

# Introduction

*Murilo Miranda*

*12 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:

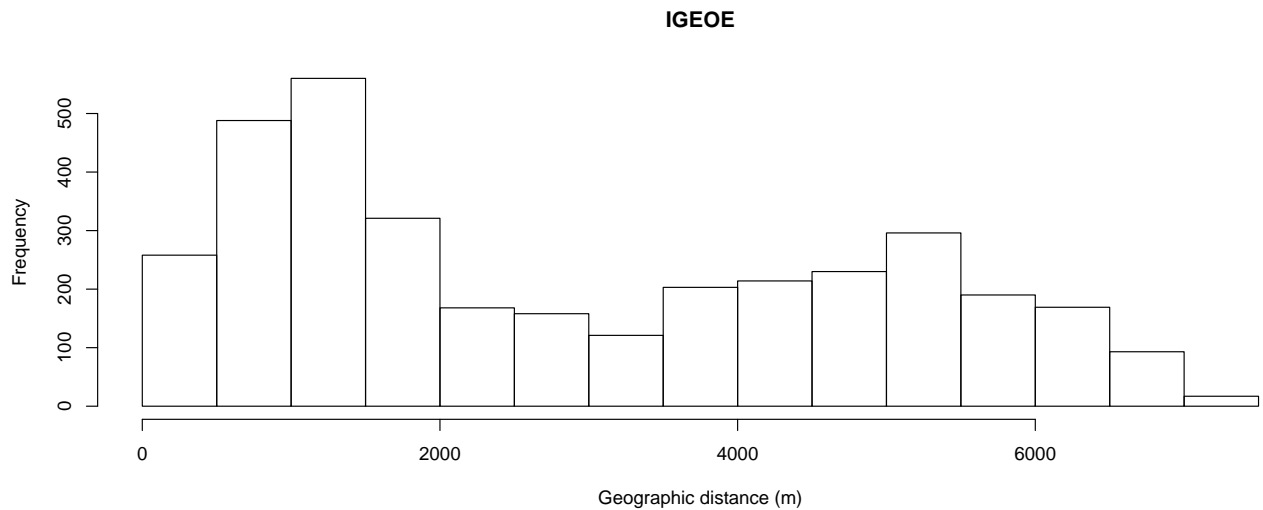
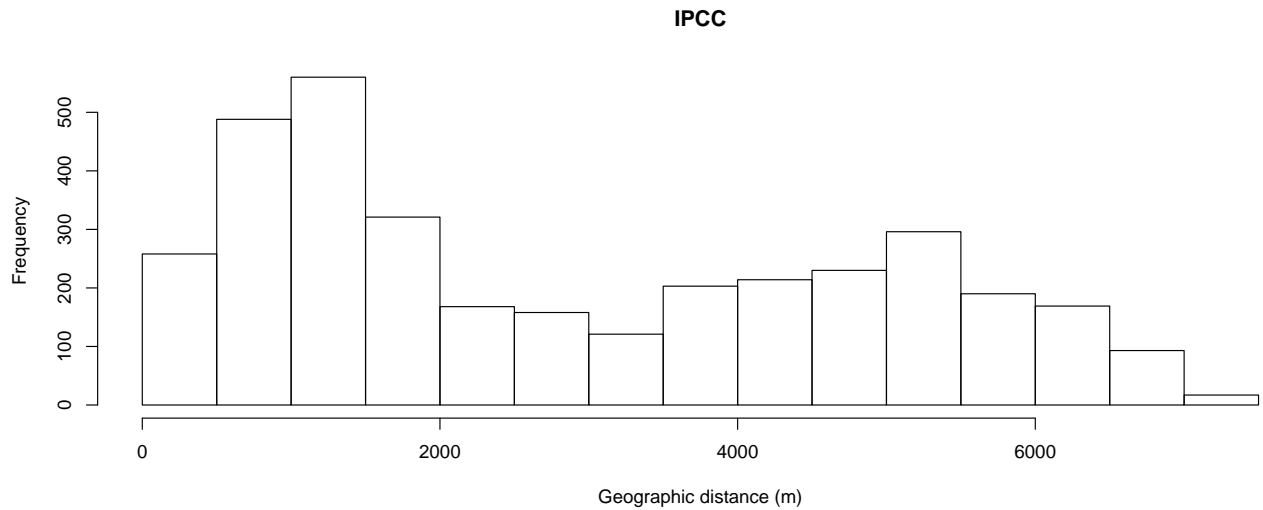
```
dist <- read.csv("dist_mat_ipcc.csv", h = T)
dist <- dist[-1]
DIST1 <- distanc(dist)
head(DIST1)
```

```
##      [,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
```



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.
- flm: 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.

```
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.

```
trisesamples <- 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 average dissimilarity per site for first year

```

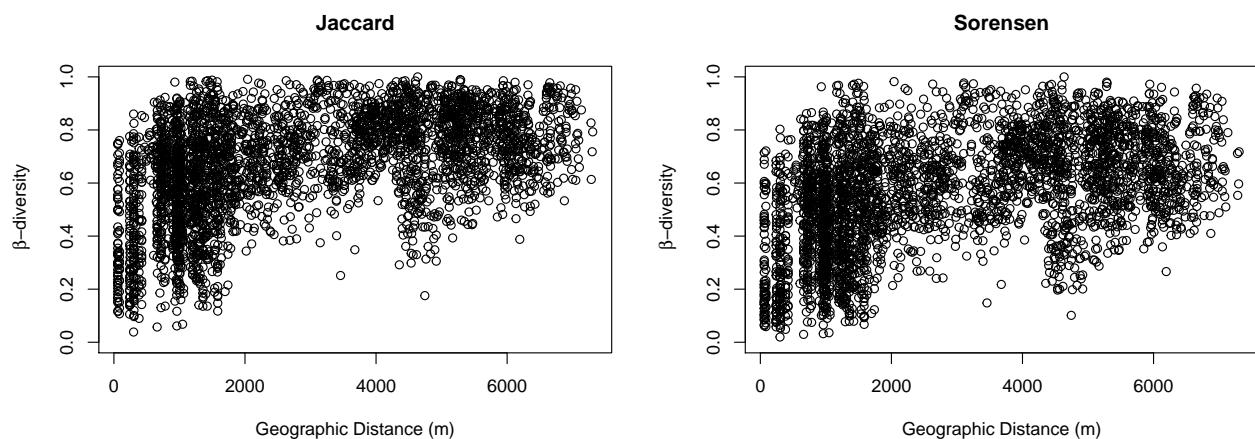
Jperiod1 <- trisamples(sample1, sample2, sample3, 19)
Lperiod1 <- trisamples(sample1, sample2, sample3, 20)

```

```

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 average dissimilarity per site for second year

```

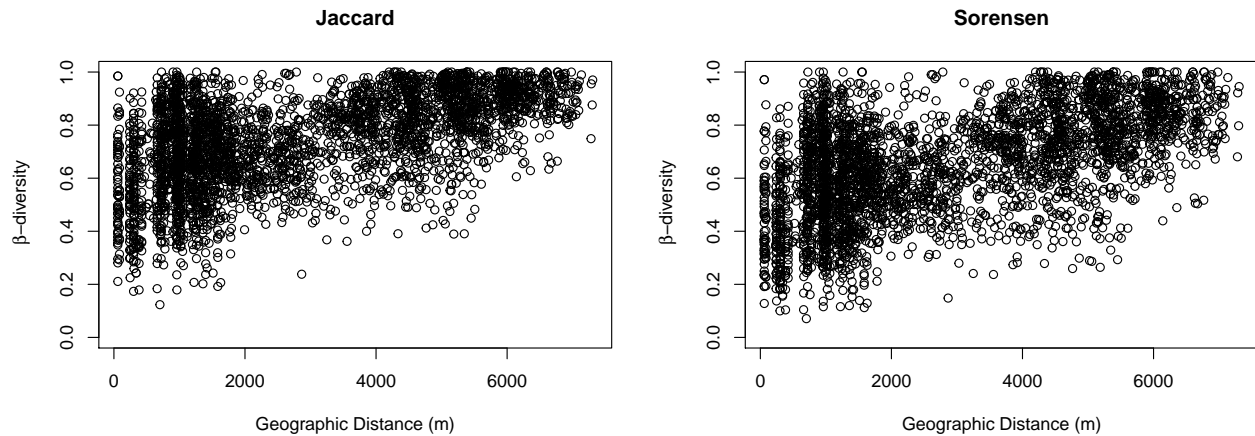
Jperiod2 <- trisamples(sample4, sample5, sample6, 19)
Lperiod2 <- trisamples(sample4, sample5, sample6, 20)

```

```

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")

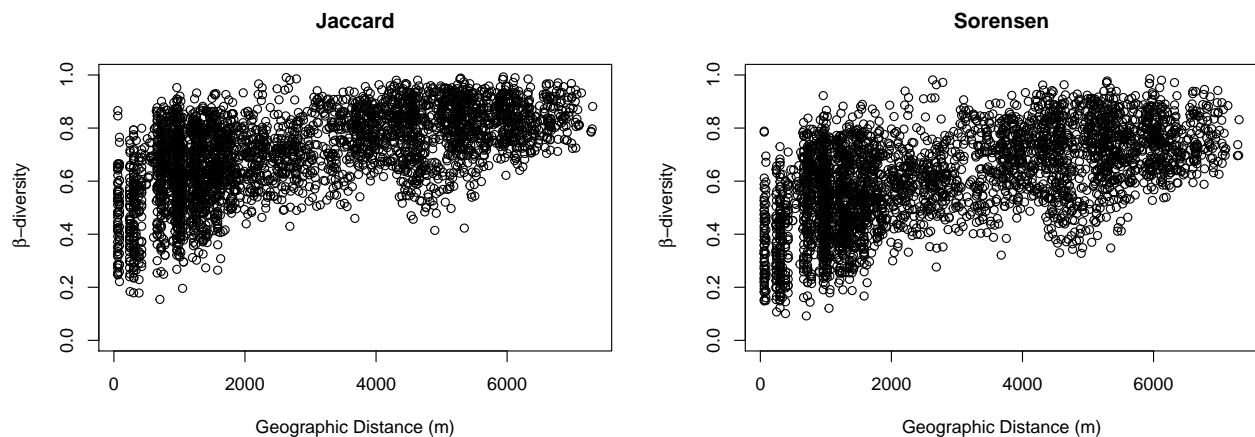
```



- Calculated the average dissimilarity 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)
```

## LANDSCAPE APPROARCH

### 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

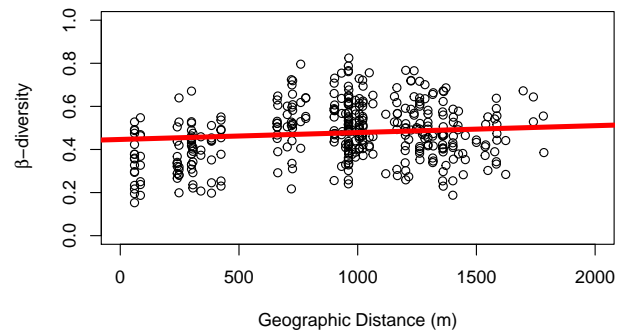
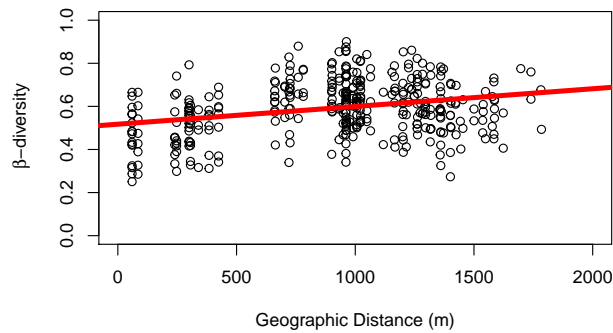
- Jaccard Estimator

```
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])
```

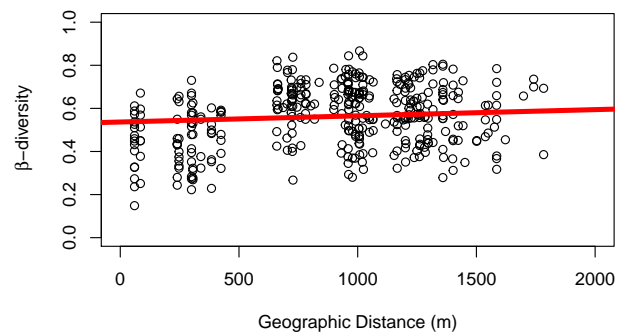
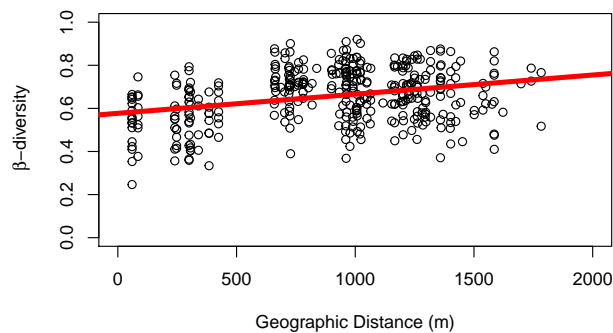
- Sorensen Estimator

```
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])
```

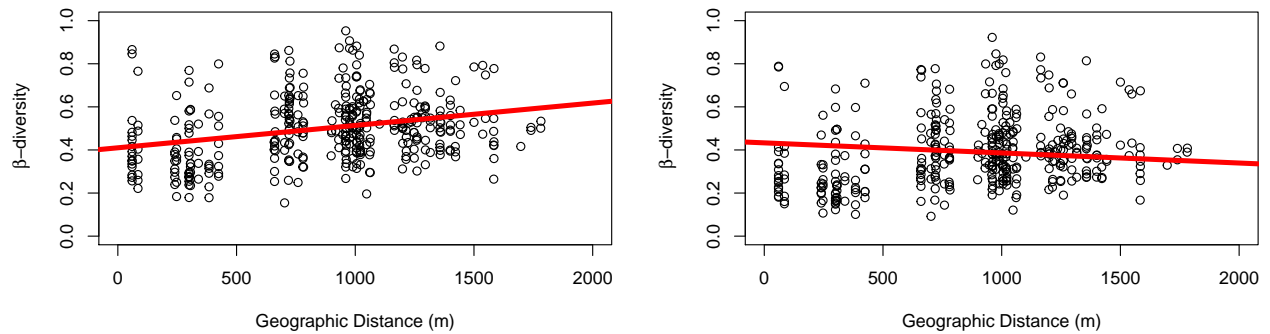
### 1. Forest



### 2. Shrub



### 3. Meadow

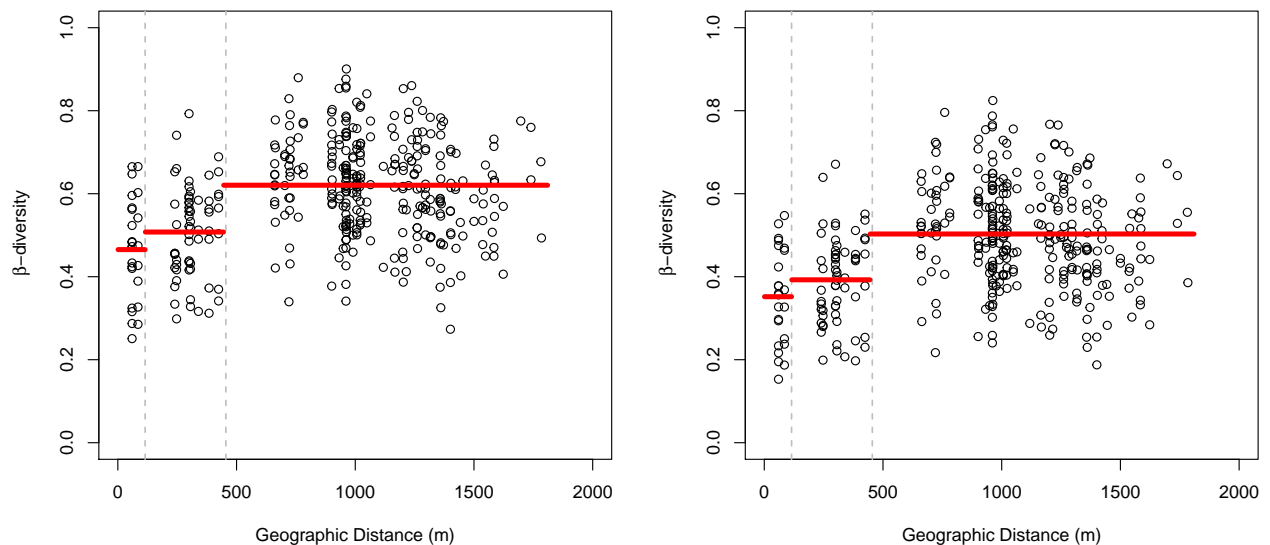


- Mean Dissimilarity for each scale (left side: Sorensen and right side: Jaccard)

```
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)
}
```

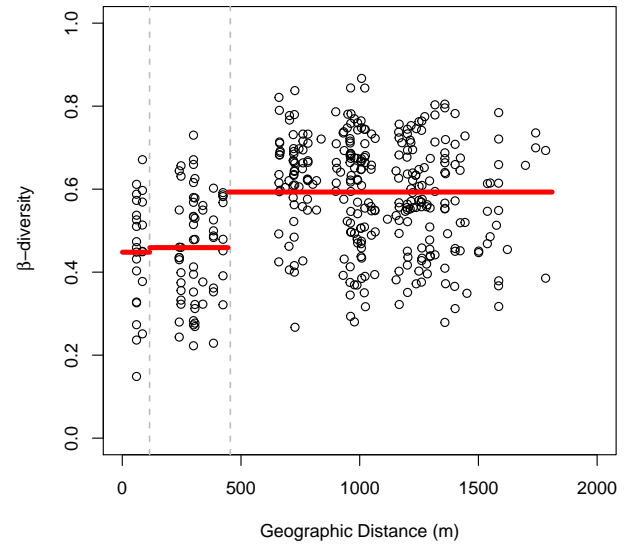
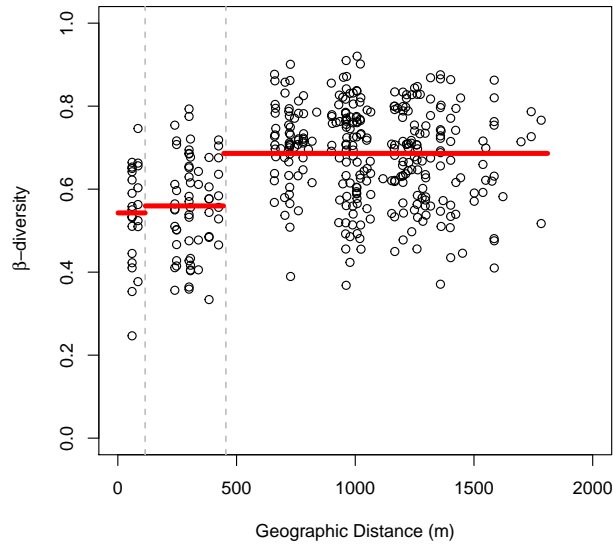
## 1. Forest

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

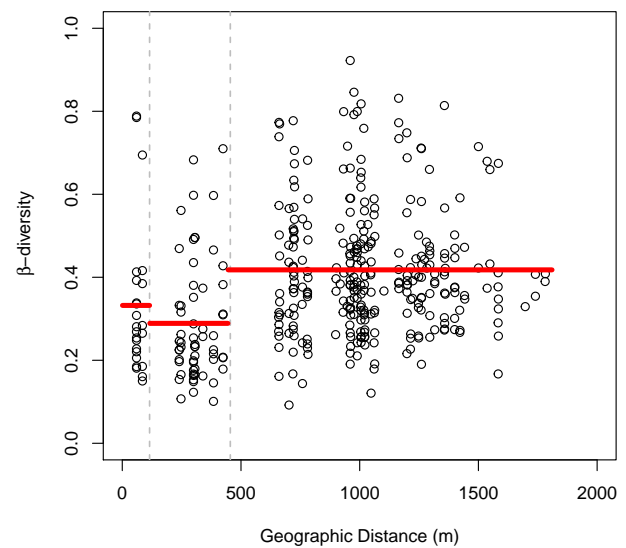
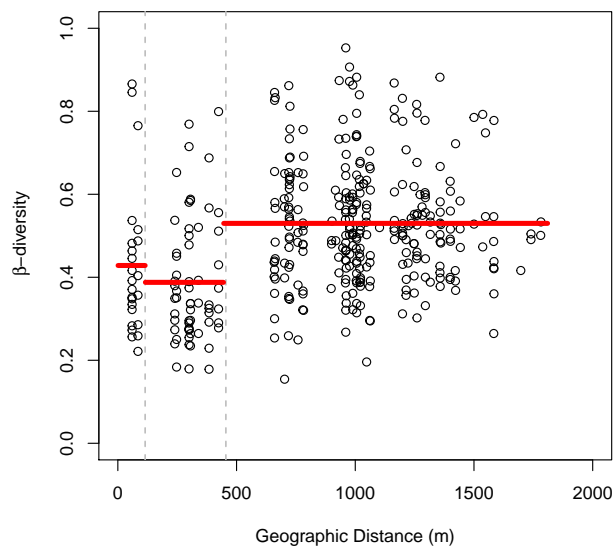


## 2. Shrub





### 3. Meadow



### Bootstrapping

```
# bootstrapping with 2000 replications
jforres <- boot(data = as.data.frame(m.fordata), statistic = bs.jac, R = 3000,
               formula = (1 - Jabd) ~ ipcc)
sforres <- boot(data = as.data.frame(m.fordata), statistic = bs.sor, R = 3000,
               formula = (1 - Labd) ~ ipcc)
jmixres <- boot(data = as.data.frame(m.mixdata), statistic = bs.jac, R = 3000,
               formula = (1 - Jabd) ~ ipcc)
smixres <- boot(data = as.data.frame(m.mixdata), statistic = bs.sor, R = 3000,
               formula = (1 - Labd) ~ ipcc)

jmeares <- boot(data = as.data.frame(m.meadata), statistic = bs.jac, R = 3000,
               formula = (1 - Jabd) ~ ipcc)
smeares <- boot(data = as.data.frame(m.meadata), statistic = bs.sor, R = 3000,
               formula = (1 - Labd) ~ ipcc)
```

## Confidence Interval

- Bootstrap 95% CI for regression coefficients

### *Jaccard*

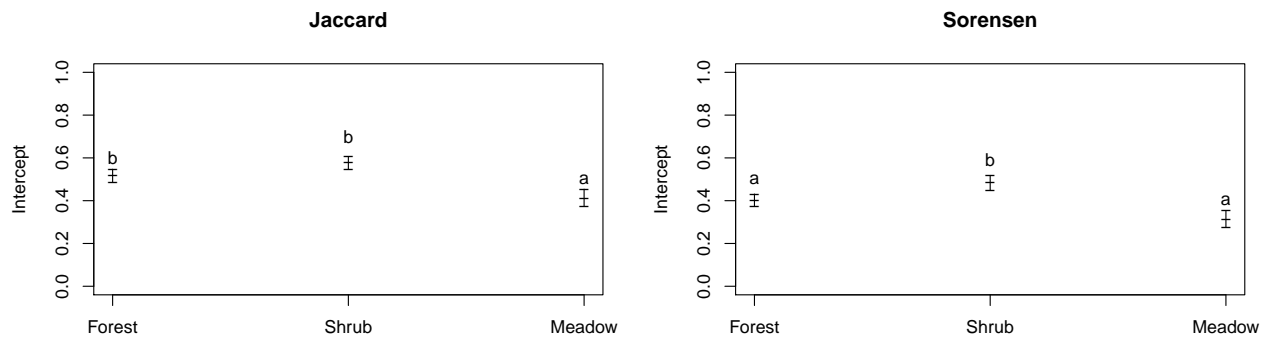
```
jforres.ci <- ci.boot(jforres)
jmixres.ci <- ci.boot(jmixres)
jmeares.ci <- ci.boot(jmeares)
```

### *Sorensen*

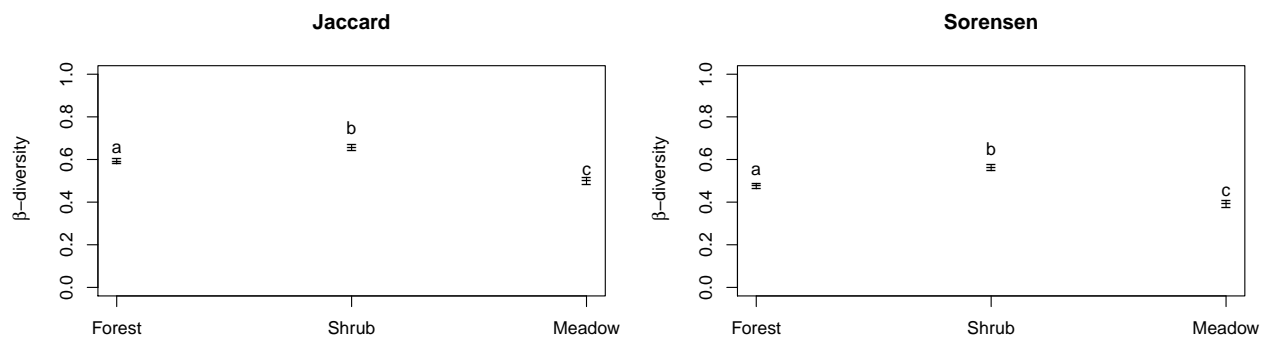
```
jforres.ci <- ci.boot(jforres)
jmixres.ci <- ci.boot(jmixres)
jmeares.ci <- ci.boot(jmeares)
```

## Results

- Intercept



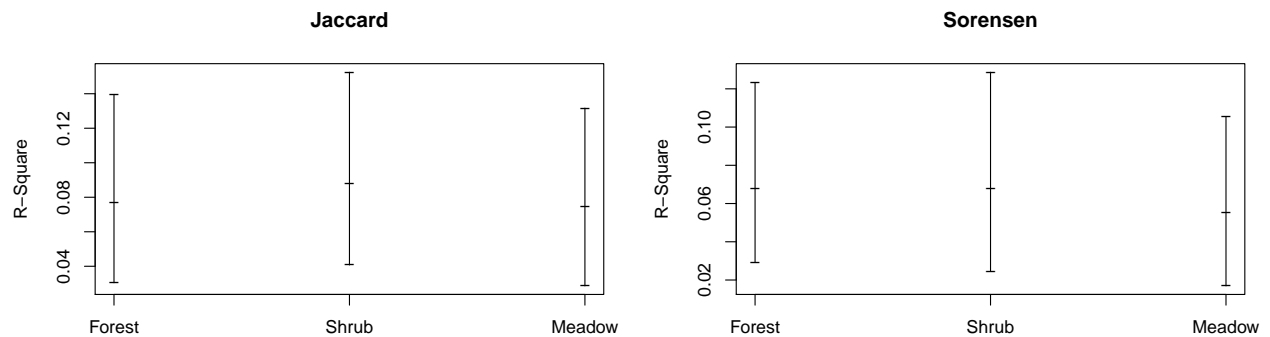
- Dissimilarity



- Slope



- R-Square



## BIOTOPE APPROACH

### within landscapes

```
agr.mea <- landsc_within(b.div.mean, ma, ma)
shr.shr <- landsc_within(b.div.mean, sm, sm)
tal.shr <- landsc_within(b.div.mean, tm, tm)
for.woo <- landsc_within(b.div.mean, wf, wf)
mix.shr <- landsc_within(b.div.mean, c(sm, tm), c(sm, tm))
```

This function creates a database for each landscape. Also, I analyzed separately short and tall shrub within the shrub-dominated landscape.

In this analysis only the biotopes that correspond the type of the landscape were considered.

### Regression models

- Jaccard Estimator

```
jmodelfor.woo <- lm((1 - for.woo[, 4]) ~ for.woo[, 6])
jmodelmix.shr <- lm((1 - mix.shr[, 4]) ~ mix.shr[, 6])
jmodelagr.mea <- lm((1 - agr.mea[, 4]) ~ agr.mea[, 6])
jmodelshr.shr <- lm((1 - shr.shr[, 4]) ~ shr.shr[, 6])
jmodeltal.shr <- lm((1 - tal.shr[, 4]) ~ tal.shr[, 6])
```

- Sorensen Estimator

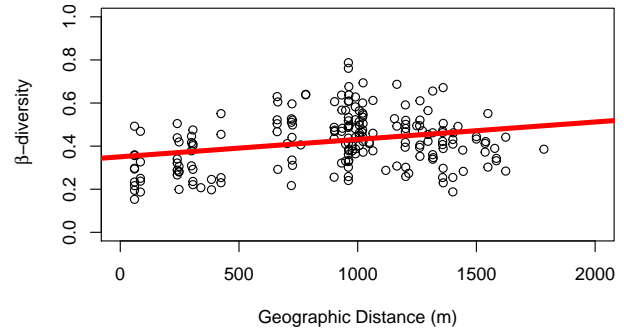
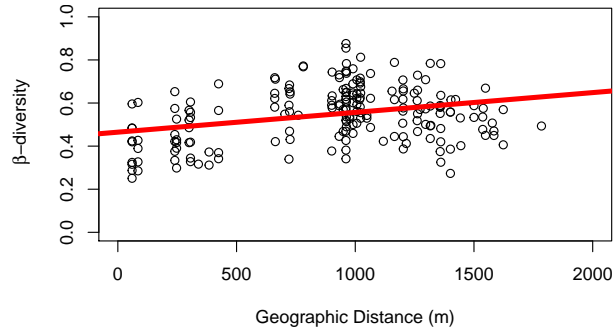
```

smodelfor.woo <- lm((1 - for.woo[, 5]) ~ for.woo[, 6])
smodelmix.shr <- lm((1 - mix.shr[, 5]) ~ mix.shr[, 6])
smodelagr.mea <- lm((1 - agr.mea[, 5]) ~ agr.mea[, 6])
smodelshr.shr <- lm((1 - shr.shr[, 5]) ~ shr.shr[, 6])
smodeltal.shr <- lm((1 - tal.shr[, 5]) ~ tal.shr[, 6])

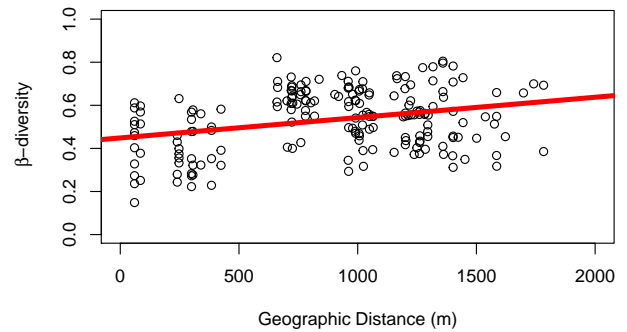
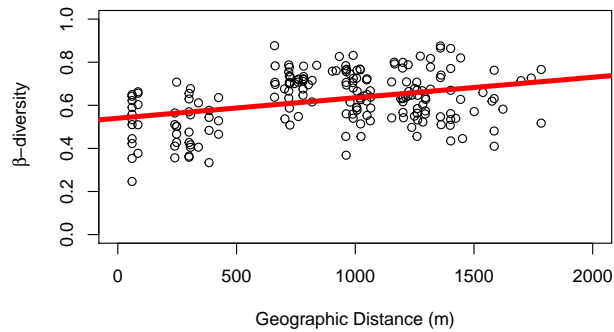
```

- Short and tall shrubs are together in Shrub (right side: Jaccard and left side: Sorensen)

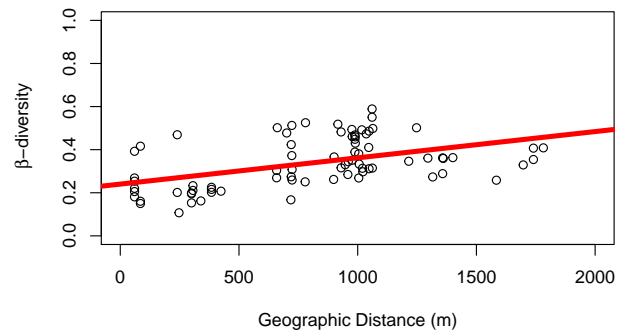
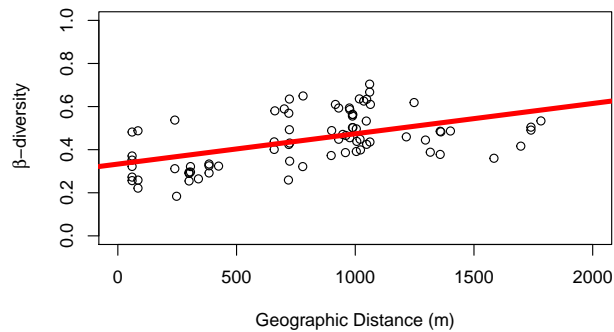
#### 1. Forest



#### 2. Shrub

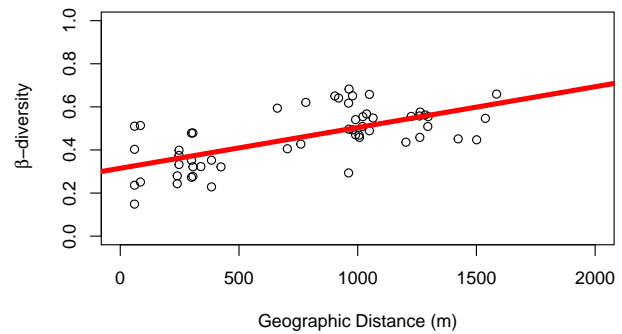
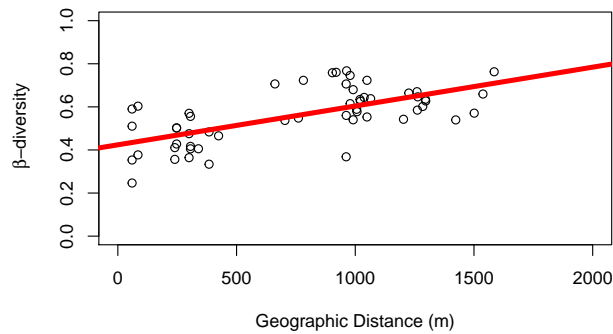


#### 3. Meadow

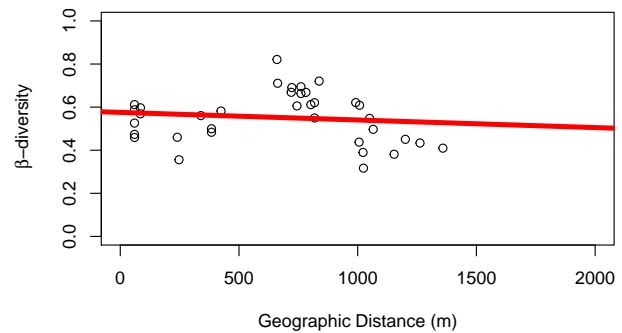
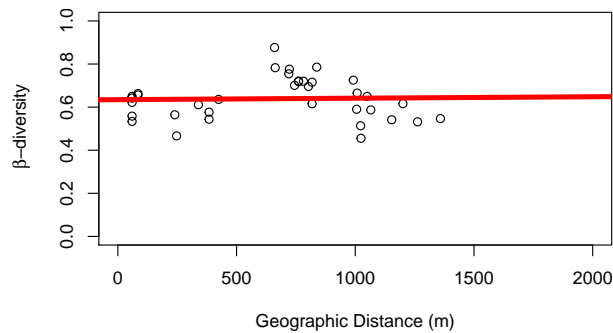


- Short and tall shrubs are separated (right side: Jaccard and left side: Sorensen)

#### 4. Tall Shrub



## 5. Short Shrub



between landscapes

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

This function creates a database of pairs of biotypes of the same type, but are in different landscapes. Also, I analyzed separately short and tall shrub among landscapes.

```
mea <- landsc_between(b.div.mean, ma, c(mm, mf))
shr <- landsc_between(b.div.mean, sm, c(sa, sf))
tal <- landsc_between(b.div.mean, tm, c(ta, tf))
woo <- landsc_between(b.div.mean, wf, c(wa, wm))
mix <- rbind(shr, tal)
```

## Regression models

- Jaccard Estimator

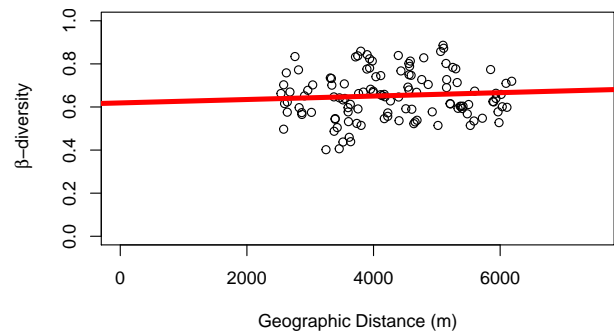
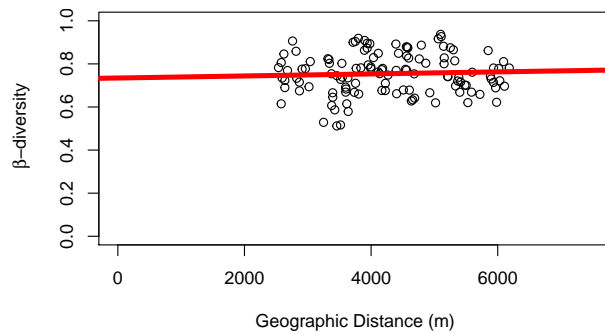
```
jmodelwoo <- lm((1 - woo[, 4]) ~ woo[, 6])
jmodelmix <- lm((1 - mix[, 4]) ~ mix[, 6])
jmodelmea <- lm((1 - mea[, 4]) ~ mea[, 6])
jmodelshr <- lm((1 - shr[, 4]) ~ shr[, 6])
jmodeltal <- lm((1 - tal[, 4]) ~ tal[, 6])
```

- Sorensen Estimator

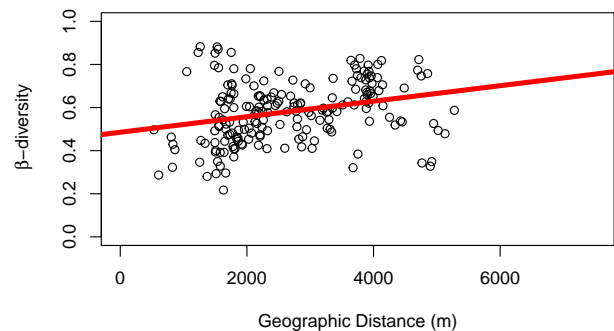
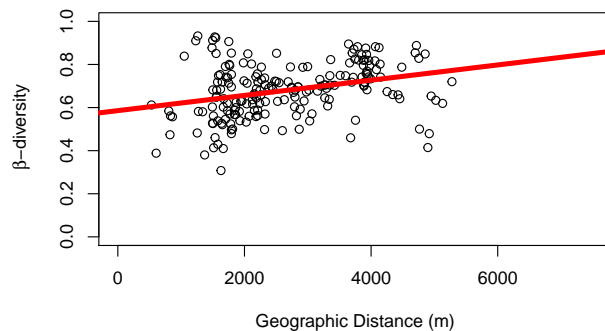
```
smodelwoo <- lm((1 - woo[, 5]) ~ woo[, 6])
smodelmix <- lm((1 - mix[, 5]) ~ mix[, 6])
smodelmea <- lm((1 - mea[, 5]) ~ mea[, 6])
smodelshr <- lm((1 - shr[, 5]) ~ shr[, 6])
smodeltal <- lm((1 - tal[, 5]) ~ tal[, 6])
```

- Short and tall shrubs are together in Shrub (right side: Jaccard and left side: Sorensen)

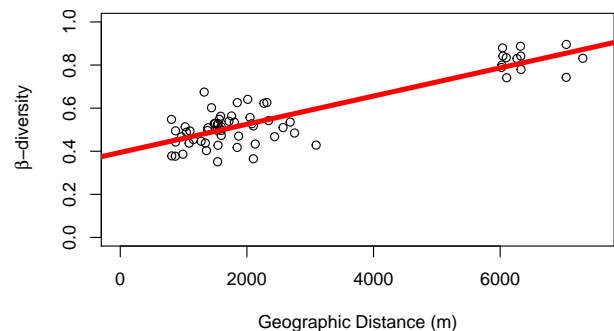
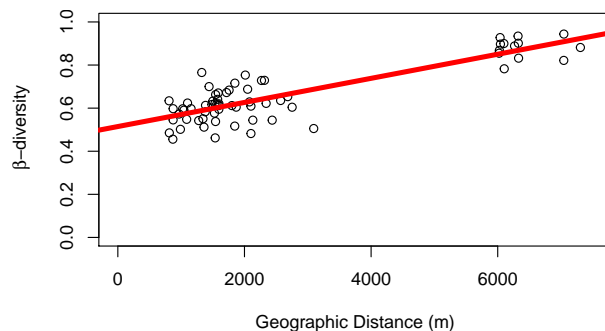
### 1. Forest



### 2. Shrub

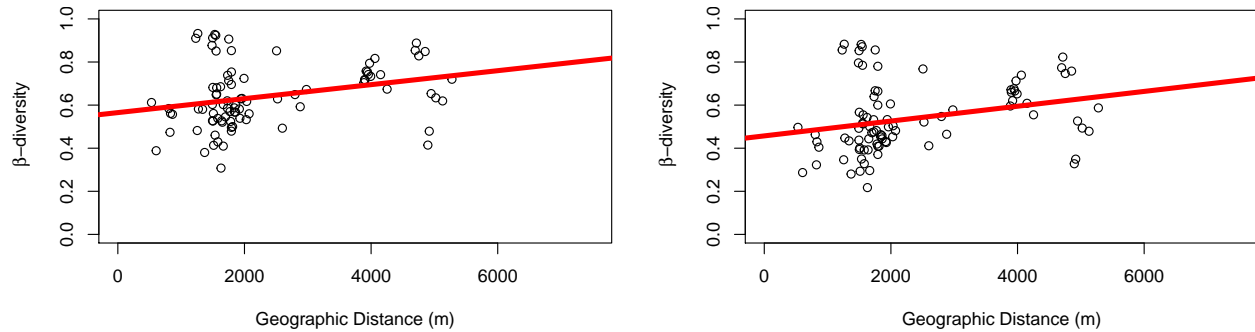


### 3. Meadow

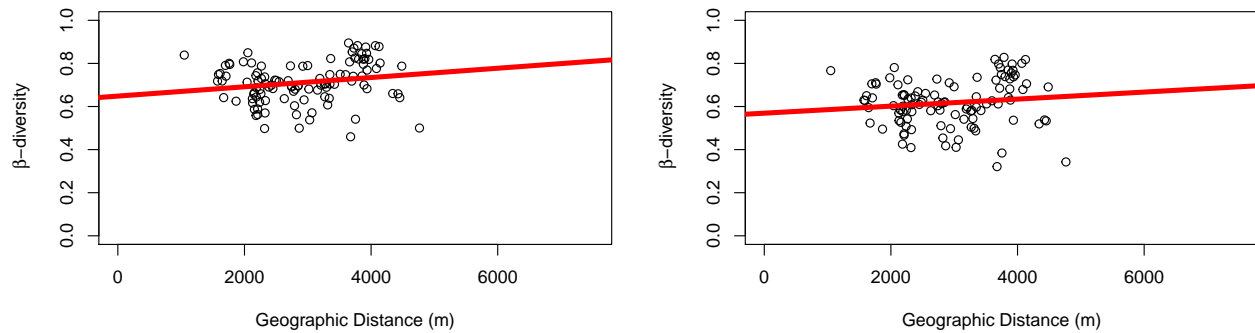


- Short and tall shrubs are separated (right side: Jaccard and left side: Sorensen)

#### 4. Tall Shrub



#### 5. Short Shrub



### Biotores

```
mfma <- c(mf, mm, ma)
sfma <- c(sf, sm, sa)
tfma <- c(tf, tm, ta)
wfma <- c(wf, wm, wa)
mixf <- c(sf, sm, sa, tf, tm, ta)
```

This function creates database for each biotope independent landscape.

```
mea.bio <- landsc_within(b.div.mean, mfma, mfma)
sho.bio <- landsc_within(b.div.mean, sfma, sfma)
tal.bio <- landsc_within(b.div.mean, tfma, tfma)
woo.bio <- landsc_within(b.div.mean, wfma, wfma)
mix.bio <- landsc_within(b.div.mean, mixf, mixf)
```

### Regression models

- Jaccard Estimator

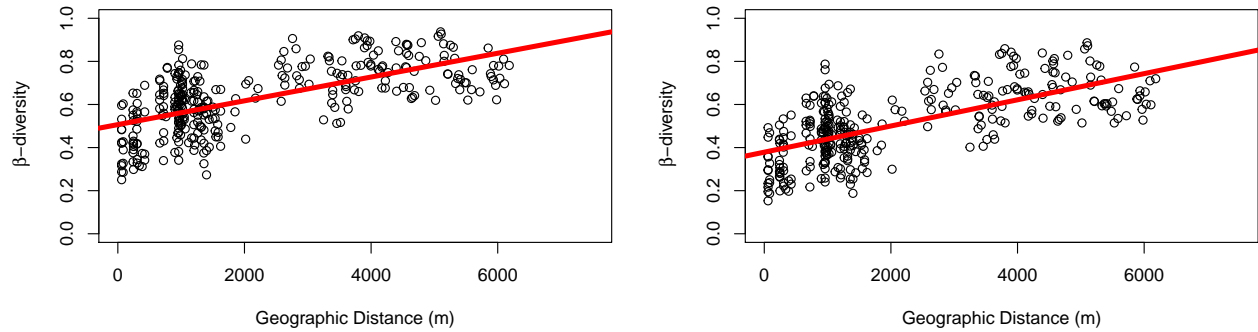
```
jmodelwoo.bio <- lm((1 - woo.bio[, 4]) ~ woo.bio[, 6])
jmodelmix.bio <- lm((1 - mix.bio[, 4]) ~ mix.bio[, 6])
jmodelmea.bio <- lm((1 - mea.bio[, 4]) ~ mea.bio[, 6])
jmodelsho.bio <- lm((1 - sho.bio[, 4]) ~ sho.bio[, 6])
jmodeltal.bio <- lm((1 - tal.bio[, 4]) ~ tal.bio[, 6])
```

- Sorensen Estimator

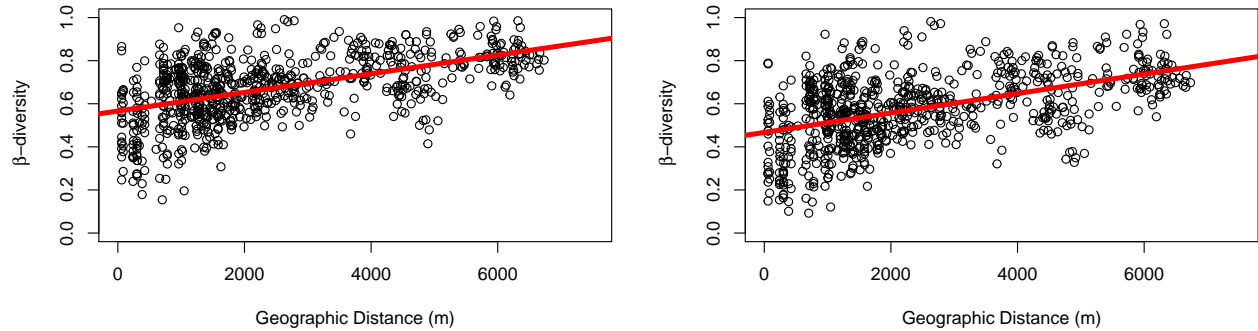
```
smodelwoo.bio <- lm((1 - woo.bio[, 5]) ~ woo.bio[, 6])
smodelmix.bio <- lm((1 - mix.bio[, 5]) ~ mix.bio[, 6])
smodelmea.bio <- lm((1 - mea.bio[, 5]) ~ mea.bio[, 6])
smodelsho.bio <- lm((1 - sho.bio[, 5]) ~ sho.bio[, 6])
smodeltal.bio <- lm((1 - tal.bio[, 5]) ~ tal.bio[, 6])
```

- Short and tall shrubs are together (right side: Jaccard and left side: Sorensen)

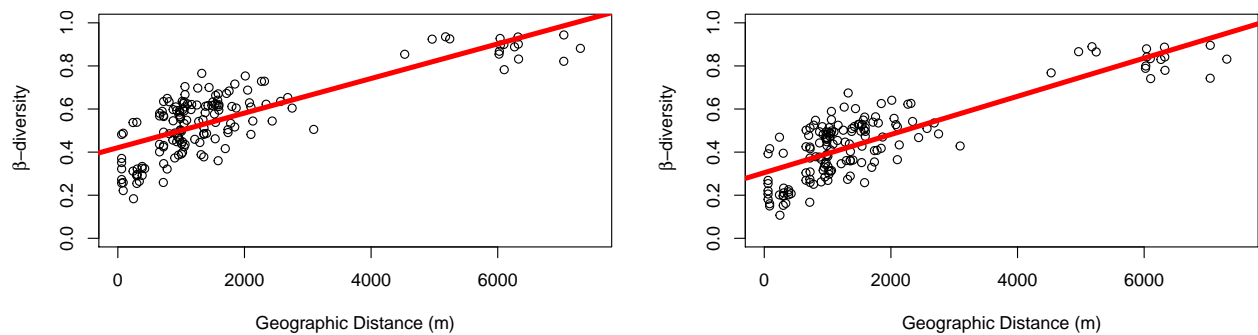
### 1. Forest



### 2. Shrub



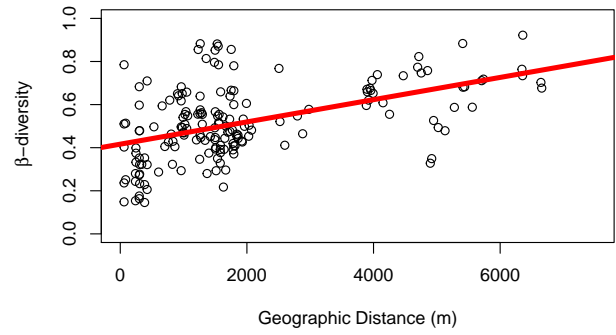
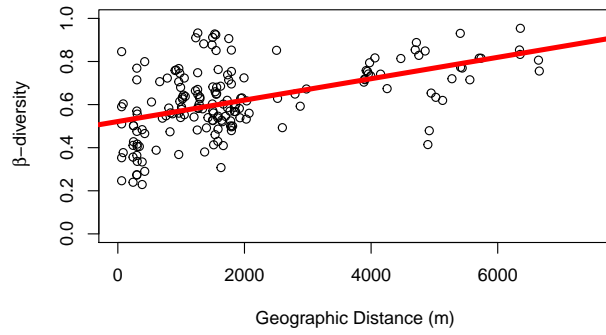
### 3. Meadow



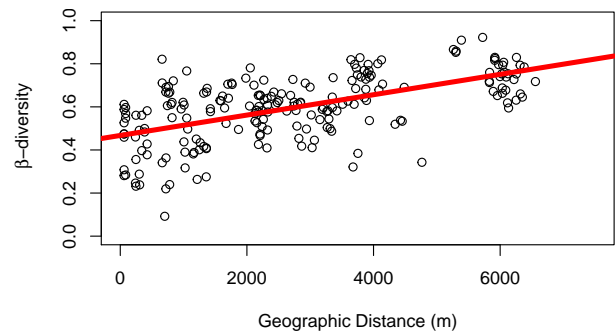
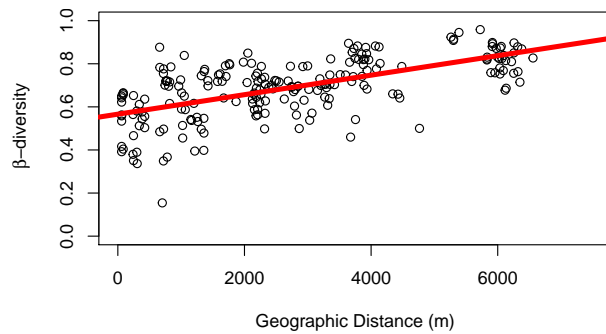
- Short and tall shrubs are separated (right side: Jaccard and left side: Sorensen)



#### 4. Tall Shrub



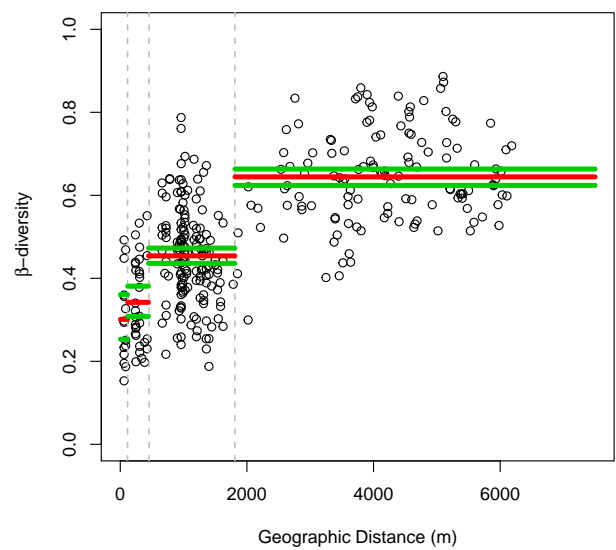
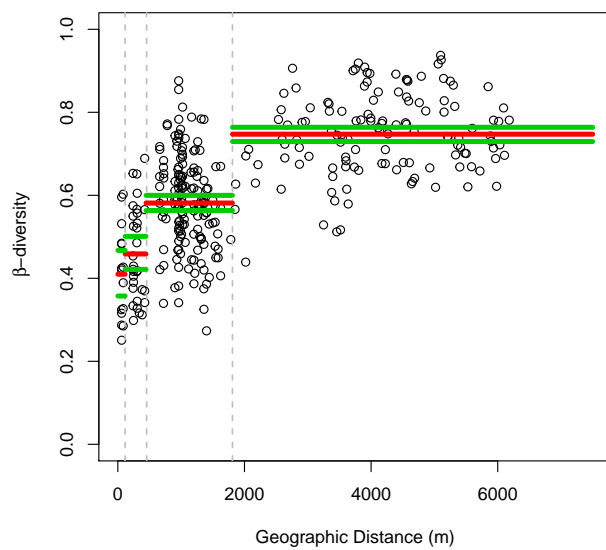
#### 5. Short Shrub



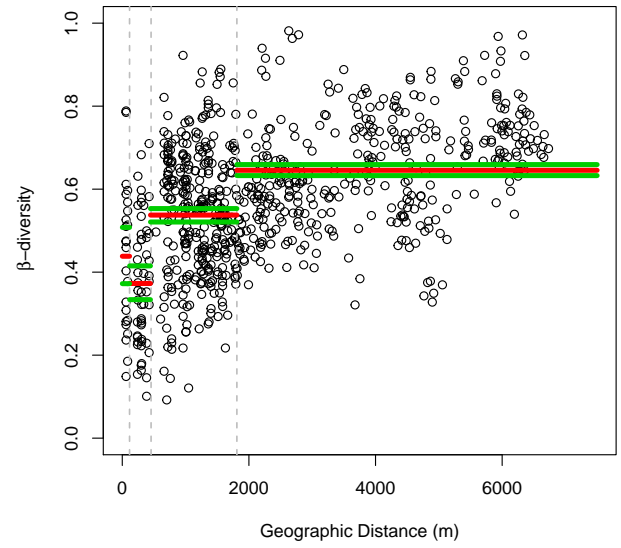
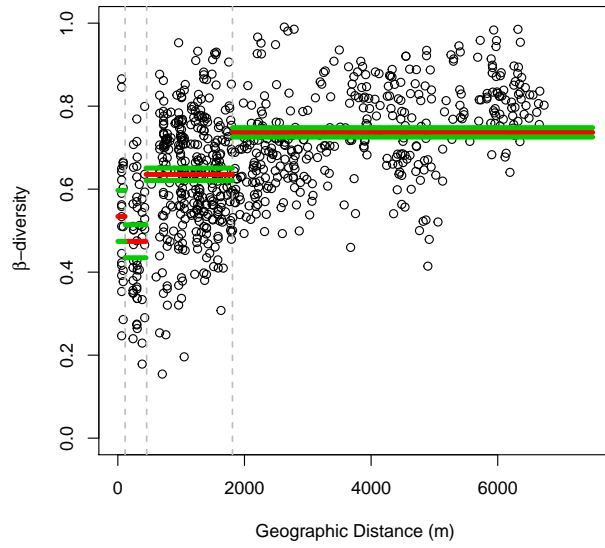
### Bootstrapping 1

#### Mean Dissimilarity for each scale

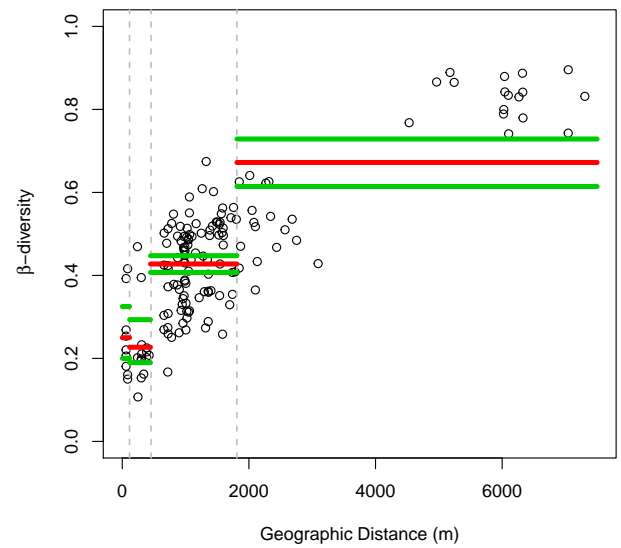
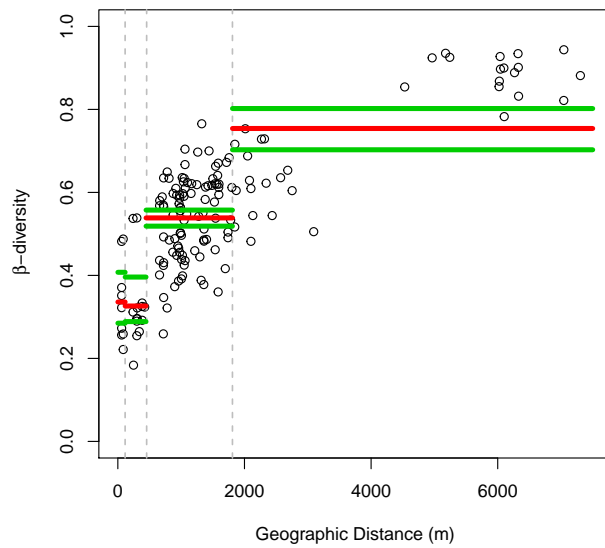
##### 1. Forest



##### 2. Shrub

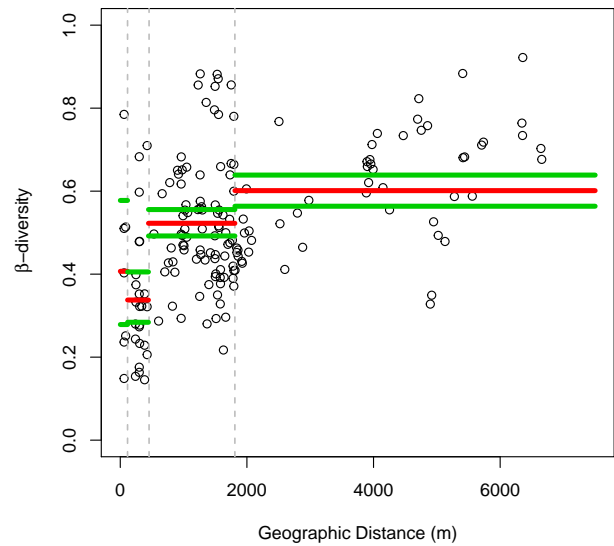
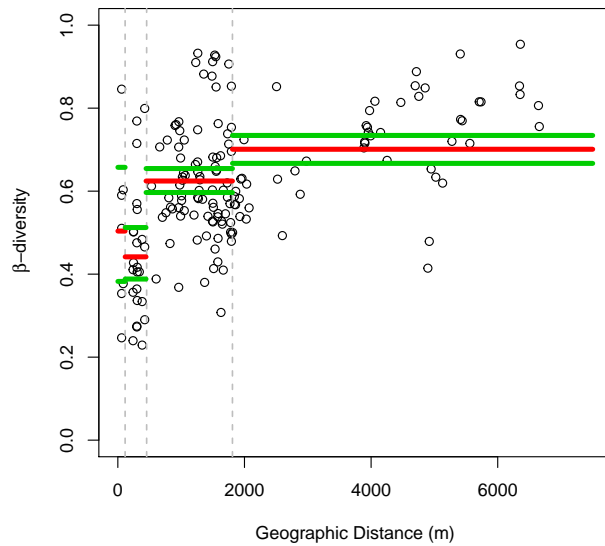


### 3. Meadow

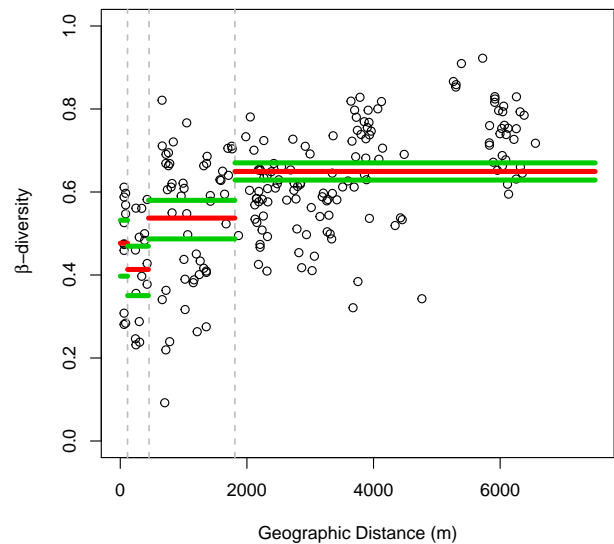
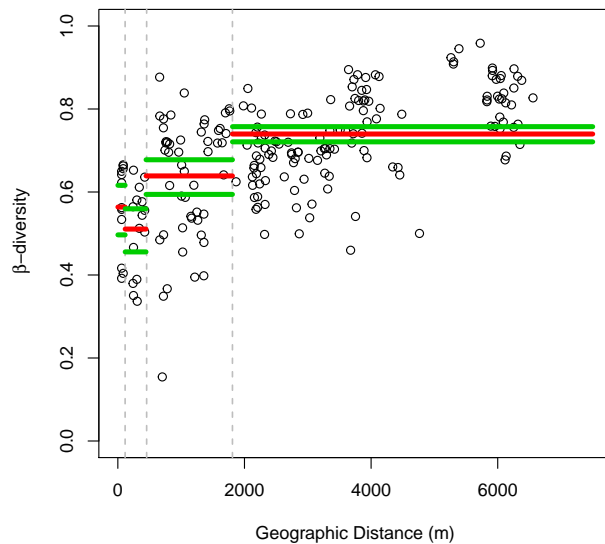


- Mean Dissimilarity for each scale (with tall and short shrubs separately)

### 4. Tall Shrub



## 5. Short Shrub



## Bootstrapping 2

### Within Landscape

```
# bootstrapping with 2000 replications
wl.jforres <- boot(data = as.data.frame(for.woo), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
wl.sforres <- boot(data = as.data.frame(for.woo), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

wl.jmixres <- boot(data = as.data.frame(mix.shr), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
wl.smixres <- boot(data = as.data.frame(mix.shr), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)
```

```

wl.jtalres <- boot(data = as.data.frame(tal.shr), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
wl.stalres <- boot(data = as.data.frame(tal.shr), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

wl.jshrres <- boot(data = as.data.frame(shr.shr), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
wl.sshrres <- boot(data = as.data.frame(shr.shr), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

wl.jmeares <- boot(data = as.data.frame(agr.mea), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
wl.smeares <- boot(data = as.data.frame(agr.mea), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

```

## Between Landscapes

```

# bootstrapping with 2000 replications
bl.jwoores <- boot(data = as.data.frame(woo), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
bl.swoores <- boot(data = as.data.frame(woo), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

bl.jmixres <- boot(data = as.data.frame(mix), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
bl.smixres <- boot(data = as.data.frame(mix), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

bl.jtalres <- boot(data = as.data.frame(tal), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
bl.stalres <- boot(data = as.data.frame(tal), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

bl.jshrres <- boot(data = as.data.frame(shr), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
bl.sshrres <- boot(data = as.data.frame(shr), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

bl.jmeares <- boot(data = as.data.frame(mea), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
bl.smeares <- boot(data = as.data.frame(mea), statistic = bs.sor, R = 3000,
  formula = (1 - Labd) ~ ipcc)

```

## All Biotopes

```

# bootstrapping with 2000 replications
al.jforres <- boot(data = as.data.frame(woo.bio), statistic = bs.jac, R = 3000,
  formula = (1 - Jabd) ~ ipcc)
al.sforres <- boot(data = as.data.frame(woo.bio), statistic = bs.sor, R = 3000,

```

```

        formula = (1 - Labd) ~ ipcc)

al.jmixres <- boot(data = as.data.frame(mix.bio), statistic = bs.jac, R = 3000,
                  formula = (1 - Jabd) ~ ipcc)
al.smixres <- boot(data = as.data.frame(mix.bio), statistic = bs.sor, R = 3000,
                  formula = (1 - Labd) ~ ipcc)

al.jtalres <- boot(data = as.data.frame(tal.bio), statistic = bs.jac, R = 3000,
                  formula = (1 - Jabd) ~ ipcc)
al.stalres <- boot(data = as.data.frame(tal.bio), statistic = bs.sor, R = 3000,
                  formula = (1 - Labd) ~ ipcc)

al.jshrres <- boot(data = as.data.frame(sho.bio), statistic = bs.jac, R = 3000,
                  formula = (1 - Jabd) ~ ipcc)
al.sshrres <- boot(data = as.data.frame(sho.bio), statistic = bs.sor, R = 3000,
                  formula = (1 - Labd) ~ ipcc)

al.jmeares <- boot(data = as.data.frame(mea.bio), statistic = bs.jac, R = 3000,
                  formula = (1 - Jabd) ~ ipcc)
al.smeares <- boot(data = as.data.frame(mea.bio), statistic = bs.sor, R = 3000,
                  formula = (1 - Labd) ~ ipcc)

```

## Confidence Interval

### Within Landscape - get 95% confidence intervals

#### *Jaccard Estimator*

```

wl.jforres.ci <- ci.boot(wl.jforres)
wl.jmixres.ci <- ci.boot(wl.jmixres)
wl.jtalres.ci <- ci.boot(wl.jtalres)
wl.jshrres.ci <- ci.boot(wl.jshrres)

```

```
## Warning: extreme order statistics used as endpoints
```

```
wl.jmeares.ci <- ci.boot(wl.jmeares)
```

#### *Sorensen Estimator*

```

wl.sforres.ci <- ci.boot(wl.sforres)
wl.smixres.ci <- ci.boot(wl.smixres)
wl.stalres.ci <- ci.boot(wl.stalres)
wl.sshrres.ci <- ci.boot(wl.sshrres)
wl.smeares.ci <- ci.boot(wl.smeares)

```

### Between Landscapes - get 95% confidence intervals

#### *Jaccard Estimator*

```
bl.jforres.ci <- ci.boot(bl.jwoores)
bl.jmixres.ci <- ci.boot(bl.jmixres)
bl.jtalres.ci <- ci.boot(bl.jtalres)
bl.jshrres.ci <- ci.boot(bl.jshrres)
bl.jmeares.ci <- ci.boot(bl.jmeares)
```

### *Sorensen Estimator*

```
bl.sforres.ci <- ci.boot(bl.swoores)
bl.smixres.ci <- ci.boot(bl.smixres)
bl.stalres.ci <- ci.boot(bl.stalres)
bl.sshrres.ci <- ci.boot(bl.sshrres)
bl.smeares.ci <- ci.boot(bl.smeares)
```

## All Biotopes - get 95% confidence intervals

### *Jaccard Estimator*

```
al.jforres.ci <- ci.boot(al.jforres)
al.jmixres.ci <- ci.boot(al.jmixres)
al.jtalres.ci <- ci.boot(al.jtalres)
al.jshrres.ci <- ci.boot(al.jshrres)
al.jmeares.ci <- ci.boot(al.jmeares)
```

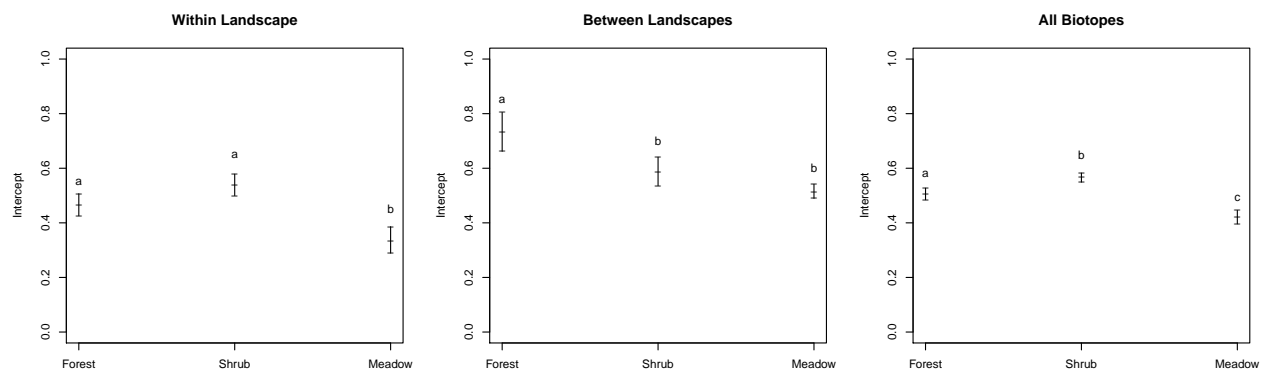
### *Sorensen Estimator*

```
al.sforres.ci <- ci.boot(al.sforres)
al.smixres.ci <- ci.boot(al.smixres)
al.stalres.ci <- ci.boot(al.stalres)
al.sshrres.ci <- ci.boot(al.sshrres)
al.smeares.ci <- ci.boot(al.smeares)
```

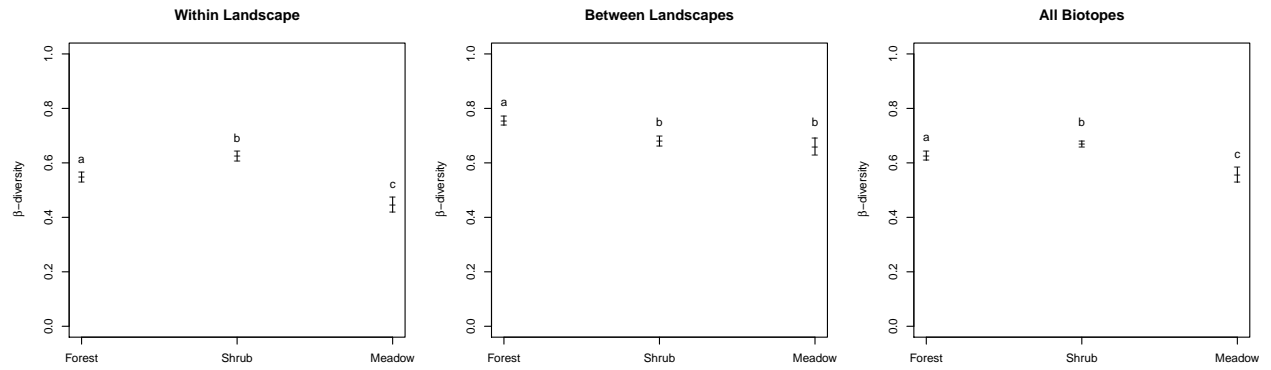
## Results 1

### Jaccard

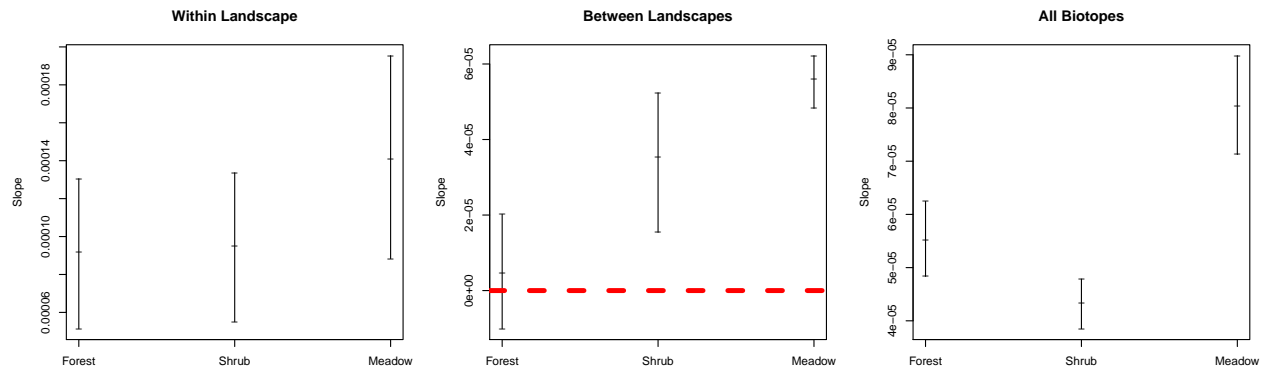
- Intercept



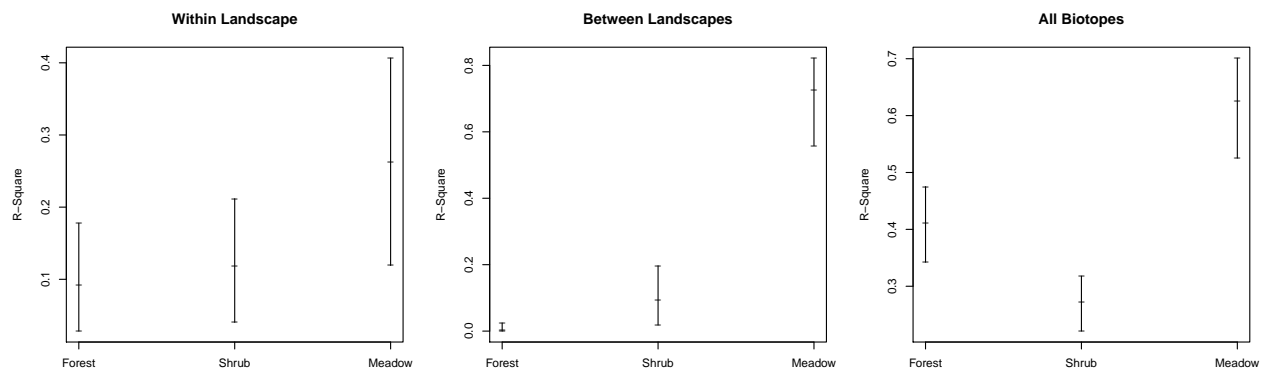
- Dissimilarity



- Slope

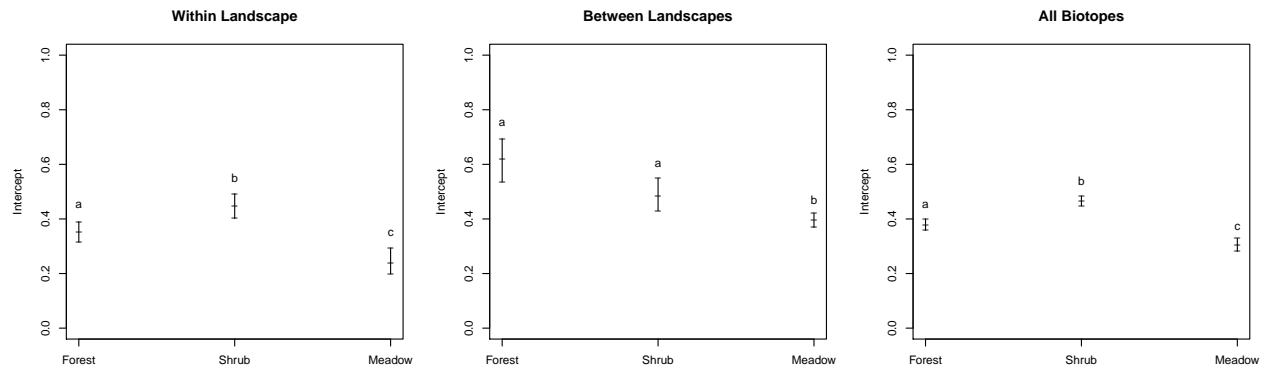


- R-Square

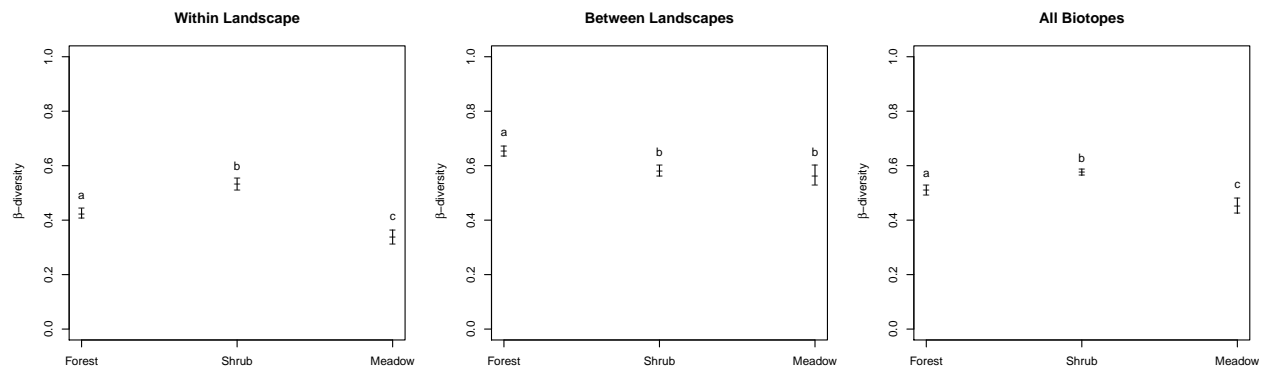


## Sorensen

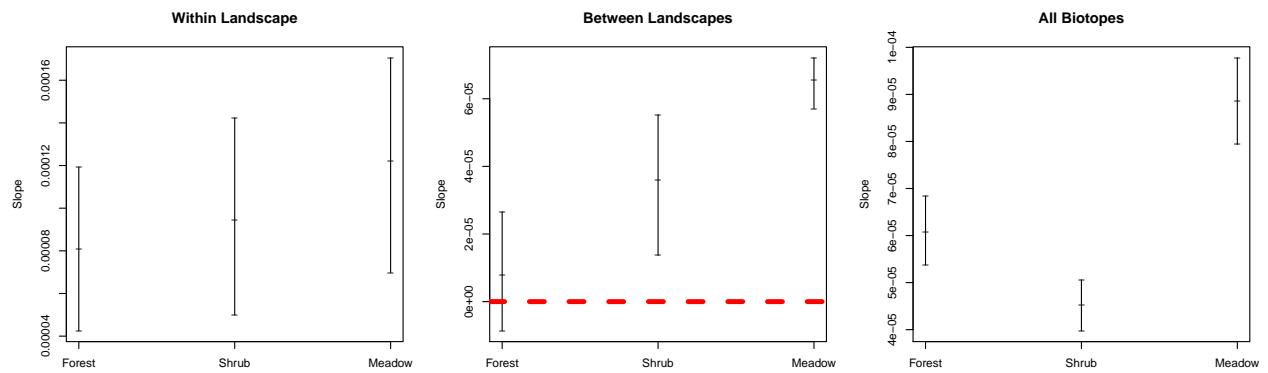
- Intercept



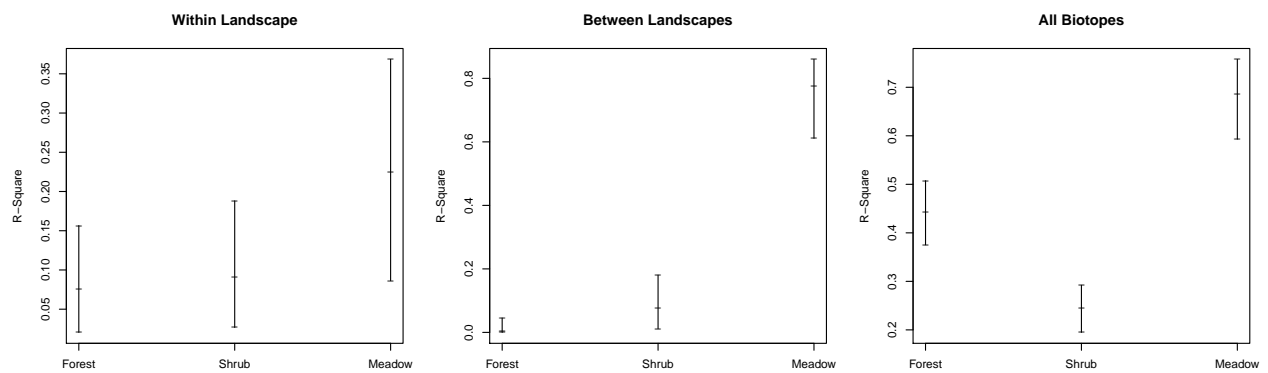
- Dissimilarity



- Slope



- R-Square

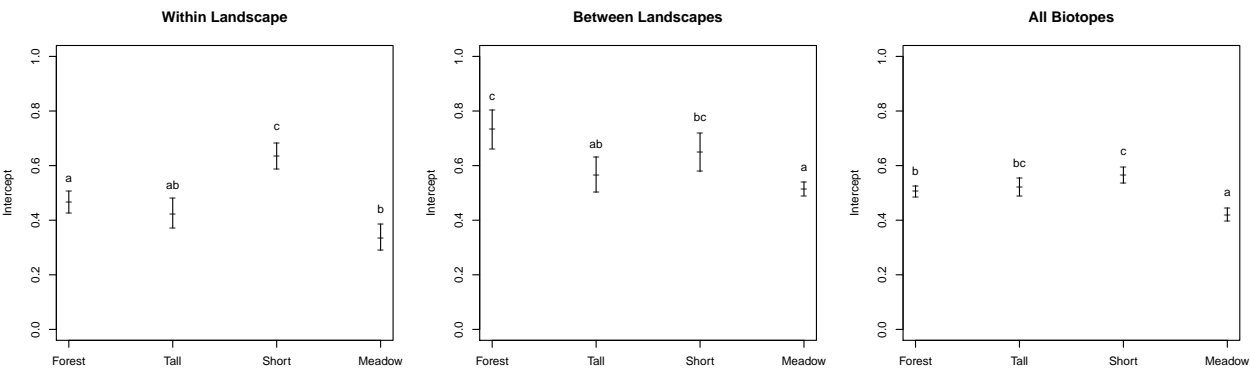




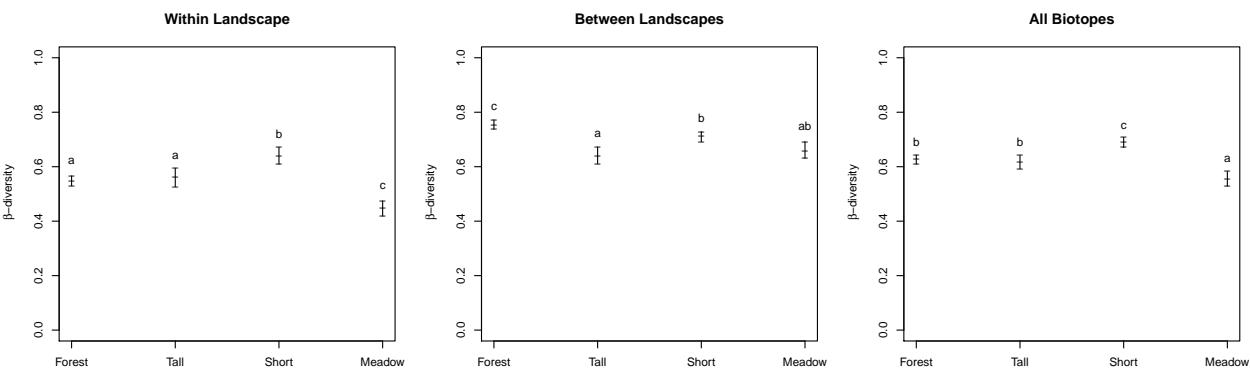
Results 2

Jaccard

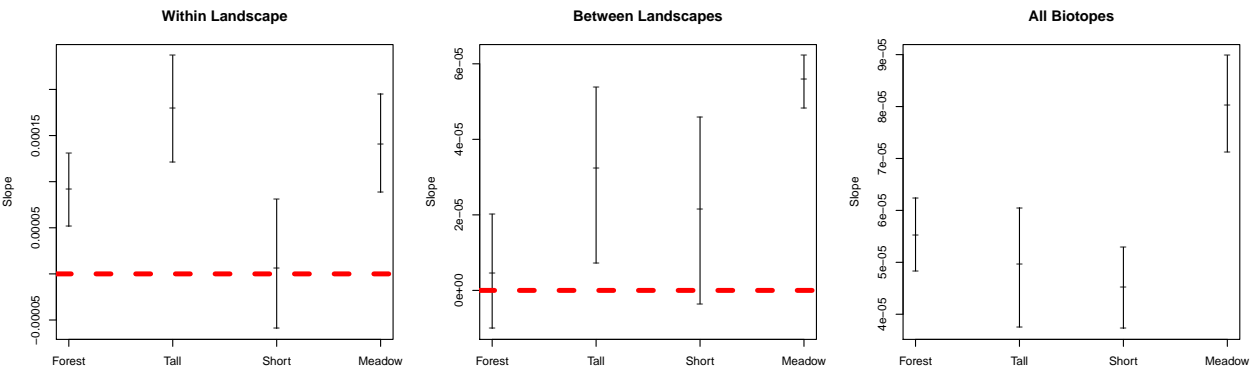
- Intercept



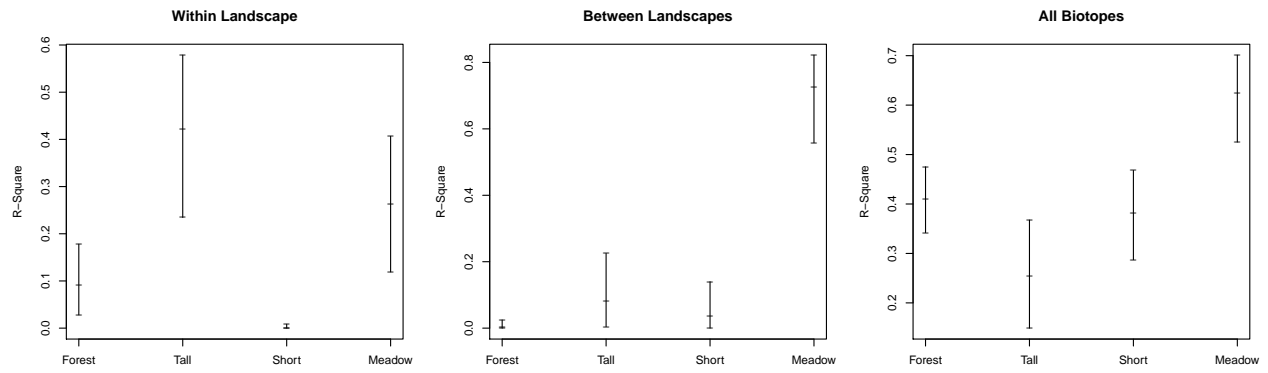
- Dissimilarity



- Slope

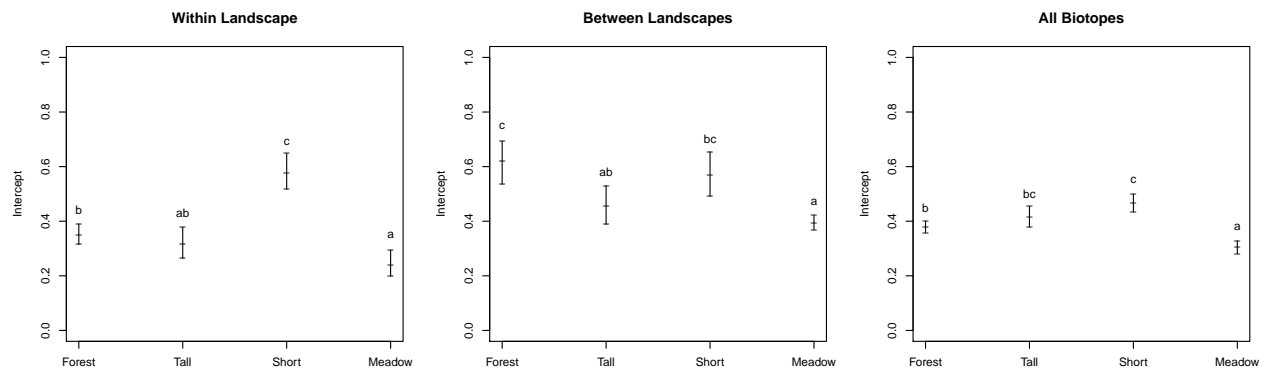


- R-Square

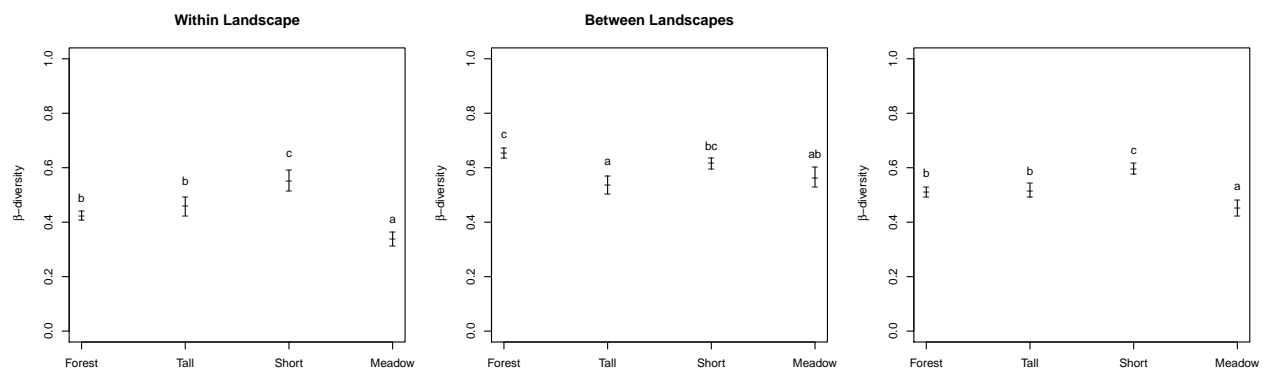


## Sorensen

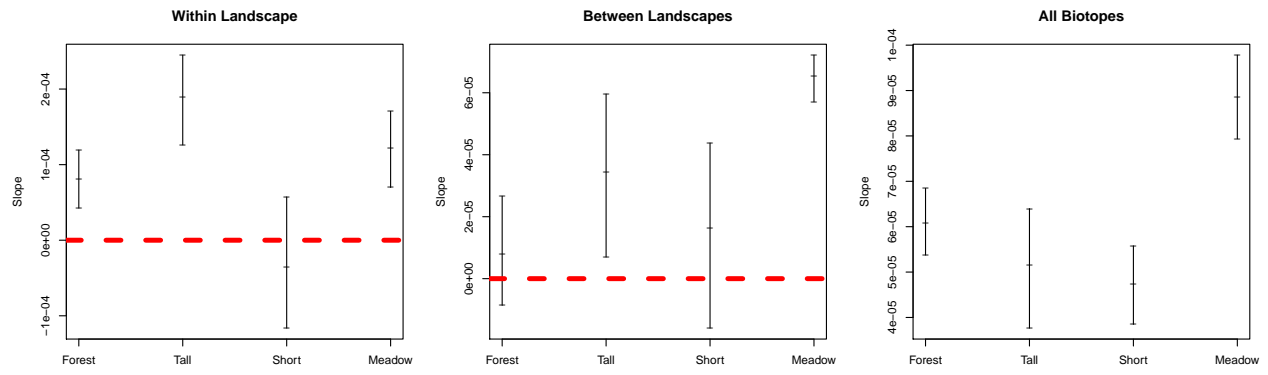
- Intercept



- Dissimilarity



- Slope



- R-Square

