

The geographical ecology of pond bacteria

April 21, 2015

```
# Load Packages
```

```
require("sp")           # Classes and methods for handling spatial data
```

```
## Loading required package: sp
```

```
require("geoR")         # Methods for geostatistical analyses
```

```
## Loading required package: geoR
## Loading required package: MASS
## -----
## Analysis of geostatistical data
## For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
## geoR version 1.7-4.1 (built on 2012-06-29) is now loaded
## -----
```

```
require("rgdal")        # Geospatial Data Abstraction Library
```

```
## Loading required package: rgdal
## rgdal: version: 0.9-1, (SVN revision 518)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 1.11.1, released 2014/09/24
## Path to GDAL shared files: /usr/local/Cellar/gdal/1.11.1_3/share/gdal
## Loaded PROJ.4 runtime: Rel. 4.8.0, 6 March 2012, [PJ_VERSION: 480]
## Path to PROJ.4 shared files: (autodetected)
```

```
require("raster")       # Methods to create a RasterLayer object
```

```
## Loading required package: raster
##
## Attaching package: 'raster'
##
## The following objects are masked from 'package:MASS':
##
## area, select
```

```
require("maptools")     # Tools for manipulating and reading geospatial data
```

```
## Loading required package: maptools
## Checking rgeos availability: TRUE
```

```
require("picante")
```

```
## Loading required package: picante
## Loading required package: ape
```

```
##
## Attaching package: 'ape'
##
## The following objects are masked from 'package:raster':
##
##     rotate, zoom
##
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.2-1
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:raster':
##
##     getData
```

```
require("ape")
require("seqinr")
```

```
## Loading required package: seqinr
## Loading required package: ade4
##
## Attaching package: 'ade4'
##
## The following object is masked from 'package:vegan':
##
##     cca
##
## Attaching package: 'seqinr'
##
## The following object is masked from 'package:nlme':
##
##     gls
##
## The following object is masked from 'package:permute':
##
##     getType
##
## The following objects are masked from 'package:ape':
##
##     as.alignment, consensus
```

```
require("vegan") # biodiversity estimators and related functions
require("fossil")
```

```
## Loading required package: fossil
## Loading required package: maps
## Loading required package: shapefiles
## Loading required package: foreign
```

```
##
## Attaching package: 'shapefiles'
##
## The following objects are masked from 'package:foreign':
##
##     read.dbf, write.dbf
```

```
require("simba")
```

```
## Loading required package: simba
## This is simba 0.3-5
##
## Attaching package: 'simba'
##
## The following object is masked from 'package:picante':
##
##     mpd
##
## The following object is masked from 'package:stats':
##
##     mad
```

```
require("reshape")
```

```
## Loading required package: reshape
```

```
require("knitr")
```

```
## Loading required package: knitr
```

```
source("~/GitHub/Dimensions/Aim3/bin/MothurTools.R")
```

C. Load Source Code

In addition to relying on contributed packages, we will also be using a source code file. A source code file has user-defined functions that are required for certain analyses. The benefit of source files is that they contain “vetted” code that can be used across multiple projects. Here, we will be using a source code file that includes a function for reading in the output files from the popular community sequencing software `mothur` (<http://www.mothur.org/>).

```
load(file = "~/GitHub/Dimensions/Aim3/Mothur/INPond_Initial.RData")
```

Overview

Here, we will explore our primary geographical patterns of interest: the taxa-area relationship (TAR), the phylogenetic diversity-area relationship, and the distance-decay relationship in taxonomic and phylogenetic community similarity.

Study area

We analyzed environmental and bacterial community data from a survey of shallow ponds found east of Bloomington, IN. These ponds were constructed in the 1940s as wildlife refuge ponds, and are scattered throughout Brown County State Park, Yellowwood State Forest, and Hoosier National Forest. In the summer of 2013, we visited approximately 50 of these ponds and recorded their geographic locations. We sampled aspects of water chemistry, physical properties, and bacterial community composition.

Figure 1. Spatially explicit data on environmental and geographic features.

```
# Load Environmental and Geographical Data
env <- read.table("~/GitHub/Dimensions/Aim3/DATA/EnvData/20130801_PondDataMod.csv", sep = ",", header = 1)
lats <- as.numeric(env[, 3]) # latitudes (north and south)
lons <- as.numeric(env[, 4]) # longitudes (east and west)
```

Environmental data

We measured 19 environmental and geographic variables. These included elevation (m), geographical coordinates (lat-long; data: WGS84), temperature (C), Diameter(m), Depth(m), redox potential (ORP), specific conductivity or SpC (uS/cm), dissolved Oxygen (mg/L), total dissolved solids (g/L), salinity (p.s.u.=ppm), color - measured at absorbance = 660; an estimate of carbon in the water sample, chlorophyll a (ug/ml), dissolved organic carbon (mg/L), dissolved organic nitrogen (mg/L), and total phosphorus (ug/L).

Microbial community data

In addition to measuring a suite of geographic and environmental variables, we characterized the diversity of bacteria in the ponds using molecular-based approaches. Specifically, we amplified the 16S rRNA gene (i.e., “DNA”) and 16S rRNA transcripts (i.e., “RNA”) of bacteria using barcoded primers on the Illumina MiSeq platform. We then used a *mothur* pipeline to quality-trim our data set and assign sequences to operational taxonomic units (OTU).

For each pond, we used the observed taxonomic richness (S), total number of gene reads (N), and number of gene reads per OTU (Ni) to estimate Shannon’s diversity index (H), and Simpson’s evenness (D/S). We should estimate a handful of diversity and evenness metrics, as well conduct richness estimation for each site (Chao1, ACE, rarefaction, jackknife). These will provide basic diversity-related variables to explore with respect to geography and environmental conditions.

```
# Select DNA Data: Use the `grep()` Command and Rename with `gsub()`
# The active portion, based on cDNA
active.comm <- Pond97[grep("*-cDNA", rownames(Pond97)), ]
rownames(active.comm) <- gsub("\\-cDNA", "", rownames(active.comm))
rownames(active.comm) <- gsub("\\_", "", rownames(active.comm))

# The community without respect to active or not, 16S rRNA gene sequences
all.comm <- Pond97[grep("*-DNA", rownames(Pond97)), ]
rownames(all.comm) <- gsub("\\-DNA", "", rownames(all.comm))
rownames(all.comm) <- gsub("\\_", "", rownames(all.comm))

# Remove Sites Not in the Environmental Data Set
active.comm <- active.comm[rownames(active.comm) %in% env$Sample_ID, ]
all.comm <- all.comm[rownames(all.comm) %in% env$Sample_ID, ]

# Remove Zero-Occurrence Taxa
```

```

active.comm <- active.comm[ , colSums(active.comm) > 0]
all.comm <- all.comm[ , colSums(all.comm) > 0]

# Import Taxonomy Data Using `read.tax()` from Source Code
tax <- read.tax(taxonomy = "~/GitHub/Dimensions/Aim3//Mothur/INPonds.bac.final.0.03.taxonomy")

#### A function to generate observed richness
S.obs <- function(x = ""){ rowSums(x > 0) * 1}

#### For each site:
# N equals numbers of reads
env$active.N <- as.vector(rowSums(active.comm))
env$all.N <- as.vector(rowSums(all.comm))

# S equals the number of non-zero abundances
env$active.S <- S.obs(active.comm)
env$all.S <- S.obs(all.comm)

# Diversity is Shannon's
env$active.H <- as.vector(diversity(active.comm, index = "shannon"))
env$all.H <- as.vector(diversity(all.comm, index = "shannon"))

# Evenness is Simpsons; divide Simpson's Diversity by S
env$active.De <- as.vector(diversity(active.comm, index = "invsimpson")/env$active.S)
env$all.De <- as.vector(diversity(all.comm, index = "invsimpson")/env$all.S)

```

Primary geographic patterns

We examined three taxa-level geographic patterns: Distant-decay (DD), Taxa-area relationship (TAR), and the specific spatial abundance distribution (SSAD). While the DD and TAR have been more or less frequently studied in microbial ecology and microbial biogeography, the SSAD has been mainly, if not entirely examined in studies of macroscopic plants and animals.

1.) Distance Decay, taxonomic and phylogenetic

Tobler's first law of geography states that "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970). This law is a formulation of the concept of spatial autocorrelation. In short, spatial autocorrelation is the degree to which spatial variables are either clustered in space (positive autocorrelation) or over-dispersed (negative autocorrelation).

The distance-decay relationship is a primary biogeographic pattern of spatial autocorrelation, and captures the rate of decreasing similarity with increasing distance. This pattern addresses whether communities close to one another are more similar than communities that are farther away. The distance-decay pattern can also be used to address whether near environments have greater similarity than far ones. We looked at decay in both taxonomic level compositional similarity via bray-curtis (should also do for Sorensens) and phylogenetic distance via unifrac distance.

RESULTS: Distance-Decay

```

plot.new()
#par(mfrow=c(2, 2))

# Geographic Distances (Kilometers) Among Ponds
long.lat <- as.matrix(cbind(env$long, env$lat))
coord.dist <- earth.dist(long.lat, dist = TRUE)

# Taxonomic Distances Among Ponds (Bray-Curits)
active.bray.curtis.dist <- 1 - vegdist(active.comm)
all.bray.curtis.dist <- 1 - vegdist(all.comm)

# Transform All Distances Into List Format:
active.bray.curtis.dist.ls <- liste(active.bray.curtis.dist, entry = "bray.curtis")
all.bray.curtis.dist.ls <- liste(all.bray.curtis.dist, entry = "bray.curtis")
coord.dist.ls <- liste(coord.dist, entry = "geo.dist")

# Create a Data Frame from the Lists of Distances
df <- data.frame(coord.dist.ls, active.bray.curtis.dist.ls[, 3],
                 all.bray.curtis.dist.ls[, 3])

names(df)[4:5] <- c("active.bray.curtis", "all.bray.curtis")
attach(df)

# Now, let's plot the DD relationships:

# Set Initial Plot Parameters
par(mfrow=c(1, 2))#, mar = c(1, 5, 2, 1) + 0.1, oma = c(2, 0, 0, 0))

# Make Plot for Taxonomic DD
plot(coord.dist, active.bray.curtis, xlab = "", xaxt = "n", las = 1, ylim = c(0.1, 0.9),
     ylab="Bray-Curtis Similarity",
     main = "Distance Decay, Active taxa", col = "SteelBlue")

# Regression for Taxonomic DD
DD.reg.bc <- lm(active.bray.curtis ~ geo.dist)
summary(DD.reg.bc)

```

```

##
## Call:
## lm(formula = active.bray.curtis ~ geo.dist)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25218 -0.08125 -0.00485  0.06762  0.42022
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4094763  0.0058157   70.41  <2e-16 ***
## geo.dist     -0.0066754  0.0005241  -12.74  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1043 on 1273 degrees of freedom

```

```
## Multiple R-squared:  0.113, Adjusted R-squared:  0.1123
## F-statistic: 162.2 on 1 and 1273 DF,  p-value: < 2.2e-16
```

```
abline(DD.reg.bc , col = "red4")

# Make Plot for Taxonomic DD
plot(coord.dist, all.bray.curtis, xlab = "", xaxt = "n", las = 1, ylim = c(0.1, 0.9),
      ylab="Bray-Curtis Similarity",
      main = "Distance Decay, All taxa", col = "SteelBlue")

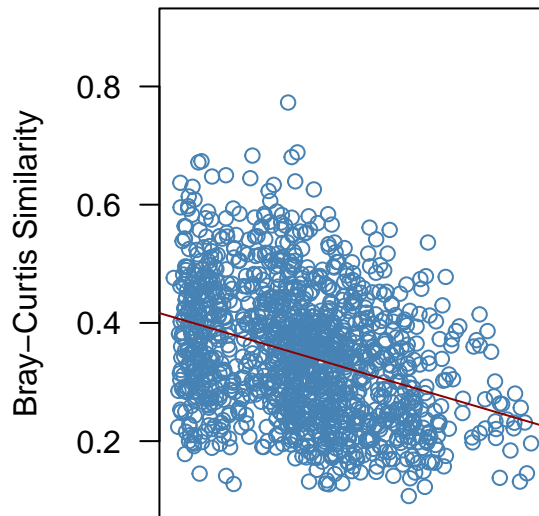
# Regression for Taxonomic DD
DD.reg.bc <- lm(all.bray.curtis ~ geo.dist)
summary(DD.reg.bc)
```

```
##
## Call:
## lm(formula = all.bray.curtis ~ geo.dist)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29942 -0.09183  0.00064  0.07955  0.48117
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4140290  0.0070007  59.141  < 2e-16 ***
## geo.dist    -0.0042916  0.0006309  -6.802 1.58e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1255 on 1273 degrees of freedom
## Multiple R-squared:  0.03507, Adjusted R-squared:  0.03432
## F-statistic: 46.27 on 1 and 1273 DF,  p-value: 1.579e-11
```

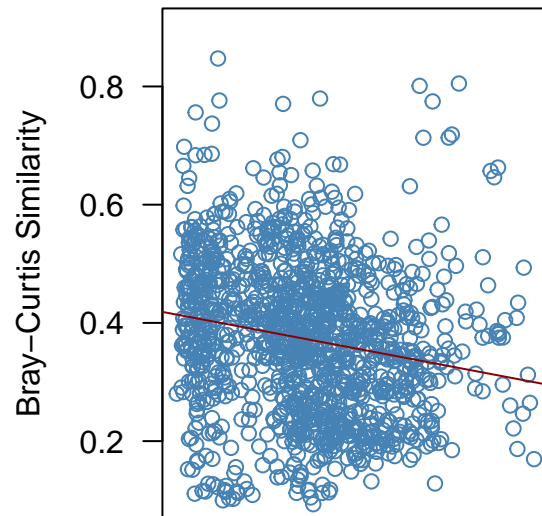
```
abline(DD.reg.bc , col = "red4")

# Add X-Axis Label to Plot
mtext("Geographic Distance, km", side = 1, adj = 0.55,
      line = 0.5, outer = TRUE)
```

Distance Decay, Active taxa



Distance Decay, All taxa



```
diffslope(geo.dist, active.bray.curtis, geo.dist, all.bray.curtis)
```

2.) Species- or taxa- area relationship (SAR)

The species-area relationship describes the rate at which species are discovered with increasing area. The SAR is one of ecology's oldest and most intensively studied patterns. Arrhenius (1921) first described the general form of the *species-area relationship (SAR)* as a power-law: $S = cA^z$ where S is species richness and A is area. Arrhenius's formula predicts a rate of increase in richness that is approximately linear in log-log space. That is, $\log(S) = c + z\log(A)$, where z is the scaling exponent.

```
# A function to generate the species-area relationship by
# Random Accumulating Sites
```

```
SAR.rand accum <- function(com){
  Alist <- c()
  Slist <- c()

  num.ponds <- c(1,2,4,6,8,12,16,24,32,42,51)
  for (i in num.ponds) {
    areas <- c() # hold iterated area values
    Ss <- c() # hold iterated S values

    for(j in 1:50){
      pond.sample <- sample(51, replace = FALSE, size = i)
      area <- 0
```



```

cum.abs <- vector(length = length(com[1, ]))

for (k in pond.sample) { # Loop through each randomly drawn pond
  area <- area + pond.areas[k] # aggregating area
  cum.abs <- cum.abs + com[k, ]
} # End random pond samples loop

Ss <- c(Ss, length(cum.abs[cum.abs > 0]))
areas <- c(areas, area)
}

Alist <- rbind(Alist, mean(areas))
Slist <- rbind(Slist, mean(Ss))
print(c(mean(areas), mean(Ss)))
}

return(cbind(log10(Alist), log10(Slist)))
}

```

*# A function to generate the species-area relationship by
accumulating area according to distance*

```

SAR.accum.dist <- function(com){
  Alist <- c()
  Slist <- c()
  num.ponds <- c(1,2,4,6,8,12,16,24,32,42,51)

  for (i in num.ponds) {
    areas <- c() # hold iterated area values
    Ss <- c() # hold iterated S values

    for(j in 1:50){
      pondID <- sample(51, size = 1)
      Area <- as.numeric(pond.areas[pondID]) # aggregating area
      cum.abs <- com[pondID, ]
      used <- c()

      for (k in 2:i) { # Loop through ponds
        sdata <- subset(coord.dist.ls, FALSE == is.element(NBX, used) & FALSE == is.element(NBY, used))
        sdata <- subset(sdata, NBX == pondID | NBY == pondID)
        sdata <- subset(sdata, geo.dist == min(sdata[, 3]))

        if (dim(sdata)[1] > 1) {
          x <- sample(dim(sdata)[1], size=1)
          sdata <- sdata[x,]
        }

        sdata <- t(as.matrix(as.numeric(as.matrix(sdata))))
        used <- c(used, as.integer(pondID))
        Area <- Area + as.numeric(pond.areas[pondID]) # aggregating area
        cum.abs <- cum.abs + com[pondID, ]

        if (sdata[1] - pondID == 0) {

```

```

        pondID <- sdata[2]
      } else {
        pondID <- sdata[1]
      }
    }
    Ss <- c(Ss, length(cum.abs[cum.abs > 0]))
    areas <- c(areas, Area)
  }
  # End random pond samples loop
  Alist <- rbind(Alist, mean(areas))
  Slist <- rbind(Slist, mean(Ss))
  print(c(mean(areas), mean(Ss)))
}
return(cbind(log10(Alist), log10(Slist)))
}

```

RESULTS: Taxa-area relationship

```

plot.new()
par(mfrow=c(1, 1))
pond.areas <- as.vector(pi * (env$Diameter/2)^2) # Find areas of all 51 ponds

sar <- SAR.accum.dist(all.comm)

```

```

## [1] 829.8636 3440.8000
## [1] 532.7978 1908.6200
## [1] 1376.148 4700.400
## [1] 1771.87 6907.58
## [1] 1915.558 8786.500
## [1] 3776.457 11868.820
## [1] 4856.689 14621.760
## [1] 6460.554 19440.900
## [1] 8557.519 22506.240
## [1] 11847.88 25561.68
## [1] 15054.54 27600.62

```

```

sar <- as.data.frame(sar)
plot(sar, xlab = "log(Area)", ylab = "log(Richness)",
     main = "Taxa-Area Relationship
     aggregating area by distance", col = "SteelBlue")

OLS <- lm(sar$V2 ~ sar$V1)
abline(OLS, col = "SteelBlue", lw = 2)
slope <- round(coefficients(OLS)[2], 3)
legend("bottomright", legend = paste("slope(All) =", slope),
      bty = "n", lw = 2, col = "SteelBlue")

sar <- SAR.accum.dist(active.comm)

```

```

## [1] 1088.693 3126.900

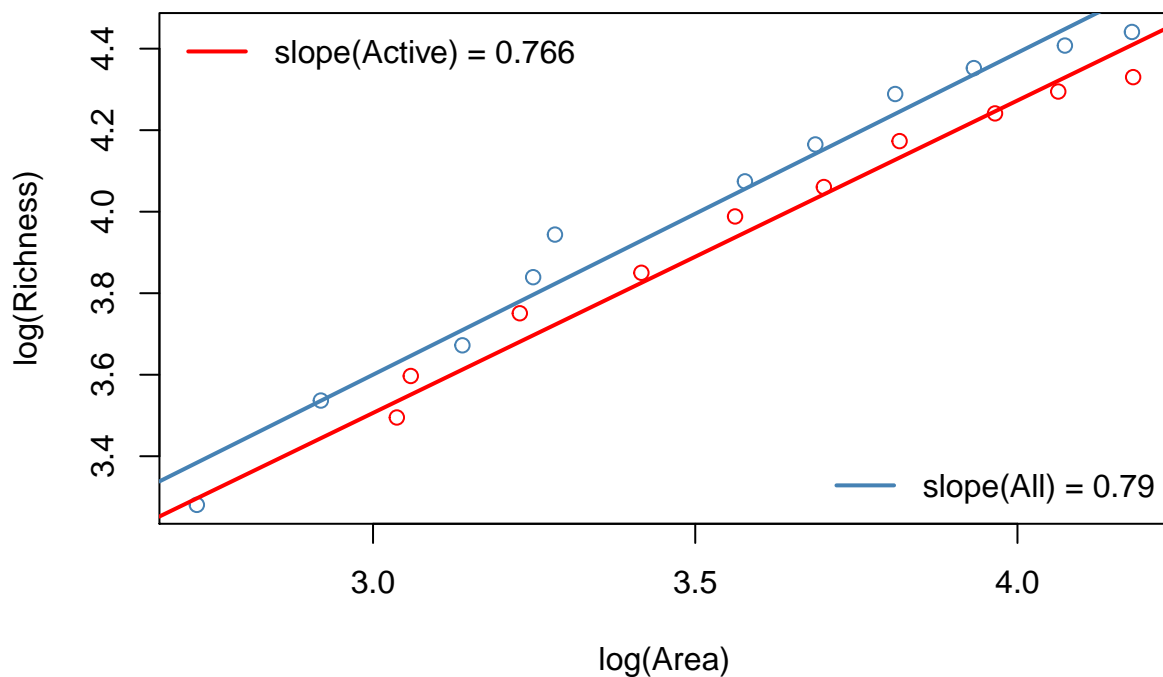
```

```
## [1] 556.06 1626.90
## [1] 1144.258 3953.000
## [1] 1689.378 5633.200
## [1] 2608.456 7081.800
## [1] 3645.592 9734.840
## [1] 5006.80 11494.28
## [1] 6562.429 14899.160
## [1] 9229.006 17423.080
## [1] 11569.20 19713.98
## [1] 15118.95 21383.74
```

```
sar <- as.data.frame(sar)
points(sar, xlab = "log(Area)", ylab = "log(Richness)",
       main = "Species-Area Relationship (Active)", col = "red")

OLS <- lm(sar$V2 ~ sar$V1)
abline(OLS, col = "red", lw = 2)
slope <- round(coefficients(OLS)[2], 3)
legend("topleft", legend = paste("slope(Active) =", slope),
       bty = "n", lw = 2, col = "red")
```

Taxa–Area Relationship aggregating area by distance



```
par(mfrow=c(1, 1))

sar <- SAR.rand accum(all.comm)
```

```
## [1] 333.2245 1793.3800
```

```
## [1] 593.5248 3631.1200
## [1] 1307.671 6333.000
## [1] 1676.126 8211.120
## [1] 2432.083 9957.380
## [1] 3642.401 12926.680
## [1] 4703.708 15441.780
## [1] 7333.975 19583.520
## [1] 9495.488 22549.740
## [1] 12481.72 25741.74
## [1] 15159.69 27839.00
```

```
sar <- as.data.frame(sar)
plot(sar, xlab = "log(Area)", ylab = "log(Richness)",
     main = "Taxa-Area Relationship
           aggregating area at random", col = "SteelBlue")

OLS <- lm(sar$V2 ~ sar$V1)
abline(OLS, col = "SteelBlue", lw = 2)
slope <- round(coefficients(OLS)[2], 3)
legend("bottomright", legend = paste("slope(All) =", slope),
      bty = "n", lw = 2, col = "SteelBlue")

sar <- SAR.rand.accum(active.comm)
```

```
## [1] 270.3573 1679.8600
## [1] 602.0572 3035.1800
## [1] 1273.337 5082.460
## [1] 1724.227 6789.700
## [1] 2426.755 8143.740
## [1] 3498.775 10385.600
## [1] 4676.941 12150.200
## [1] 7099.192 15122.880
## [1] 9625.251 17566.380
## [1] 12459.16 19834.56
## [1] 15159.69 21510.00
```

```
sar <- as.data.frame(sar)
points(sar, xlab = "log(Area)", ylab = "log(Richness)",
      main = "Species-Area Relationship (Active)", col = "red")

OLS <- lm(sar$V2 ~ sar$V1)
abline(OLS, col = "red", lw = 2)
slope <- round(coefficients(OLS)[2], 3)
legend("topleft", legend = paste("slope(Active) =", slope),
      bty = "n", lw = 2, col = "red")
```

Taxa–Area Relationship aggregating area at random

