**#1. R Code for Structural Equation Model with lavaan package**

**# This is the final model reported in main text**

# load libraries

library(scales)

library(lsmeans)

library(vegan)

library(PerformanceAnalytics)

library(lavaan)

library(semPaths)

library(semPlot)

# load data

biod = read.csv(file="biodiversa.database.full.csv", header= T, sep=",")

#Subset data just for exploring Agricultural Sites

d <- biod[which(biod$Treatment\_main == "Agriculture"),]

#Index Calculator

#This is using raw data (i.e. non-detrended)

### 1. Our construct variable for Management Index ####

d$fert\_tot\_N\_min\_amt = d$fertilizer\_min\_N\_amount + d$fertilizer\_organic\_N\_min\_amount

MI = c(

"fert\_tot\_N\_min\_amt",

"tillage\_no\_times",

"tillage\_depth\_max\_cm",

"fung\_no\_app",

"insect\_no\_app",

"herb\_no\_app"

)

MI.scale = d[,MI]

MI.scale = scale(MI.scale)

pairs(MI.scale)

plot(rda(MI.scale[complete.cases(MI.scale), MI], scale = TRUE))

pc = prcomp(MI.scale[complete.cases(MI.scale), MI], center = TRUE, scale. = F)

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$MI = NA

d[complete.cases(MI.scale),]$MI = pc$x %\*% (dr\*wt)

biplot(pc, cex=1.3, xlim = c(-0.3,0.4), ylim = c(-0.3,0.2))

# re-scale to make values ranging from 0 to 1

d$MI = d$MI + abs(min(d$MI, na.rm = T))

d$MI = d$MI/max(d$MI, na.rm=T)

# Check correlation of indicators are correctly reflecting the new index

cor(d[is.na(d$MI)==F,c("MI", MI)], method="spearman")

# plot of correlations of indicators to composite variable

LCL=UCL=COR=NULL

ind = MI # indicators

for (i in ind){

Test = cor.test(d$MI, d[,i])

LCL = c(LCL, Test$conf[1])

UCL = c(UCL, Test$conf[2])

COR = c(COR, Test$estimate)

}

pdf(paste(outputFolder, "MI\_corr\_barplot.pdf", sep=""), width = 3, height = 4)

par(mfrow=c(1,1), mar=c(6,4,2,2))

barplot(sort(COR, decreasing=T), border=NA, tck=0.02, mgp=c(1.5, 0.1,0), cex.axis=0.8, names=ind[order(COR, decreasing=T)], cex.names=0.8, las=3)

# dev.off()

write.table(d, file="updateddatafile.csv", sep = ",")

d$MI

### 2. Our construct variable for Soil Multifunctionality Index ####

#Composed of four different parameters: soil structure, soil fertility, soil activity, and potential N cycling

## a) Soil Structure ####

Str = c(

"MWD",

"bulk\_density"

)

Str.scale = d[,Str]

str.scale=scale(Str.scale)

pairs(Str.scale)

plot(rda(Str.scale[complete.cases(Str.scale), Str]))

# Calcualte Soil Strucutre index

Str.scale[ ,"bulk\_density"] = -1\*Str.scale[ ,"bulk\_density"] #I inversed this since lower bulk density is better indicator of soil structure

pc = prcomp(Str.scale[complete.cases(Str.scale), Str]); pc

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$Str = NA

d[complete.cases(Str.scale),]$Str = pc$x %\*% (dr\*wt)

biplot(pc, cex= 1.3, xlim = c(-0.2, 0.3))

# re-scale to make values ranging from 0 to 1

d$Str = d$Str+abs(min(d$Str, na.rm=T))

d$Str = d$Str/max(d$Str, na.rm=T)

# Check correlation of indicators are correctly reflecting the new index

cor(d[is.na(d$Str)==F,c("Str", Str)], method="spearman")

# plot of correlations of indicators to composite variable

LCL=UCL=COR=NULL

ind = Str # indicators

for (i in ind){

Test = cor.test(d$Str, d[,i])

LCL = c(LCL, Test$conf[1])

UCL = c(UCL, Test$conf[2])

COR = c(COR, Test$estimate)

}

# pdf(paste(outputFolder, "Soil\_structure\_corr\_barplot.pdf", sep=""), width = 3, height = 4)

par(mfrow=c(1,1), mar=c(6,4,2,2))

barplot(sort(COR, decreasing=T), border=NA, tck=0.02, mgp=c(1.5, 0.1,0), cex.axis=0.8, names=ind[order(COR, decreasing=T)], cex.names=0.8, las=3)

# dev.off()

## b) Soil Fertility ####

Nut = c(

"N\_tot\_mg",

"P\_tot\_mg",

"P\_Olsen",

"Calcium",

"Magnesium",

"Potasium",

"CEC",

"WHC",

"C\_org\_mg"

)

Nut.scale = d[, Nut]

Nut.scale=scale(Nut.scale)

pairs(Nut.scale)

plot(rda(Nut.scale[complete.cases(Nut.scale), Nut]))

# Calcualte Soil Fertility Index without scaled variable

pc = prcomp(Nut.scale[complete.cases(Nut.scale), Nut]); pc

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$Nut = NA

d[complete.cases(Nut.scale),]$Nut = pc$x %\*% (dr\*wt)

biplot(pc, xlim= c(-0.2,0.45), ylim=c(-0.3,0.3), cex=1.3)

# re-scale to make values ranging from 0 to 1

d$Nut = d$Nut +abs(min(d$Nut, na.rm=T))

d$Nut = d$Nut/max(d$Nut, na.rm=T)

# plot of correlations of indicators to composite variable

LCL=UCL=COR=NULL

ind = Nut # indicators

for (i in ind){

Test = cor.test(d$Nut, d[,i])

LCL = c(LCL, Test$conf[1])

UCL = c(UCL, Test$conf[2])

COR = c(COR, Test$estimate)

}

# Check correlation to indicators

pdf(paste(outputFolder, "Soil\_fertility\_corr\_barplot.pdf", sep=""), width = 3.3, height = 4)

par(mfrow=c(1,1), mar=c(6,4,2,2))

barplot(sort(COR, decreasing=T), border=NA, tck=0.02, mgp=c(1.5, 0.1,0), cex.axis=0.8, names=ind[order(COR, decreasing=T)], cex.names=0.8, las=3)

#dev.off()

## a) Soil Activity ####

Act = c(

"basal\_respiration",

"XYL",

"BG",

"AG",

"NAG",

"CB",

"LAP",

"PHOS"

)

Act.scale = d[, Act]

Act.scale = scale(Act.scale)

pairs(Act.scale)

plot(rda(Act.scale[complete.cases(Act.scale), Act]))

# Calcualte Soil Activity index without scaled variable

pc = prcomp(Act.scale[complete.cases(Act.scale), ]); pc

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$Act = NA

d[complete.cases(Act.scale),]$Act = pc$x %\*% (dr\*wt)

biplot(pc, xlim=c(-0.2,0.3), ylim=c(-0.3,0.25), cex=1.3)

# re-scale to make values ranging from 0 to 1

d$Act = d$Act +abs(min(d$Act, na.rm=T))

d$Act = d$Act/max(d$Act, na.rm=T)

# plot of correlations of indicators to composite variable

LCL=UCL=COR=NULL

ind = Act # indicators

for (i in ind){

Test = cor.test(d$Act, d[,i])

LCL = c(LCL, Test$conf[1])

UCL = c(UCL, Test$conf[2])

COR = c(COR, Test$estimate)

}

# Check correlation to indicators

pdf(paste(outputFolder, "Soil\_Activity\_corr\_barplot.pdf", sep=""), width = 3.3, height = 4)

par(mfrow=c(1,1), mar=c(6,4,2,2))

barplot(sort(COR, decreasing=T), border=NA, tck=0.02, mgp=c(1.5, 0.1,0), cex.axis=0.8, names=ind[order(COR, decreasing=T)], cex.names=0.8, las=3)

#dev.off()

## d) Soil N Cycling ####

Nit = c(

"AMP",

"NIP",

"DEP",

"MIN",

"NTR",

"PDA"

)

Nit.scale = d[, Nit]

Nit.scale = scale(Nit.scale)

pairs(Nit.scale)

plot(rda(Nit.scale[complete.cases(Nit.scale), Nit]))

biplot(rda(Nit.scale[complete.cases(Nit.scale), Nit]))

# Calcualte Soil Nitrogen Cycling index without scaled variable

pc = prcomp(Nit.scale[complete.cases(Nit.scale), Nit]); pc

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$Nit = NA

d[complete.cases(Nit.scale),]$Nit = pc$x %\*% (dr\*wt)

biplot(pc, cex=1.3)

# re-scale to make values ranging from 0 to 1

d$Nit = d$Nit +abs(min(d$Nit, na.rm=T))

d$Nit = d$Nit/max(d$Nit, na.rm=T)

# plot of correlations of indicators to composite variable

LCL=UCL=COR=NULL

ind = Nit # indicators

for (i in ind){

Test = cor.test(d$Nit, d[,i])

LCL = c(LCL, Test$conf[1])

UCL = c(UCL, Test$conf[2])

COR = c(COR, Test$estimate)

}

# Check correlation to indicators

pdf(paste(outputFolder, "Soil\_Nitrogen\_Cycling\_corr\_barplot.pdf", sep=""), width = 3.3, height = 4)

par(mfrow=c(1,1), mar=c(6,4,2,2))

barplot(sort(COR, decreasing=T), border=NA, tck=0.02, mgp=c(1.5, 0.1,0), cex.axis=0.8, names=ind[order(COR, decreasing=T)], cex.names=0.8, las=3)

#dev.off()

### Soil Multifunctionality Index ####

d$SMFgg = rowMeans(cbind(d$Str, d$Nut, d$Act, d$Nit))

# re-scale to make values ranging from 0 to 1

d$SMFgg = d$SMFgg + abs(min(d$SMFgg, na.rm = T))

d$SMFgg = d$SMFgg/max(d$SMFgg, na.rm=T)

#Composite variables for non-linearity (taking the first PCA axis according to Manuel's paper, here we use the weighted average)

#non-linear pH

pc = prcomp(scale(cbind(d$pH, d$pH^2)))

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$NL.pH = NA

d$NL.pH = pc$x %\*% (dr\*wt)

#non-linear Clay

pc = prcomp(scale(cbind(d$Clay, d$Clay^2)))

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$NL.Clay = NA

d$NL.Clay = pc$x %\*% (dr\*wt)

#non-linear MAT

pc = prcomp(scale(cbind(d$avg\_temp\_longterm, d$avg\_temp\_longterm^2)))

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$NL.MAT = NA

d$NL.MAT = pc$x %\*% (dr\*wt)

#non-linear MAP

pc = prcomp(scale(cbind(d$avg\_precip\_longterm, d$avg\_precip\_longterm^2)))

dr = ifelse(colMeans(pc$rotation)<0,-1,1)

wt = pc$sdev/sum(pc$sdev) # weighting of axis

d$NL.MAP = NA

d$NL.MAP = pc$x %\*% (dr\*wt)

#Model 1 --> baseline

model.Lavaan <-"

std\_yield\_network ~

NL.MAT

+ NL.MAP

+ Shannon\_bac

+ Shannon\_fun

+ Shannon\_arc

+ cerco\_shannon

+ MI

+ SMFgg

+ X10a\_months\_cropped\_month

+ Shannon\_10

+ NL.pH

+ NL.Clay

SMFgg ~

NL.MAT

+ NL.MAP

+ Shannon\_bac

+ Shannon\_fun

+ Shannon\_arc

+ cerco\_shannon

+ MI

# + SMFgg

+ X10a\_months\_cropped\_month

+ Shannon\_10

+ NL.pH

+ NL.Clay

Shannon\_bac ~

NL.MAT

+ NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

+ MI

# + SMFgg

+ X10a\_months\_cropped\_month

+ Shannon\_10

+ NL.pH

+ NL.Clay

Shannon\_fun ~

NL.MAT

+ NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

+ MI

# + SMFgg

+ X10a\_months\_cropped\_month

+ Shannon\_10

+ NL.pH

+ NL.Clay

Shannon\_arc ~

NL.MAT

+ NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

+ MI

# + SMFgg

+ X10a\_months\_cropped\_month

+ Shannon\_10

+ NL.pH

+ NL.Clay

cerco\_shannon ~

NL.MAT

+ NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

+ MI

# + SMFgg

+ X10a\_months\_cropped\_month

+ Shannon\_10

+ NL.pH

+ NL.Clay

Shannon\_bac ~~ Shannon\_arc + Shannon\_fun + cerco\_shannon

Shannon\_fun ~~ Shannon\_arc + cerco\_shannon

Shannon\_arc ~~ cerco\_shannon

"

fit <- lavaan:::cfa(model.Lavaan, data = d, std.lv = TRUE)

summary(fit, rsq=T, fit.measures =T)

standardizedSolution(fit, pvalue=TRUE)

#narrowed down non-significant terms until best model is reached

#Here is our final model

model.Lavaan <-"

std\_yield\_network ~

NL.MAT

# + NL.MAP

+ Shannon\_bac

+ Shannon\_fun

+ Shannon\_arc

# + cerco\_shannon

+ MI

+ SMFgg

+ X10a\_months\_cropped\_month

# + Shannon\_10

# + NL.pH

#+ NL.Clay

SMFgg ~

NL.MAT

+ NL.MAP

+ Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

# + MI

# + SMFgg

+ X10a\_months\_cropped\_month

+ Shannon\_10

# + NL.pH

+ NL.Clay

Shannon\_bac ~

NL.MAT

# + NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

# + MI

# + SMFgg

+ X10a\_months\_cropped\_month

# + Shannon\_10

# + NL.pH

+ NL.Clay

Shannon\_fun ~

# NL.MAT

+ NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

+ MI

# + SMFgg

# + X10a\_months\_cropped\_month

+ Shannon\_10

# + NL.pH

+ NL.Clay

Shannon\_arc ~

NL.MAT

+ NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

# + MI

# + SMFgg

+ X10a\_months\_cropped\_month

# + Shannon\_10

+ NL.pH

+ NL.Clay

cerco\_shannon ~

NL.MAT

+ NL.MAP

# + Shannon\_bac

# + Shannon\_fun

# + Shannon\_arc

# + cerco\_shannon

+ MI

# + SMFgg

+ X10a\_months\_cropped\_month

# + Shannon\_10

+ NL.pH

+ NL.Clay

Shannon\_bac ~~ Shannon\_arc + cerco\_shannon

Shannon\_fun ~~ cerco\_shannon

Shannon\_arc ~~ cerco\_shannon

"

fit <- lavaan:::cfa(model.Lavaan, data = d, std.lv = TRUE)

summary(fit, rsq=T, fit.measures =T)

standardizedSolution(fit, pvalue=TRUE)

**# 2. Structural Equation Model Code for testing effects of different crop types (i.e. cash, cover, and leys)**

#The indices were created the same as above, however, instead of including Crop diversity as a factor (i.e. Shannon\_10), we included the proportion of each crop type individually (i.e. cash, cover, ley) and further divided between legume and non-legume (see Supplementary Fig. 7). We also tried with the proportion of non-cash crops planted (i.e. cover and leys combined).

#For this, we start here with the baseline of all groups, and remove non-significant factors after each model iteration until the best model is achieved.

#Below is just the baseline for all groups individually, since in the end these models were not as strong at the model including overall crop cover.

#Model 1 --> all groups baseline

model.Lavaan <-"

std\_yield\_network ~

                    NL.MAT

                  + NL.MAP

                  + Shannon\_bac

                  + Shannon\_fun

                  + Shannon\_arc

                  + cerco\_shannon

                  + MI

                  + SMFgg

                + Cash\_crop\_prop

                + Cover\_crop\_prop

                + Ley\_prop

              #   + Non\_cash\_cropped

                + Shannon\_10

              + Richness\_10

                  + NL.pH

                  + NL.Clay

SMFgg ~

                   NL.MAT

                  + NL.MAP

                  + Shannon\_bac

                  + Shannon\_fun

                  + Shannon\_arc

                  + cerco\_shannon

                  + MI

                 + Cash\_crop\_prop

                + Cover\_crop\_prop

                + Ley\_prop

                   # + Non\_cash\_cropped

                  + Shannon\_10

                + Richness\_10

                   + NL.pH

                  + NL.Clay

Shannon\_bac ~

                  NL.MAT

                  + NL.MAP

                  + MI

                 + Cash\_crop\_prop

             #   + Cover\_crop\_prop

                + Ley\_prop

               # + Non\_cash\_cropped

                  + Shannon\_10

                + Richness\_10

                  + NL.pH

                  + NL.Clay

Shannon\_fun ~

                    NL.MAT

                  + NL.MAP

                  + MI

                + Cash\_crop\_prop

                + Cover\_crop\_prop

                + Ley\_prop

               # + Non\_cash\_cropped

                  + Shannon\_10

                 + Richness\_10

                  + NL.pH

                  + NL.Clay

Shannon\_arc ~

                   NL.MAT

                  + NL.MAP

                  + MI

                  + Cash\_crop\_prop

                  + Cover\_crop\_prop

                + Ley\_prop

               # + Non\_cash\_cropped

                + Shannon\_10

                + Richness\_10

                  + NL.pH

                  + NL.Clay

cerco\_shannon ~

                  NL.MAT

                  + NL.MAP

                  + MI

              + Cash\_crop\_prop

              + Cover\_crop\_prop

                + Ley\_prop

                #  + Non\_cash\_cropped

                  + Shannon\_10

                + Richness\_10

                  + NL.pH

                  + NL.Clay

Shannon\_bac ~~ Shannon\_arc + Shannon\_fun + cerco\_shannon

Shannon\_fun ~~  Shannon\_arc + cerco\_shannon

Shannon\_arc ~~ cerco\_shannon

"

fit <- lavaan:::cfa(model.Lavaan, data = d, std.lv = TRUE)

summary(fit, rsq=T, fit.measures =T)

standardizedSolution(fit, pvalue=TRUE)