60.00
## [2,] 1  1  3 0.7169 0.8153  60.00
## [3,] 1  1  4 0.2347 0.3054  84.85
## [4,] 1  1  5 0.5494 0.6839 247.39
## [5,] 1  1  6 0.6330 0.7575 247.40
## [6,] 1  1  7 0.6073 0.7244 339.41
```

Samples were taken in three landscapes that represent the farmland abandonment gradient: meadow-dominated, shrub-dominated and forest-dominated. In each landscape had 28 fixed sampling sites divided into four biotopes (meadow, short shrub, tall shrub and woodland), totaling 84 sampling sites.

- forest-dominated:

```
mf <- c(56) #m - meadow
sf <- c(36, 39, 42, 45, 46) #s - short shrub
tf <- c(34, 49) #t - tall shrub
wf <- c(29, 30, 31, 32, 33, 35, 37, 38, 40, 41, 43, 44, 47, 48, 50, 51, 52, 53, 54, 55)
#w - woodland
```

- shrub-dominated (m - mixed)

```
mm <- c(68, 75, 77, 84)
sm <- c(60, 64, 65, 66, 67, 70, 78, 79, 82)
tm <- c(57, 58, 59, 61, 62, 63, 69, 71, 72, 74, 76)
wm <- c(73, 80, 81, 83)
```

- meadow-dominated (a - agriculture)

```
ma <- c(1, 9, 14, 15, 16, 17, 18, 19, 21, 23, 24, 25, 27)
sa <- c(2, 8, 10, 11, 12, 13, 20)
ta <- c(3, 4, 5, 6, 7, 28)
wa <- c(22, 26)
```

FIRST APROARCH

Within landscape

```
landsc_within <- function(base, s1, s2){
  result <- NULL
  for(i in s1){
    for(j in s2){
      if(i != j){
        result <- rbind(result, base[base[, 2] == i & base[, 3] == j, ])
      }
    }
  }
  result
}
```

```
idmea <- c(ma, sa, ta, wa)
idmix <- c(mm, sm, tm, wm)
idfor <- c(mf, sf, tf, wf)
```

```
m.mea <- landsc_within(b.div.mean, idmea, idmea)
m.mix <- landsc_within(b.div.mean, idmix, idmix)
m.for <- landsc_within(b.div.mean, idfor, idfor)
```

This function creates a database for each landscape, but with all pairs within landscape.

Regression models

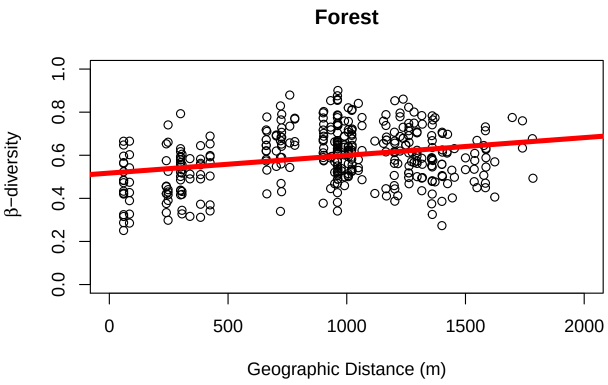
- Jaccard Estimator

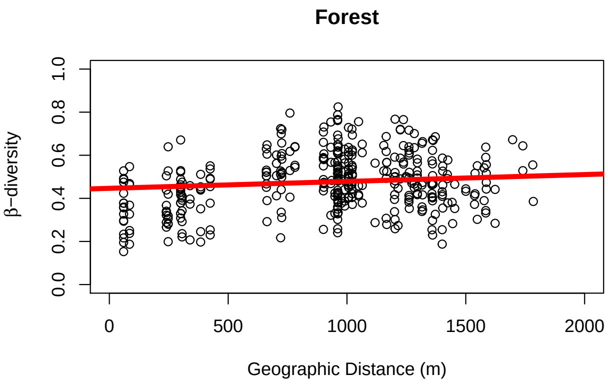
```
jm.modelfor <- lm((1 - m.for[, 4]) ~ m.for[, 6])
jm.modelmix <- lm((1 - m.mix[, 4]) ~ m.mix[, 6])
jm.modelmea <- lm((1 - m.mea[, 4]) ~ m.mea[, 6])
```

- Sorensen Estimator

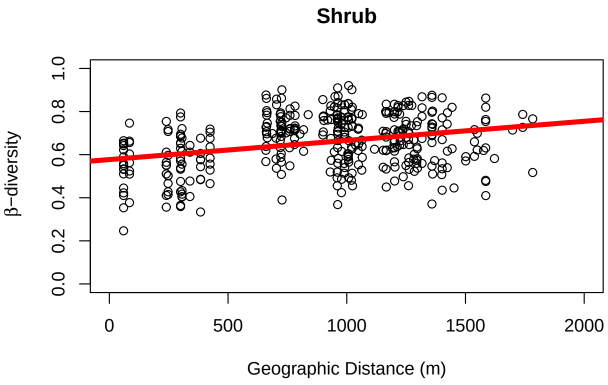
```
sm.modelfor <- lm((1 - m.for[, 5]) ~ m.fordata[, 6])
sm.modelmix <- lm((1 - m.mix[, 5]) ~ m.mixdata[, 6])
sm.modelmea <- lm((1 - m.mea[, 5]) ~ m.meadata[, 6])
```

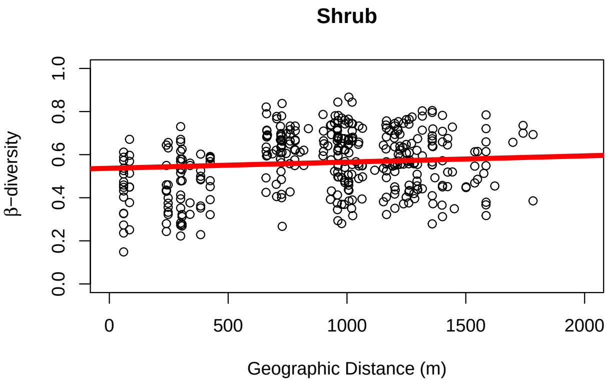
1. Forest



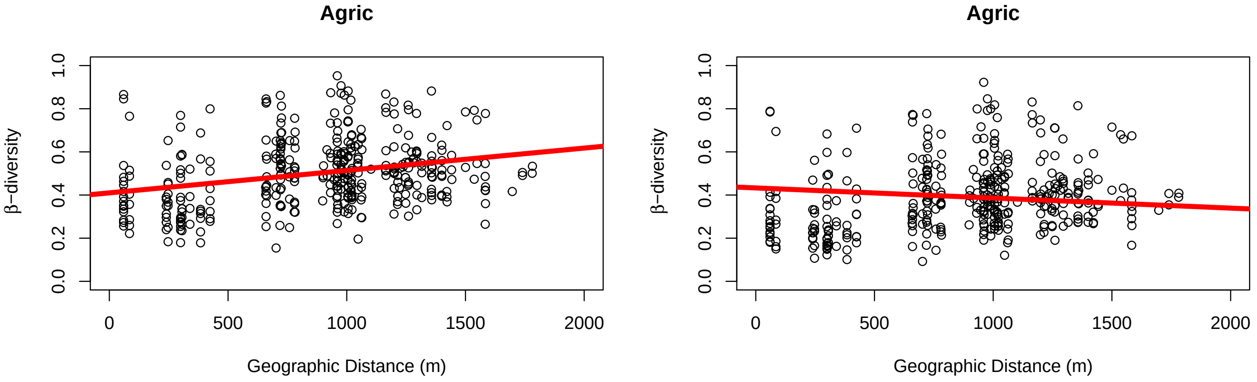


2. Shrub





3. Meadow

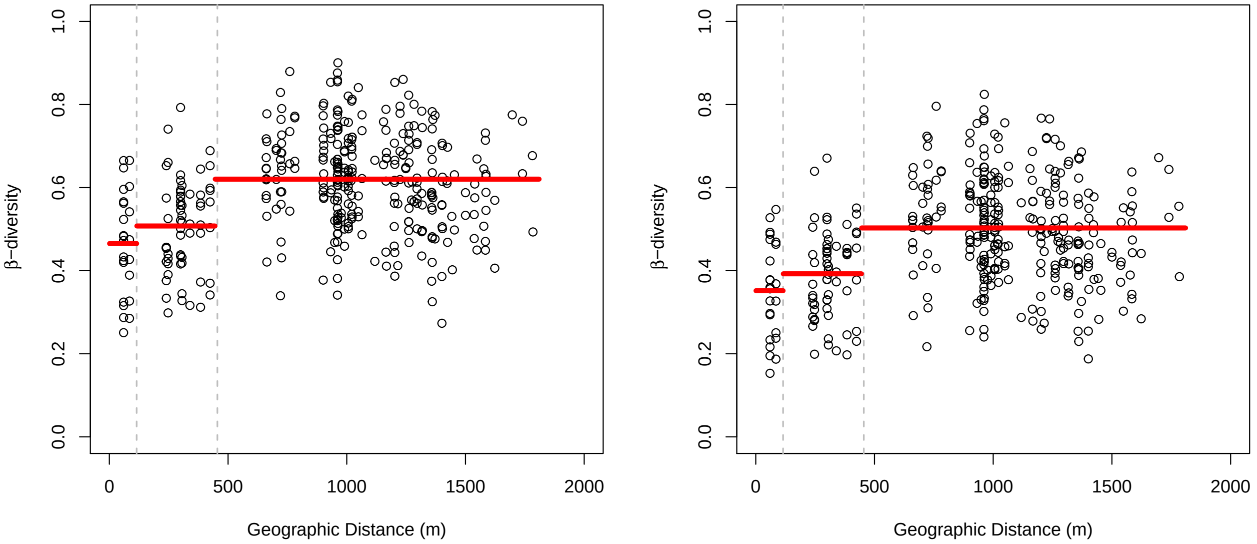


- Mean Dissimilarity for each scale

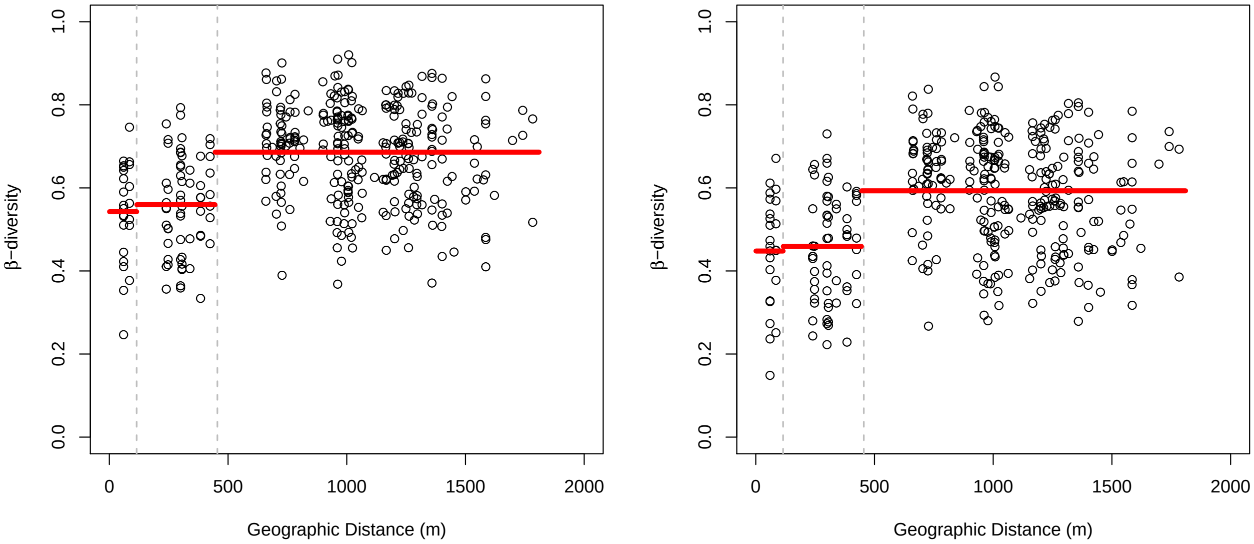
```
plot.scale2 <- function(base, estimator = 0){
  #if estimator == 0: Jaccard, otherwise: Sorensen (default is Jaccard)
  area <- base[, 6]
  if(estimator) div <- base[, 5]
  else div <- base[, 4]
  plot(area, 1 - div, ylim = c(0, 1), xlim = c(0, 2000), xlab = "Geographic Distance (m)",
       ylab = expression(~beta*"-diversity"))
  abline(v = c(115, 455), col = "gray", lwd = 1.5, lty = 2)
  lines(1:115, rep(1 - mean(div[area <= 115]), 115), col = 2, lwd = 4.5)
  lines(116:445, rep(1 - mean(div[(area <= 445) & (area > 115)]), 330), col = 2, lwd = 4.5)
  lines(446:1810, rep(1 - mean(div[area > 445]), 1365), col = 2, lwd = 4.5)
}
```

1. Forest

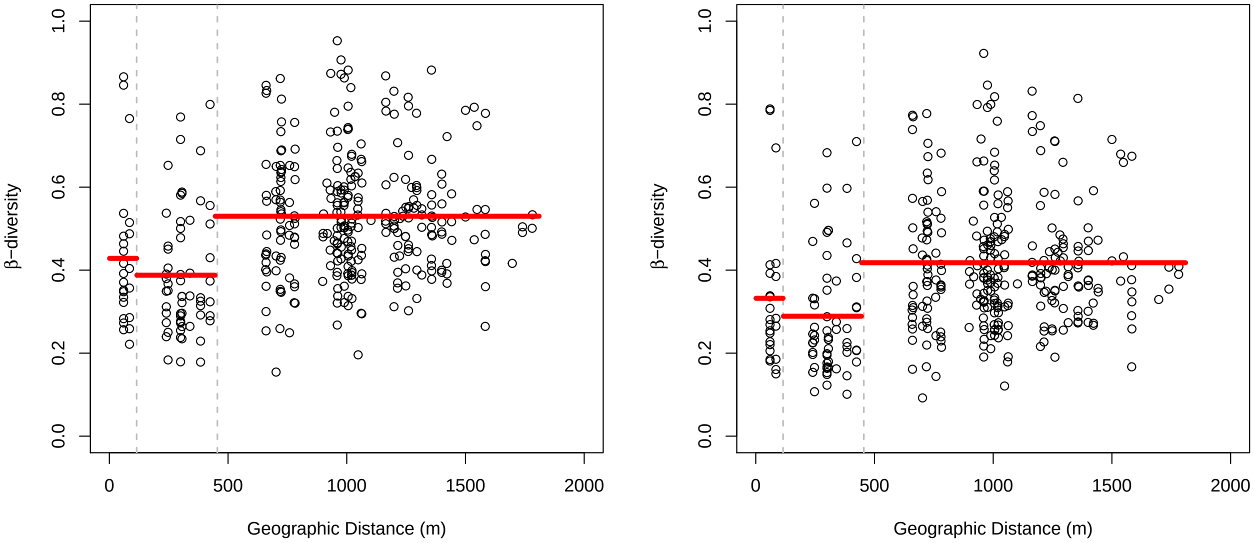
```
par(mfrow = c(1, 2))
plot.scale2(m.for)
plot.scale2(m.for, 1)
```



2. Shrub



3. Meadow



Bootstrapping

```r
# function to obtain regression weights
bs.jac <- function(formula, data, ind){
  d <- data[ind,] # allows boot to select sample
  fit <- lm(formula, data = d)
  return(c(coef(fit), summary(fit)$r.square, mean(d[, 4])))
}

bs.sor <- function(formula, data, ind){
  d <- data[ind,] # allows boot to select sample
  fit <- lm(formula, data = d)
  return(c(coef(fit), summary(fit)$r.square, mean(d[, 5])))
}
```

Within landscape

```
# bootstrapping with 2000 replications
jforres <- boot(data = as.data.frame(m.fordata), statistic = bs.jac, R = 2000,
                formula = (1 - Jabd) ~ ipcc)
sforres <- boot(data = as.data.frame(m.fordata), statistic = bs.sor, R = 2000,
                formula = (1 - Labd) ~ ipcc)
jmixres <- boot(data = as.data.frame(m.mixdata), statistic = bs.jac, R = 2000,
                formula = (1 - Jabd) ~ ipcc)
smixres <- boot(data = as.data.frame(m.mixdata), statistic = bs.sor, R = 2000,
                formula = (1 - Labd) ~ ipcc)

jmeares <- boot(data = as.data.frame(m.meadata), statistic = bs.jac, R = 2000,
                formula = (1 - Jabd) ~ ipcc)
smeares <- boot(data = as.data.frame(m.meadata), statistic = bs.sor, R = 2000,
                formula = (1 - Labd) ~ ipcc)
```

Confidence Interval

- Bootstrap 95% CI for regression coefficients

Jaccard

```
jforres.ci <- ci.boot(jforres)
jmixres.ci <- ci.boot(jmixres)
jmeares.ci <- ci.boot(jmeares)
```

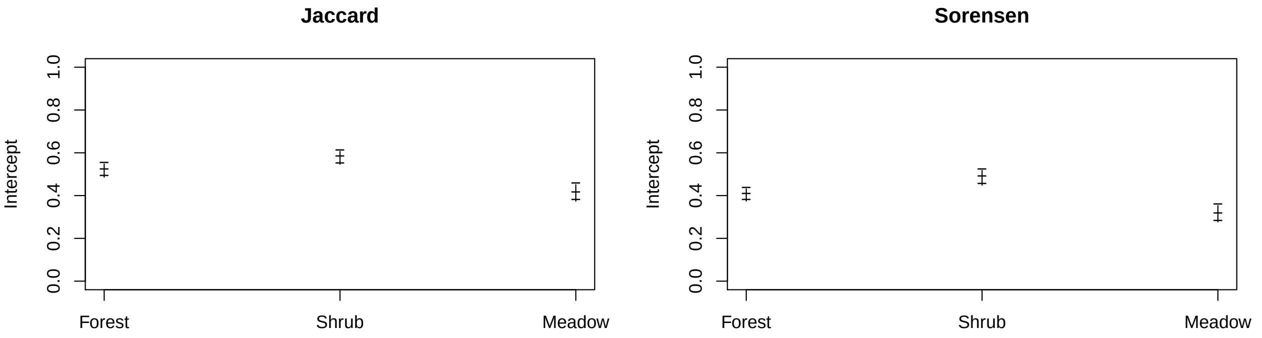
Sorensen

```
jforres.ci <- ci.boot(jforres)
jmixres.ci <- ci.boot(jmixres)
jmeares.ci <- ci.boot(jmeares)
```

Results

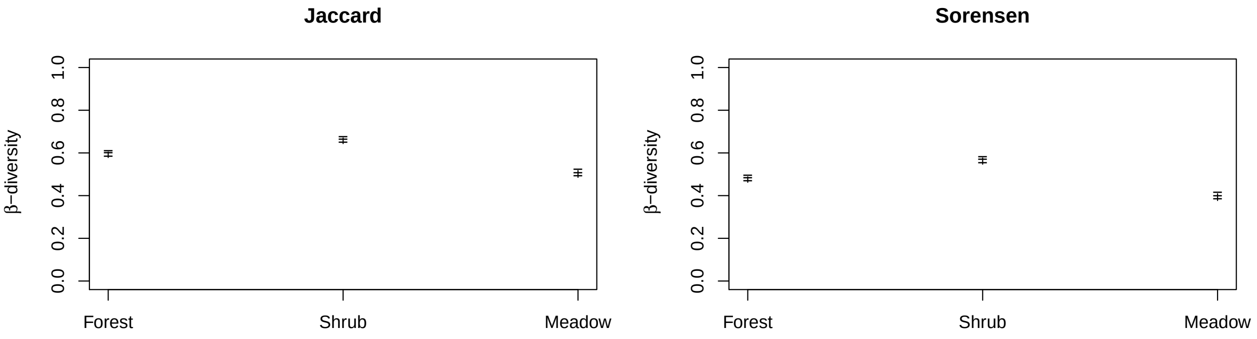
- Intercept

```
par(mfrow = c(1, 2))
interc.plot3(jforres.ci, jmixres.ci, jmeares.ci)
title("Jaccard", cex = 1.5)
interc.plot3(sforres.ci, smixres.ci, smeares.ci)
title("Sorensen", cex = 1.5)
```



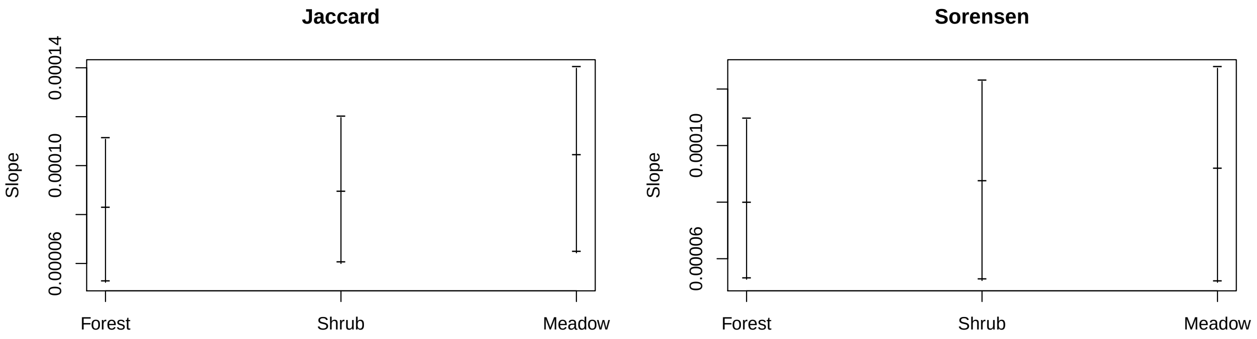
- Dissimilarity

```r
par(mfrow = c(1, 2))
dissim.plot3(jforres.ci, jmixres.ci, jmeares.ci)
title("Jaccard", cex = 1.5)
dissim.plot3(sforres.ci, smixres.ci, smeares.ci)
title("Sorensen", cex = 1.5)
```



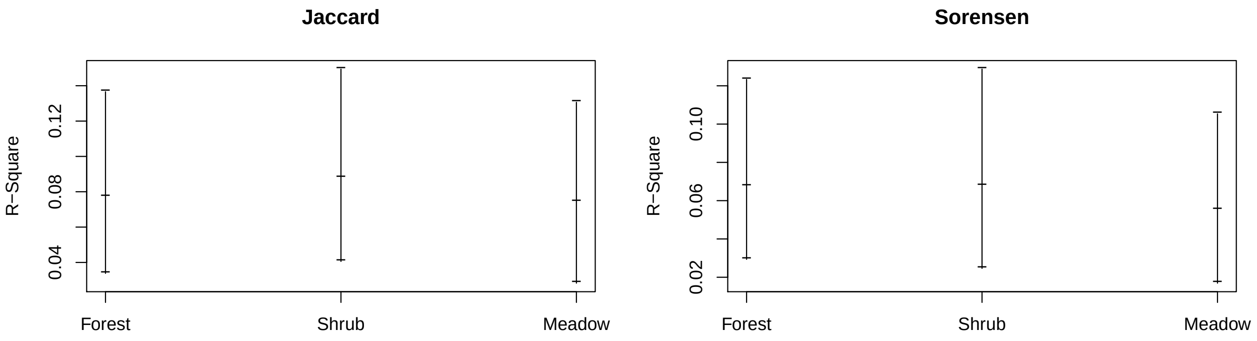
- Slope

```r
par(mfrow = c(1, 2))
slope.plot3(jforres.ci, jmixres.ci, jmeares.ci)
title("Jaccard", cex = 1.5)
slope.plot3(sforres.ci, smixres.ci, smeares.ci)
title("Sorensen", cex = 1.5)
```



- R-Square

```r
par(mfrow = c(1, 2))
rsquar.plot3(jforres.ci, jmixres.ci, jmeares.ci)
title("Jaccard", cex = 1.5)
rsquar.plot3(sforres.ci, smixres.ci, smeares.ci)
title("Sorensen", cex = 1.5)
```



Tables

Jaccard

```r
print(xtable(boot.table2(jforres.ci, jmixres.ci, jmeares.ci),
             digits = 4, caption = "Intercept"), type = "latex", comment = FALSE)
```

	mean.ci	lower.ci	upper.ci
Forest	0.5172	0.4862	0.5470
Shrub	0.5771	0.5464	0.6065
Meadow	0.4101	0.3736	0.4508

Table 1: Intercept

	mean.ci	lower.ci	upper.ci
Forest	0.5927	0.6040	0.5785
Shrub	0.6569	0.6699	0.6437
Meadow	0.5009	0.5176	0.4855

Table 2: Dissimilarity

	mean.ci	lower.ci	upper.ci
Forest	0.000082	0.000052	0.000111
Shrub	0.000089	0.000060	0.000120
Meadow	0.000104	0.000064	0.000140

Table 3: Slope

Sorensen

```r
print(xtable(boot.table2(sforres.ci, smixres.ci, smeares.ci),
             digits = 4, caption = "Intercept"), type = "latex", comment = FALSE)
```

	mean.ci	lower.ci	upper.ci
Forest	0.0771	0.0338	0.1366
Shrub	0.0880	0.0406	0.1494
Meadow	0.0744	0.0283	0.1306

Table 4: R-square

	mean.ci	lower.ci	upper.ci
Forest	0.4029	0.3741	0.4311
Shrub	0.4844	0.4485	0.5180
Meadow	0.3119	0.2768	0.3534

Table 5: Intercept

	mean.ci	lower.ci	upper.ci
Forest	0.4758	0.4884	0.4625
Shrub	0.5626	0.5763	0.5473
Meadow	0.3920	0.4089	0.3771

Table 6: Dissimilarity

	mean.ci	lower.ci	upper.ci
Forest	0.000079	0.000053	0.000109
Shrub	0.000087	0.000052	0.000123
Meadow	0.000091	0.000052	0.000127

Table 7: Slope

	mean.ci	lower.ci	upper.ci
Forest	0.0676	0.0293	0.1233
Shrub	0.0678	0.0246	0.1287
Meadow	0.0552	0.0169	0.1054

Table 8: R-square