Productivity, biodiversity, and pathogens influence global hunter-gatherer population density: Data analyses

Miikka Tallavaara, Jussi T. Eronen, and Miska Luoto

2017

Contents

1 Introduction **1**

2 Download R packages **2**

3 Data **2**
3.1 Predictor variables . . . 2
3.1.1 Net primary productivity (NPP) . . . 2
3.1.2 Biodiversity . . . 4
3.1.3 Pathogen stress . . . 7
3.2 Response variable . . . 11
3.2.1 Hunter-gatherer population density . . . 11
3.3 Combined data of population density and environmental predictors. . . . 13

4 Statistical analyses **13**
4.1 Breakpoint analysis . . . 13
4.2 Structural equation modelling (SEM) . . . 17
4.2.1 Function for the analysis of spatial autocorraltion . . . 17
4.2.2 Fitting of structural equation models . . . 18
4.2.3 Partial residuals and predictive power of the SE model . . . 24
4.2.4 Direct, indirect, and total effects of predictors on population density . . . 25
4.2.5 Linear model with interactions . . . 27
4.3 Global projection of hunter-gatherer population density and its constraints . . . 29
4.3.1 Performance of the prediction model . . . 29
4.3.2 Global dataset of predicted hunter-gatherer population density . . . 33
4.3.3 Analysis of limiting factors . . . 38
4.3.4 Biome-wise analyses . . . 44

References **50**

1 Introduction

This document makes it possible to reproduce data manipulations and analyses in Tallavaara et al. 2017, by running the following code (grey background) in *R* (https://www.r-project.org/). It assumes that the ethnographic hunter-gatherer dataset named `Tallavaara_Dataset_4.xls` is in the working directory. This dataset can be downloaded from Zenodo repository. Other data are loaded from the web as the code runs. However, there is also code for downloading these data from the working directory.

NOTE that if there are problems in downloading the data from the web, check your browser settings (or try again later)!

2 Download R packages

Install packages, if needed.

```
packages <- c("ape", "car", "fossil", "gstat", "piecewiseSEM",
              "rgdal", "raster", "segmented", "sp", "XML", "plyr", "pbapply", "readxl",
              "jtools", "visreg", "e1071", "plot3D")
install.packages(packages, dependencies = TRUE)
```

Next, load packages into workspace.

```
library(gstat)
library(rgdal)
library(raster)
library(ape)
library(car)
library(fossil)
library(piecewiseSEM)
library(segmented)
library(sp)
library(XML)
library(plyr)
library(pbapply)
library(readxl)
library(jtools)
library(visreg)
library(e1071)
library(plot3D)
```

3 Data

We use existing data sources from where we extract information on three environmental predictor variables and an ethnographic response variable.

3.1 Predictor variables

Environmental predictor variables, whose effect on hunter-gahtherer population density we analyse are:

- Climate-based net primary productivity
- Biodiversity (scaled mammal, bird and vascular plant richness)
- Pathogen stress (scaled prevalence measure of 10 pathogens)

3.1.1 Net primary productivity (NPP)

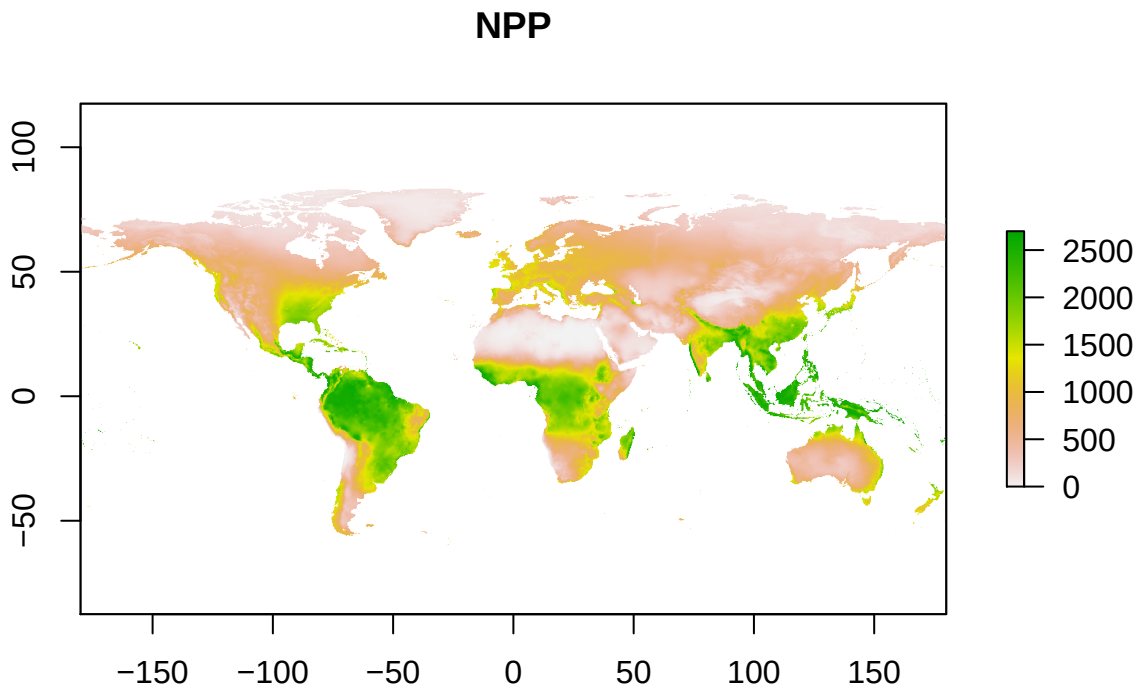
We downloade annual mean temperature and annual precipitation data (BIOCLIM variables bio1 and bio12) from WorldClim database (http://www.worldclim.org/current). Based on these values, we calculated NPP using the Miami model of Lieth (1973):

$$NPP(Temp, Precip) = min\{\frac{3000}{1 + e^{1.315 - 0.119 \times Temp}},\ 3000 \times (1 - e^{-0.000664 \times Precip})\}$$

```
URL <- "http://biogeo.ucdavis.edu/data/climate/worldclim/1_4/grid/cur/bio_10m_bil.zip"

wd <- getwd()
td <- tempdir()
setwd(td)
temp <- tempfile(fileext = ".zip")
download.file(URL, temp)
unzip(temp)
tempRast <- raster("bio1.bil")/10
precipRast <- raster("bio12.bil")/1
unlink(dir(td))
setwd(wd)
```

```
# Calculate NPP using the Miami model.
npp.mia.t <- 3000/(1+exp(1.315-0.119*tempRast))
npp.mia.p <- 3000*(1-exp(-0.000664*precipRast))
nppRast<-min(stack(npp.mia.t,npp.mia.p))
```

```
plot(nppRast, main="NPP")
```




```
# Alternately, if the Dataset S2 is in the working directory, it can be loaded using
# following code (remove hash mark):
# nppRast <- raster("Tallavaara_Dataset_1.tif")
```

3.1.2 Biodiversity

Biodiversity variable is based on global mammal, bird, and vascular plant richness data. We scaled these richness values to vary between 0 an 1 and finally averaged them to form a single biodiversity variable.

3.1.2.1 Mammal and bird richness

We downloaded mammal and bird richness data from Biodiversitymapping.org (http://biodiversitymapping.org/wordpress/index.php/home/).

```
URL <- paste("http://biodiversitymapping.org/","wordpress/wp-content/uploads/2016/11/",
 "biodiversitymapping_TIFFs_20Dec2016.zip", sep="") # Use paste just to avoid problems
                                                   # to RMarkdown output caused by
                                                   # long strings

wd <- getwd()
td <- tempdir()
setwd(td)
temp <- tempfile(fileext = ".zip")
download.file(URL, temp)
unzip(temp)
birDRast <- raster("biodiversitymapping_TIFFs/Birds/richness_10km_all_raster.tif")
mamDRast <- raster("biodiversitymapping_TIFFs/Mammals/richness_10km_all_spp_raster.tif")
unlink(dir(td))
setwd(wd)

# Use the projection of the NPP raster as a template projection for mammal and bird
# diversity rasters, combine them into 'animal' diversity, and scale animal diversity
# to vary between 0 to 1.

birDRast<-projectRaster(birDRast, nppRast)
mamDRast<-projectRaster(mamDRast, nppRast)
animDRast <- birDRast + mamDRast
animDRast.scal <- animDRast/cellStats(animDRast,'max')
```

3.1.2.2 Plant richness

Our plant richness data are based on Kreft and Jetz (2007) and are part of the Ellis et al. (2012) data (native plant richness 'N'). The data can be downloaded from ecotope.org (http://ecotope.org/anthromes/biodiversity/plants/data/). We interpolated these global data over the same global grid as NPP and animal diversity (see above) using inverse distance weighting (IDW) interpolation. Below, we will also eveluate the interpolation by comparing interpolated richness to original (model-based) richness data.

```
URL <- "http://ecotope.org/files/ellis_2012/ellis_2012_shapefile.zip"
wd <- getwd()
td <- tempdir()
setwd(td)
temp <- tempfile(fileext = ".zip")
download.file(URL, temp)
unzip(temp)
dsn <- dir(tempdir(), "*.shp$")
layer <- sub(".shp$", "", dsn)
plants <- readOGR(dsn=dsn, layer=layer)
unlink(dir(td))
```

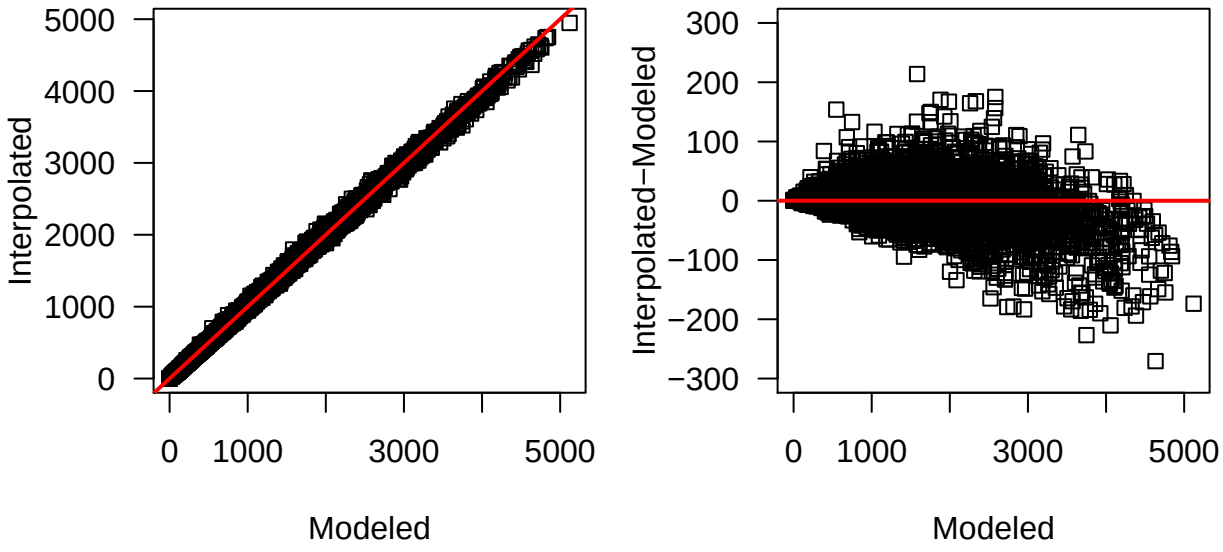
```r
setwd(wd)

# Transform Spatial Polygons Data into a normal data frame, subset, and transfor it into
# a Spatial Points Data Frame and change projection to the global Mercator projection.
plants <- as.data.frame(plants)
plants <- data.frame(plants$X, plants$Y, plants$N)
names(plants) <- c("x","y","N")

plants2 <- plants # This data will be used later, when comparing interpolaton
                  # to original data

coordinates(plants) <- ~x+y
proj4string(plants) = CRS("+proj=longlat +datum=WGS84")
plantsM <- spTransform(plants, CRS("+init=epsg:3395"))

# Create a global grid with the same extent and resolution as NPP raster and change the
# projection of global grid to the global Mercator projection. This grid is used when
# interpolating the plant richness data to cover the same area as other environmetal
# variables.

globalGrid <- as.data.frame(expand.grid(x=seq(-180,180, by=0.1666667),
                                        y=seq(-80,84, by=0.1666667)))
coordinates(globalGrid) <- ~x+y
proj4string(globalGrid) = CRS("+proj=longlat +datum=WGS84")

globalGridM <- spTransform(globalGrid, CRS("+init=epsg:3395"))
```

```r
# IDW-interpolation to the plant richness data using krige-function (without any model)
# of the gstat package
plantsIDW <- krige(N ~ 1, plantsM, globalGridM, NULL, nmax=10)

# Change projection of the interpolated data to lon-lat, make a raster out of the data
# with NPP raster as a template, and scale the plant diversity.

plantsIDWLL <- spTransform(plantsIDW, CRS("+proj=longlat +datum=WGS84"))
plantsDataIDW <- as.data.frame(plantsIDWLL)
plantsDataIDW <- data.frame(plantsDataIDW$x, plantsDataIDW$y, plantsDataIDW$var1.pred)
plantDRast <- rasterFromXYZ(plantsDataIDW)
crs(plantDRast) <- "+proj=longlat +datum=WGS84"
plantDRast<-projectRaster(plantDRast, nppRast)
plantDRast<-mask(plantDRast, nppRast)
plantDRast.scal <- plantDRast/cellStats(plantDRast,'max')
```

Comparison of the interpolated richness to the original model-based richness

```r
# Extract interpolated values for original data points
pointsRast <- extract(plantDRast, cbind(plants2$x,plants2$y))

plants2$rastpoints <- pointsRast # add to the data

# Diff. between interpolated and original richness
plants2$diffs <- plants2$rastpoints-plants2$N

# Plot
```

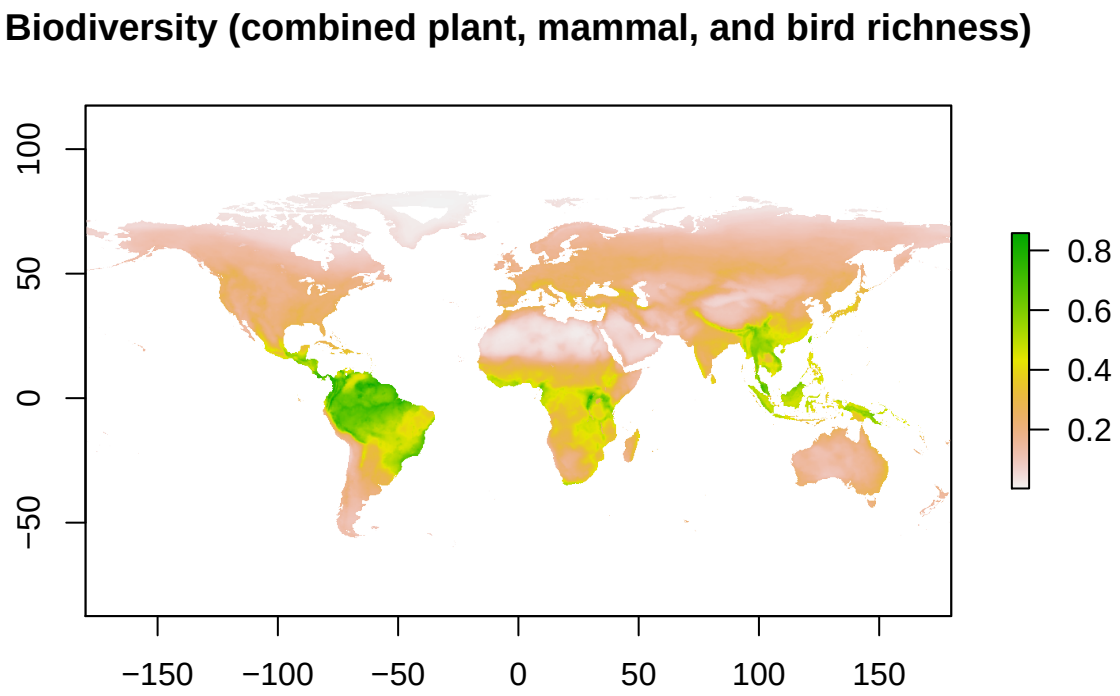
```
par(mfrow=c(1,2), mar=c(4,4,1,1))
plot(plants2$N,plants2$rastpoints, las=1, pch=0, ylab="Interpolated",
     xlab="Modeled")
abline(0,1, col="red", lwd=2)

plot(plants2$N,plants2$diffs, yaxt="n", ylim=c(-300,300), pch=0, xlab="Modeled",
     ylab="Interpolated-Modeled")
axis(side=2, at=c(-300,-200,-100,0,100,200,300), las=2)
abline(h=0, col="red", lwd=2)
```



3.1.2.3 Combined diversity

To form the final biodiversity variable, we averaged scaled animal (mammal and bird richness) and plant richness values.

```
bioDRast <- (animDRast.scal + plantDRast.scal)/2
plot(bioDRast, main="Biodiversity (combined plant, mammal, and bird richness)")
```




```
# Alternately, if the Dataset S3 is in the working directory, it can be loaded using
# following code (remove hash mark):
# bioDRast <- raster("Tallavaara_Dataset_2.tif")
```

3.1.3 Pathogen stress

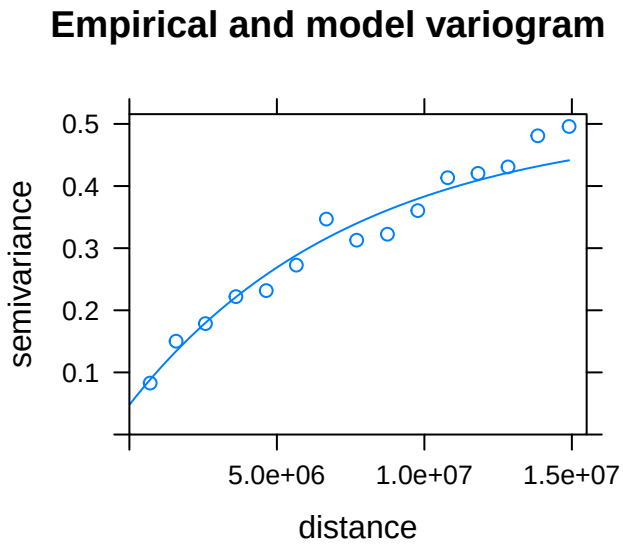
We extracted the pathogen data from Standard Cross-Cultural Sample (`SCCS`). The pathogen data are based on Cashdan (2014) and Low (1994). We interpolated the point data over the same global grid as other environmental variables using kriging interpolation with exponential variogram model in 'gstat' package. We downloaded the pathogen data from the E. Anthon Eff's webpage (http://capone.mtsu.edu/eaeff/DEf_SCCS.html). See also WorldCultures.org (http://www.worldcultures.org/) for the official site of the `SCCS` data. Following code loads the `SCCS` data and extracts variables needed for analysis. Variable `v2178` is the z-score variable of 10 pathogen prevalence values, which we use as the pathogen stress variable in our analyses. We added five pseudo data points in the western polar region and northern central Asia. These areas lack data in `SCCS` data. Western polar region z-score value of -0.96 is based on the mean of arctic and subarctic values in North America. Central Asian value of -0.77 is the mean of four northern Eurasian values (Saami (Lapps), Samoyed, Yukaghir, and Chukchee).Below, we will also evaluate interpolation of pathogen data by comparing interpolated and original pathogen stress values.

```
# Download the data.
load(url("http://capone.mtsu.edu/eaeff/downloads/DEf01f.Rdata"))
pathogens1 <- SCCS[,c("ord", "socname", "long", "lati")]
pathogens2 <- SCCS[,c("v2178")]
pathogens <- as.data.frame(cbind(pathogens1,pathogens2))
names(pathogens) <- c("ord", "socname", "lon", "lat", "pathoscore10")
```

```r
# Add pseudo data points
patho1 <- c(187, "pseudo1", -43.386172, 60.147033, -0.96)
patho2 <- c(188, "pseudo2", -23.163979, 70.899655, -0.96)
patho3 <- c(189, "pseudo3", -33.707298, 82.355691, -0.96)
patho4 <- c(192, "pseudo6", -86.851987, 81.298900, -0.96)
patho5 <- c(194, "pseudo8", 101.504786, 65.214012, -0.77)


pathogens <- as.data.frame(rbind(pathogens, patho1, patho2, patho3, patho4, patho5))
pathogens$lon <- as.numeric(pathogens$lon)
pathogens$lat <- as.numeric(pathogens$lat)
pathogens$pathoscore10 <- as.numeric(pathogens$pathoscore10)

pathogens2 <- pathogens # This data will be used later to evaluate the interpolation.
```

```r
# Interpolate these data over global grid (see above) using kriging interpolation.
# Switch projection to global Mercator
coordinates(pathogens) <- ~lon+lat
proj4string(pathogens) = CRS("+proj=longlat +datum=WGS84")
pathogensM <- spTransform(pathogens, CRS("+init=epsg:3395"))

# Calculate empirical semi variogram

pathoSemivgm<-variogram(pathoscore10~1, locations=pathogensM, data=pathogensM)
pathoSemivgm

# Based on its values, define range, nugget and partial sill

range <- 14905075/3 # 1/3*largest distance
nugget <- (0.08281362 + 0.15023716 + 0.17855292)/3 # average of the first three
                                                   # semivariogram values
psill  <- (0.41257526 + 0.41962169 + 0.43027830  # average of the last five values
           + 0.48063340 + 0.49545721)/5


# Model variogram with exponential model

pathoModVgm<-vgm(psill=psill, model="Exp", nugget=nugget, range=range)

pathoFitVgm <- fit.variogram(pathoSemivgm, pathoModVgm)
```

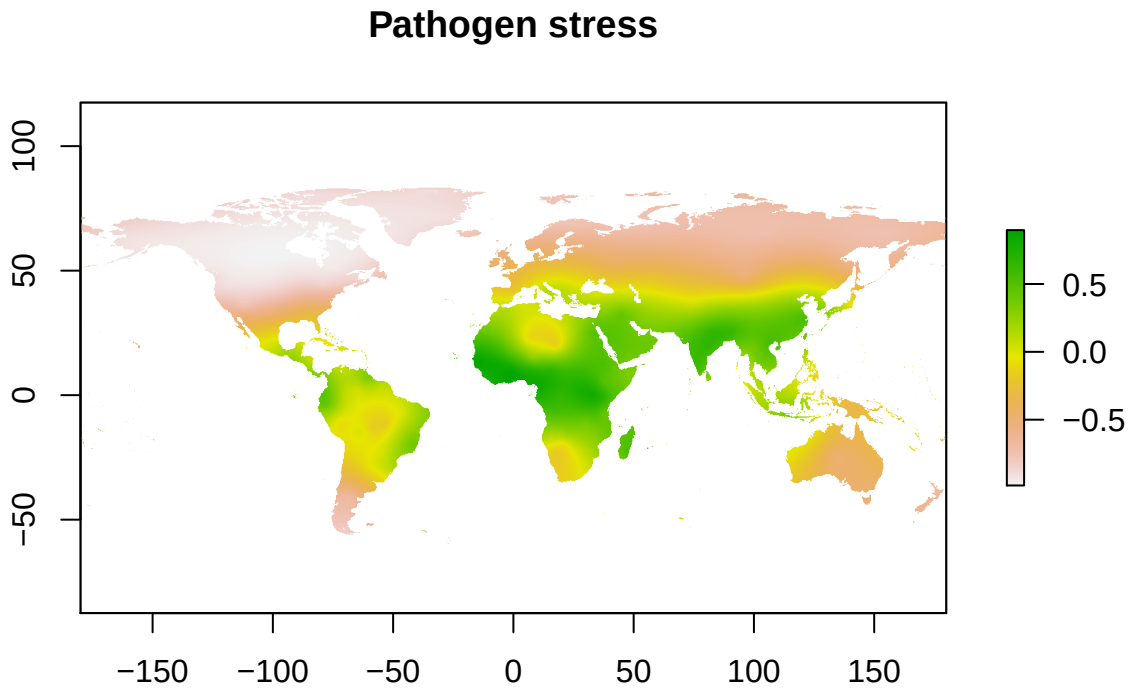
```r
plot(pathoSemivgm, pathoFitVgm, main="Empirical and model variogram", cex.main=0.8)
```




```
# Finally, kriging interpolation itself.

pathoKrig<-krige(formula=pathoscore10 ~ 1, locations=pathogensM, newdata=globalGridM,
                 model=pathoModVgm)
```

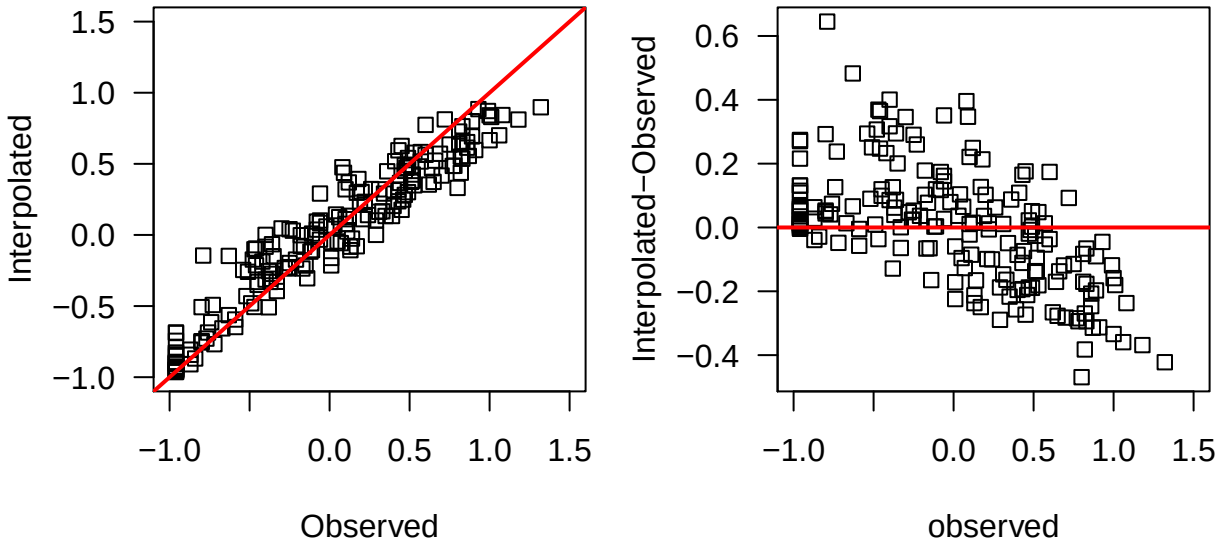
```
## [using ordinary kriging]
```

```
# Switch the projection of the interpolated data back to lon-lat.

pathoKrigedLL <- spTransform(pathoKrig, CRS("+proj=longlat +datum=WGS84"))

# Transform spatial data to 'normal' data frame.

pathoData <- as.data.frame(pathoKrigedLL)
pathoData <- data.frame(pathoKrigedLL$x, pathoKrigedLL$y, pathoKrigedLL$var1.pred)

# Create a raster.

pathoRast <- rasterFromXYZ(pathoData)
crs(pathoRast) <- "+proj=longlat +datum=WGS84"

# Use NPP raster as a template.

pathoRast<-projectRaster(pathoRast, nppRast)
pathoRast<-mask(pathoRast, nppRast)

# Plot results.

plot(pathoRast, main="Pathogen stress")
```



Comparison of the interpolated richness to the original model-based richness:

```
pointsRastPatho <- extract(pathoRast,cbind(pathogens2$lon,pathogens2$lat))
pathogens2$rastpoints <- pointsRastPatho
pathogens2$diffs <- pathogens2$rastpoints-pathogens2$pathoscore10

par(mfrow=c(1,2), mar=c(4,4,1,1))
plot(pathogens2$pathoscore10, pathogens2$rastpoints,las=1, pch=0,
     xlim=c(-1,1.5), ylim=c(-1,1.5), xlab="Observed", ylab="Interpolated")
abline(0,1, col="red", lwd=2)
plot(pathogens2$pathoscore10, pathogens2$diffs, yaxt="n", xlim=c(-1,1.5), pch=0,
     xlab="observed", ylab="Interpolated-Observed")
axis(side=2, at=c(-0.6,-0.4,-0.2,0,0.2, 0.4, 0.6), las=2)
abline(h=0, col="red", lwd=2)
```




```
# Alternately, if the Dataset S4 is in the working directory, it can be loaded using
# following code (remove hash mark):
# pathoRast <- raster("Tallavaara_Dataset_3.tif")
```

3.2 Response variable

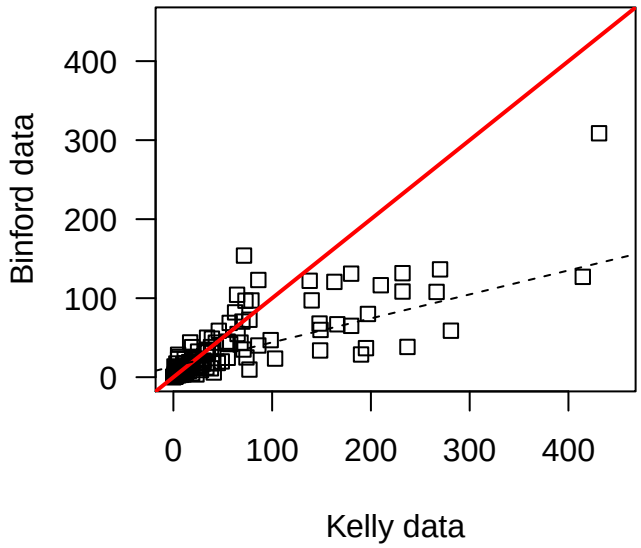
3.2.1 Hunter-gatherer population density

Our hunter-gatherer population density data is a combination of the datasets of Binford (2001) and Kelly (2013). If population is present in both sets, we used the mean population density of a given group. In cases where hunter-gatherer population is present only one of the datasets, we used the value given by that dataset. This (partly) combined variable is `densityC`. We used natural logarithm of population density (`ldensityC`) in our analyses.

```
# Load hunter-gatherer dataset (assumes that it is in the working directory).
# Use readxl package to read xls-file

hgdata <- read_excel("Tallavaara_Dataset_4.xls")
hgdata$densityB <- as.numeric(hgdata$densityB)
hgdata$densityK <- as.numeric(hgdata$densityK)
hgdata$densityC <- as.numeric(hgdata$densityC)
```

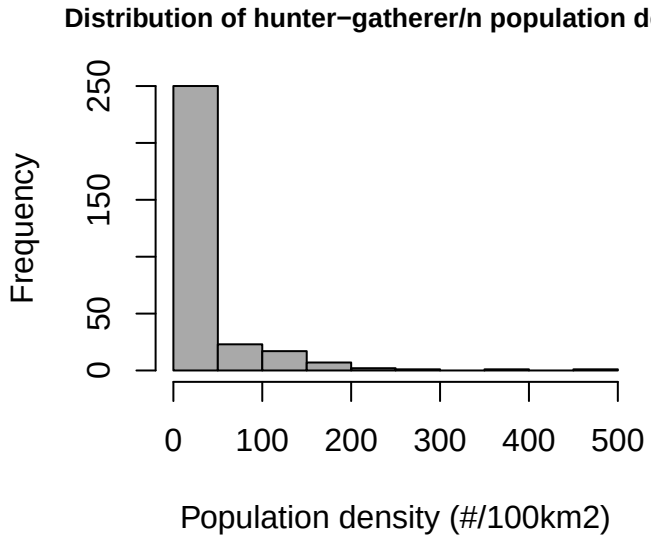
The relationship between Binford's and Kelly's data:

```
par(mar=c(4,4,1,1))
plot(hgdata$densityK, hgdata$densityB, ylim=c(0,450), xlim=c(0,450), las=1, pch=0,
     xlab="Kelly data", ylab="Binford data")
abline(0,1, col="red", lwd=2)
abline(lm(densityB~densityK, data=hgdata), lty=2)
```




```
# Add ln(population density) to the data.

hgdata$ldensityC <- log(hgdata$densityC)

# Remove suspicious "non-normal" hunter-gatherer cases.

hgdata <- hgdata[hgdata$subpop == "n",]
```

```
# Plot the distribution of hunter-gatherer population density.
par(mar=c(4,4,3,1))
hist(hgdata$densityC, main="Distribution of hunter-gatherer/n population density",
     xlab="Population density (#/100km2)", col="darkgrey", cex.main=0.8)
```




```
# Remove clear outliers

hgdata <- hgdata[hgdata$densityC < 300,]
```

```r
# Extract coordinates from the data

coords <- hgdata[,c("longitude","latitude")]
```

3.3 Combined data of population density and environmental predictors.

Here, we first extract values of the three environmental variables for locations of hunter-gatherer groups in the ethnographic data.Then, we combine population density with environmental variables to form the primary datset used in the analyses.

```r
# Make a raster stack of environmental variables

predictors <- stack(nppRast, bioDRast, pathoRast)

# Extract environmental information for hunter-gatherer populations

predictordata <- as.data.frame(extract(predictors,coords,layer=1,nl=3))
names(predictordata) <- c("npp","biodiv","pathos")

# Combine hunter-gatherer and environmental data

data <- cbind(hgdata,predictordata)
```

4 Statistical analyses

This section covers all the statistical analyses of the data constructed above. In addition, this section contains global projections of hunter-gatherer population density and the analyses of these global data.

4.1 Breakpoint analysis

Because visual inspection suggested changes in linear trends between NPP and population density and pathogen stress, we fitted piecewise regression models using segmented package. It estimates potential breakpoints and tests if the effects are indeed different before and after the estimated breakpoints. We divided our data into low and high productivity environments based on the threshold in NPP identified by the breakpoint analysis.

```r
# First, between npp and population density
fit1     <- lm(ldensityC ~ npp, data=data)
set.seed(1234)
fit1.s <- segmented(fit1, seg.Z=~npp,
                    control=seg.control(n.boot=1000,stop.if.error=T,it.max=1000))
summary(fit1.s)
```

```
##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
## segmented.lm(obj = fit1, seg.Z = ~npp, control = seg.control(n.boot = 1000,
##     stop.if.error = T, it.max = 1000))
##
## Estimated Break-Point(s):
```

```
##      Est.   St.Err
## 1371.664  103.425
##
## Meaningful coefficients of the linear terms:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1352714  0.1901257  -0.711    0.477
## npp          0.0028623  0.0002669  10.722   <2e-16 ***
## U1.npp      -0.0030745  0.0005137  -5.985       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.277 on 296 degrees of freedom
## Multiple R-Squared: 0.4504,  Adjusted R-squared: 0.4449
##
## Convergence attained in 8 iterations with relative change 0
```

```
davies.test(fit1,~npp,k=10)
```

```
##
##  Davies' test for a change in the slope
##
## data:  formula = ldensityC ~ npp ,   method = lm
## model = gaussian , link = identity
## segmented variable = npp
## 'best' at = 1467.5, n.points = 10, p-value = 1.742e-07
## alternative hypothesis: two.sided
```

```
# Next, between npp and pathogenic stress

fit2     <- lm(pathos ~ npp, data=data)
set.seed(1234)
fit2.s <- segmented(fit2, seg.Z=~npp,
                    control=seg.control(n.boot=1000,stop.if.error=T,it.max=1000))
summary(fit2.s)
```

```
##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
## segmented.lm(obj = fit2, seg.Z = ~npp, control = seg.control(n.boot = 1000,
##     stop.if.error = T, it.max = 1000))
##
## Estimated Break-Point(s):
##      Est.   St.Err
## 1349.258  118.018
##
## Meaningful coefficients of the linear terms:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.080e-01  3.750e-02 -21.545   <2e-16 ***
## npp          7.120e-05  5.326e-05   1.337    0.182
## U1.npp       5.244e-04  9.936e-05   5.278       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2491 on 296 degrees of freedom
```

```
## Multiple R-Squared: 0.3001,  Adjusted R-squared: 0.293
##
## Convergence attained in 3 iterations with relative change 1.93523e-16
```

```
davies.test(fit2,~npp,k=5)
```

```
##
##  Davies' test for a change in the slope
##
## data:  formula = pathos ~ npp ,   method = lm
## model = gaussian , link = identity
## segmented variable = npp
## 'best' at = 1334.1, n.points = 5, p-value = 4.979e-06
## alternative hypothesis: two.sided
```

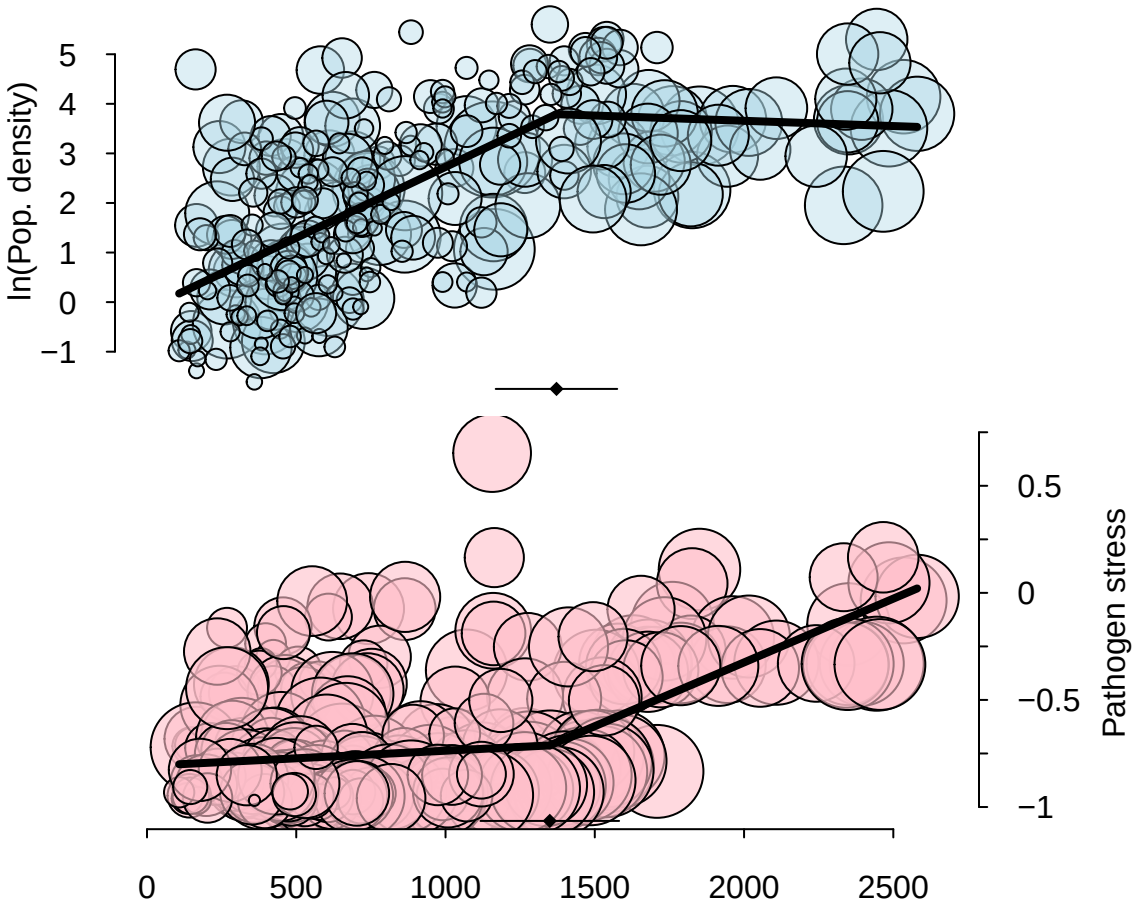
```
# Divide the data into low and high productivity regimes using the mean of
# the above-mentioned npp thresholds

nppThresh <- (fit1.s$psi[2]  + fit2.s$psi[2])/2
data$nrg <- ifelse(data$npp > nppThresh,
c("hi"), c("lo"))
data$nrg <- as.factor(data$nrg)
```

```
# Plot the figure

par(mar=c(3,5,0,5))
par(fig=c(0,1,0.45,1))
symbols(data$npp,data$ldensityC,
        circle=sqrt((data$pathos+sqrt(min(data$pathos)^2)+0.03)/pi), inches=0.25, xlab="",
        fg="black", bg="#ADD8E666", axes=F,ylab="",xlim=c(0,max(data$npp)+100),
        ylim=c(min(data$ldensityC),max(data$ldensityC+0.5)))
axis(side=2, at=c(-1,0,1,2,3,4,5),las=2, tck=-0.02)
mtext("ln(Pop. density)", side=2, line=2)
plot(fit1.s,add=T,rug=F,lwd=4,col="black")
lines(fit1.s)
#radius <- sqrt((data$pathos+sqrt(min(data$pathos)^2))/pi)
#ylim <- c(min(data$ldensityC),max(data$ldensityC+0.5))
#legPop <- c(0.03, median(data$pathos+sqrt(min(data$pathos)^2)+0.03),
#            max(data$pathos+sqrt(min(data$pathos)^2)+0.03))
#legRad <- sqrt( legPop / pi )
#hin <- par('pin')[2]
#burgPerInch <- (ylim[2]-ylim[1]) / hin
#radPerInch <- max(radius)/0.25
#heightAdj <- legRad/radPerInch*burgPerInch
#symbols( rep(2000,3), rep(-1,3), circles = legRad, inches = 0.25, add = TRUE)

par(fig=c(0,1,0.0,0.55), new=TRUE)
symbols(data$npp,data$pathos,
        circle=sqrt((data$ldensityC + abs(min(data$ldensityC)) + 0.1)/pi),
        inches=0.25, xlab="NPP", fg="black", bg="#FFC0CB99", axes=F, ylab="",
        xlim=c(0,max(data$npp)+100), ylim=c(min(data$pathos)-0.05, max(data$pathos)+0.1))
axis(side=4, at=c(-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75),
     lab=c(-1,"",-0.5,"",0,"",0.5,""),las=2, tck=-0.02)
mtext("Pathogen stress", side=4, line=3)
axis(side=1, tck=-0.02)
```

```
plot(fit2.s,add=T,rug=F,lwd=4,col="black")
lines(fit2.s)
#radius <- sqrt((data$ldensityC + abs(min(data$ldensityC)) + 0.1)/pi)
#ylim <- c(min(data$pathos)-0.1,max(data$pathos)+0.1)
#legPop <- c(min(data$ldensityC + abs(min(data$ldensityC)) + 0.1),
#            median(data$ldensityC + abs(min(data$ldensityC)) + 0.1),
#            max(data$ldensityC + abs(min(data$ldensityC)) + 0.1))
#legRad <- sqrt( legPop / pi )
#hin <- par('pin')[2]
#burgPerInch <- (ylim[2]-ylim[1]) / hin
#radPerInch <- max(radius)/0.25
#heightAdj <- legRad/radPerInch*burgPerInch
#symbols( rep(2000,3), rep(-0.8,3), circles = legRad, inches = 0.25, add = TRUE)
```

```
## Warning in symbols(data$npp, data$ldensityC, circle = sqrt((data$pathos + :
## "axes" is not a graphical parameter
```

```
## Warning in symbols(data$npp, data$pathos, circle = sqrt((data$ldensityC + :
## "axes" is not a graphical parameter
```



We also calculated R^2 values for models, where NPP explains population density and pathogen stress in low and high productivity.

```
npp.dens.low <- lm(ldensityC ~ npp, data=data[data$nrg=="lo",])
npp.dens.high <- lm(ldensityC ~ npp, data=data[data$nrg=="hi",])

npp.patho.low <- lm(pathos ~ npp, data=data[data$nrg=="lo",])
npp.patho.high <- lm(pathos ~ npp, data=data[data$nrg=="hi",])

summary(npp.dens.low)$adj.r.squared
```

```
## [1] 0.3052842
```

```
summary(npp.dens.high)$adj.r.squared
```

```
## [1] -0.009420932
```

```
summary(npp.patho.low)$adj.r.squared
```

```
## [1] 0.001589892
```

```
summary(npp.patho.high)$adj.r.squared
```

```
## [1] 0.4609742
```

4.2 Structural equation modelling (SEM)

To analyse the network of environmental factors and hunter-gatherer population density, we used `piecewiseSEM` package in *R* to perform piecewice structural equation model separately to low and high productivity regimes (see above).

4.2.1 Function for the analysis of spatial autocorraltion

Following function allows to analyse the impact of spatial autocorrelation on piecewise SEM model. It takes a single linear or generalized linear model or list of such models and longitude and latitude coordinates of data points as input. The function is modified from `lavSpatialCorrect` function by Jarret Byrnes (https://github.com/jebyrnes/spatial_correction_lavaan). This function is used below after fitting the SEM models.

```
spatialCorrectL <- function(obj,xvar,yvar, alpha=0.05){
require(ape)
require(fossil)
if(class(obj)=="lm" || class(obj)=="glm"){
    resids <- residuals(obj)
        distMat <- as.matrix(earth.dist(cbind(xvar, yvar), dist=F))
        distsInv <- 1/distMat
        diag(distsInv) <- 0
        mi <- Moran.I(resids, distsInv)
        if(mi$p.value > alpha){
            mi$n.eff <- length(resids)
            }else{
            mi$n.eff <- length(resids)*(1-mi$observed)/(1+mi$observed)
        }

        v  <- diag(vcov(obj))
        n  <- length(resids)
        v2 <- v*n/mi$n.eff
        ret <-  data.frame(Estimate=coef(obj),
```

```
                        n.eff = mi$n.eff, Std.err = sqrt(v2))
        ret[["Z-value"]] <- ret$Estimate/ret$Std.err
            ret[["P(>|z|)"]] <- 2*pnorm(abs(ret[["Z-value"]]), lower.tail=F)
        list1 <- list(mi, ret)
        names(list1) <- c(names(obj$model[1]),names(obj$model[1]))
        return(list1)

  } else {
    if(class(obj)=="list"){
    sp.corr.list <- list()
    for(i in 1:length(obj)){
    resids <- residuals(obj[[i]])
    distMat <- as.matrix(earth.dist(cbind(xvar, yvar),dist=F))
        distsInv <- 1/distMat
        diag(distsInv) <- 0
        mi <- Moran.I(resids, distsInv)
        if(mi$p.value > alpha){
            mi$n.eff <- length(resids)
            }else{
            mi$n.eff <- length(resids)*(1-mi$observed)/(1+mi$observed)
        }

        v  <- diag(vcov(obj[[i]]))
        n  <- length(resids)
        v2 <- v*n/mi$n.eff
        ret <-  data.frame(Estimate=coef(obj[[i]]),
                        n.eff = mi$n.eff, Std.err = sqrt(v2))
        ret[["Z-value"]] <- ret$Estimate/ret$Std.err
            ret[["P(>|z|)"]] <- 2*pnorm(abs(ret[["Z-value"]]), lower.tail=F)

        list1 <- list(mi, ret)
        names(list1) <- c(names(obj[[i]]$model[1]),names(obj[[i]]$model[1]))
        sp.corr.list[[i]] <- list1

    }
    return(sp.corr.list)
   } else {
    "Warning: object is not lm or glm model object or list of such model objects"
 }
}
}
```

4.2.2 Fitting of structural equation models

We started the analysis with the structural equation (SE) model, where the direct effect of NPP is missing. We built subsequent models by analysing the SE model fits and spatial autocorrelation statistics. Fits of SE models are evaluated using the test of direct separation with Fisher C as a test statistic.

4.2.2.1 SE model to the low productivity environments

First, we fitted SE models to the low produtivity environments.

```r
data.low <- data[data$nrg=="lo",]

modelList1 <- list(
    lm(ldensityC ~ biodiv + pathos, data = data.low),
    lm(biodiv ~ npp, data=data.low),
    lm(pathos ~ npp + biodiv, data=data.low))

sem.coefs(modelList1, data=data.low, standardize = "none")
```

```
##    response predictor      estimate    std.error p.value
## 1 ldensityC    biodiv 14.2803663359 1.335809e+00  0.0000 ***
## 2 ldensityC    pathos -0.9168160396 3.554910e-01  0.0105   *
## 3    biodiv       npp  0.0001433665 1.068487e-05  0.0000 ***
## 4    pathos    biodiv  1.5932111801 3.113404e-01  0.0000 ***
## 5    pathos       npp -0.0001659458 6.752612e-05  0.0147   *
```

```r
sem.coefs(modelList1, data=data.low, standardize = "scale")
```

```
##    response predictor   estimate  std.error p.value
## 1 ldensityC    biodiv  0.6007827 0.05619821  0.0000 ***
## 2 ldensityC    pathos -0.1449359 0.05619821  0.0105   *
## 3    biodiv       npp  0.6610153 0.04926436  0.0000 ***
## 4    pathos    biodiv  0.4239918 0.08285518  0.0000 ***
## 5    pathos       npp -0.2036171 0.08285518  0.0147   *
```

```r
sem.fit(modelList1, data=data.low, .progressBar = FALSE)
```

```
## $missing.paths
##            missing.path estimate std.error  df crit.value p.value
## 1 ldensityC ~ npp + ...   0.0016     4e-04 230     4.4127       0 ***
##
## $Fisher.C
##   fisher.c df p.value
## 1    22.12  2       0
##
## $AIC
##     AIC   AICc  K   n
## 1 44.12 45.309 11 234
```

```r
sem.model.fits(modelList1)
```

```
##   Class   Family     Link   n R.squared
## 1    lm gaussian identity 234 0.3315476
## 2    lm gaussian identity 234 0.4369412
## 3    lm gaussian identity 234 0.1070955
```

```r
# All the paths are significant, but the test of directed separation (sem.fit)
# indicates that the direct effect of NPP is ineed important. So, add
# the direct effect of NPP to the equation.

modelList2 <- list(
    lm(ldensityC ~ npp + biodiv + pathos, data = data.low),
    lm(biodiv ~ npp, data=data.low),
    lm(pathos ~ npp + biodiv, data=data.low))

sem.coefs(modelList2, data=data.low, standardize = "none")
```

```
##    response predictor      estimate    std.error p.value
## 1 ldensityC    biodiv  9.1700260235 1.730166e+00  0.0000 ***
## 2 ldensityC       npp  0.0015896879 3.602551e-04  0.0000 ***
## 3 ldensityC    pathos -0.6727461044 3.465199e-01  0.0534
## 4    biodiv       npp  0.0001433665 1.068487e-05  0.0000 ***
## 5    pathos    biodiv  1.5932111801 3.113404e-01  0.0000 ***
## 6    pathos       npp -0.0001659458 6.752612e-05  0.0147   *
```

```
sem.coefs(modelList2, data=data.low, standardize = "scale")
```

```
##    response predictor   estimate  std.error p.value
## 1 ldensityC    biodiv  0.3857879 0.07278900  0.0000 ***
## 2 ldensityC       npp  0.3083569 0.06987985  0.0000 ***
## 3 ldensityC    pathos -0.1063519 0.05478001  0.0534
## 4    biodiv       npp  0.6610153 0.04926436  0.0000 ***
## 5    pathos    biodiv  0.4239918 0.08285518  0.0000 ***
## 6    pathos       npp -0.2036171 0.08285518  0.0147   *
```

```
sem.model.fits(modelList2)
```

```
##   Class   Family     Link   n R.squared
## 1    lm gaussian identity 234 0.3837214
## 2    lm gaussian identity 234 0.4369412
## 3    lm gaussian identity 234 0.1070955
```

```
# Now the model is fully saturated and sem.fit cannot give any estimate
# of model fit (naturally). However, now the direct effect of pathogens
# on population density is not significant, so we can remove it from the
# equation

modelList3 <- list(
    lm(ldensityC ~ npp + biodiv, data = data.low),
    lm(biodiv ~ npp, data=data.low),
    lm(pathos ~ npp + biodiv, data=data.low))

sem.coefs(modelList3, data=data.low, standardize = "none")
```

```
##    response predictor      estimate    std.error p.value
## 1 ldensityC    biodiv  8.0981994085 1.649519e+00  0.0000 ***
## 2 ldensityC       npp  0.0017013274 3.577615e-04  0.0000 ***
## 3    biodiv       npp  0.0001433665 1.068487e-05  0.0000 ***
## 4    pathos    biodiv  1.5932111801 3.113404e-01  0.0000 ***
## 5    pathos       npp -0.0001659458 6.752612e-05  0.0147   *
```

```
sem.coefs(modelList3, data=data.low, standardize = "scale")
```

```
##    response predictor   estimate  std.error p.value
## 1 ldensityC    biodiv  0.3406956 0.06939615  0.0000 ***
## 2 ldensityC       npp  0.3300119 0.06939615  0.0000 ***
## 3    biodiv       npp  0.6610153 0.04926436  0.0000 ***
## 4    pathos    biodiv  0.4239918 0.08285518  0.0000 ***
## 5    pathos       npp -0.2036171 0.08285518  0.0147   *
```

```
sem.fit(modelList3, data=data.low, .progressBar = FALSE)
```

```
## $missing.paths
##               missing.path estimate std.error  df crit.value p.value
## 1 pathos ~ ldensityC + ...   -0.024    0.0123 230    -1.9414  0.0534
```

```
##
## $Fisher.C
##   fisher.c df p.value
## 1     5.86  2   0.053
##
## $AIC
##     AIC   AICc  K   n
## 1 27.86 29.049 11 234
```

```
sem.model.fits(modelList3)
```

```
##   Class   Family     Link   n R.squared
## 1    lm gaussian identity 234 0.3736220
## 2    lm gaussian identity 234 0.4369412
## 3    lm gaussian identity 234 0.1070955
```

```
# All fine for now, but we need to test how spatial autocorrelation
# influences our results. For full results (expected and observed Moran I, etc.)
# use spatialCorrectL(modelList3,data.low$longitude,data.low$latitude) without
# subsetting.

# First, effects on population density
spatialCorrectL(modelList3,data.low$longitude,data.low$latitude)[[1]][[2]]
```

```
##                 Estimate    n.eff      Std.err   Z-value      P(>|z|)
## (Intercept) -0.839711747 166.0504 0.2837783316 -2.959041 3.085979e-03
## npp          0.001701327 166.0504 0.0004246994  4.005957 6.176694e-05
## biodiv       8.098199409 166.0504 1.9581472196  4.135644 3.539609e-05
```

```
# Next, effects on biodiversity
spatialCorrectL(modelList3,data.low$longitude,data.low$latitude)[[2]][[2]]
```

```
##                 Estimate    n.eff     Std.err   Z-value      P(>|z|)
## (Intercept) 0.0869872837 143.4852 0.009718255  8.950916 3.525441e-19
## npp         0.0001433665 143.4852 0.000013645 10.506890 8.029820e-26
```

```
# Lastly, effects on pathogen stress
spatialCorrectL(modelList3,data.low$longitude,data.low$latitude)[[3]][[2]]
```

```
##                  Estimate    n.eff      Std.err    Z-value      P(>|z|)
## (Intercept) -0.9421943877 88.63548 0.0733117369 -12.851890 8.393402e-38
## npp         -0.0001659458 88.63548 0.0001097175  -1.512483 1.304111e-01
## biodiv       1.5932111801 88.63548 0.5058708072   3.149443 1.635821e-03
```

```
# Reults suggest that effect of NPP on pathogens is not
# significant when spatial autocottelation present in the data
# is taken into account. We can remove that from the equation

modelList4 <- list(
    lm(ldensityC ~ npp + biodiv, data = data.low),
    lm(biodiv ~ npp, data=data.low),
    lm(pathos ~ biodiv, data=data.low))

sem.coefs(modelList4, data=data.low, standardize = "none")
```

```
##    response predictor     estimate    std.error p.value
## 1 ldensityC    biodiv 8.0981994085 1.649519e+00       0 ***
## 2 ldensityC       npp 0.0017013274 3.577615e-04       0 ***
```

```
## 3    biodiv       npp 0.0001433665 1.068487e-05       0 ***
## 4    pathos    biodiv 1.0874545630 2.361449e-01       0 ***
```

```
sem.coefs(modelList4, data=data.low, standardize = "scale")
```

```
##    response predictor  estimate  std.error p.value
## 1 ldensityC    biodiv 0.3406956 0.06939615       0 ***
## 2 ldensityC       npp 0.3300119 0.06939615       0 ***
## 3    biodiv       npp 0.6610153 0.04926436       0 ***
## 4    pathos    biodiv 0.2893978 0.06284384       0 ***
```

```
sem.fit(modelList4, data=data.low, .progressBar = FALSE)
```

```
## $missing.paths
##               missing.path estimate std.error  df crit.value p.value
## 1       pathos ~ npp + ...  -0.0002    0.0001 231    -2.4575  0.0147 *
## 2 pathos ~ ldensityC + ...  -0.0240    0.0123 230    -1.9414  0.0534
##
## $Fisher.C
##   fisher.c df p.value
## 1     14.3  4   0.006
##
## $AIC
##    AIC   AICc  K   n
## 1 34.3 35.287 10 234
```

```
sem.model.fits(modelList4)
```

```
##   Class   Family     Link   n  R.squared
## 1    lm gaussian identity 234 0.37362204
## 2    lm gaussian identity 234 0.43694119
## 3    lm gaussian identity 234 0.08375111
```

Above model (4) is our final model (coefficients) for low productivity, even though Fisher C indicates that important paths are missing. However, these paths are not significant when spatial autocorrelation is taken into account (see above).

4.2.2.2 SE model to the high productivity environments

Again, we started with the model, where direct effect of NPP is missing.

```
data.high <- data[data$nrg=="hi",]

modelList5 <- list(
    lm(ldensityC ~ biodiv + pathos, data = data.high),
    lm(biodiv ~ npp, data=data.high),
    lm(pathos ~ npp + biodiv, data=data.high))

sem.coefs(modelList5, data=data.high, standardize = "none")
```

```
##    response predictor      estimate    std.error p.value
## 1 ldensityC    pathos -2.0085485744 3.722668e-01  0.0000 ***
## 2 ldensityC    biodiv  1.5745058270 1.374214e+00  0.2562
## 3    biodiv       npp  0.0001203376 2.449572e-05  0.0000 ***
## 4    pathos       npp  0.0005138889 8.997246e-05  0.0000 ***
## 5    pathos    biodiv  0.5620815224 3.912448e-01  0.1558
```

```
sem.coefs(modelList5, data=data.high, standardize = "scale")
```

```
##    response predictor   estimate std.error p.value
## 1 ldensityC    pathos -0.6282078 0.1164328  0.0000 ***
## 2 ldensityC    biodiv  0.1334029 0.1164328  0.2562
## 3    biodiv       npp  0.5232874 0.1065195  0.0000 ***
## 4    pathos       npp  0.6053524 0.1059860  0.0000 ***
## 5    pathos    biodiv  0.1522647 0.1059860  0.1558
```

```
sem.fit(modelList5, data=data.high, .progressBar = FALSE)
```

```
## $missing.paths
##            missing.path estimate std.error df crit.value p.value
## 1 ldensityC ~ npp + ...   0.0016     4e-04 62     4.5482       0 ***
##
## $Fisher.C
##   fisher.c df p.value
## 1    21.14  2       0
##
## $AIC
##     AIC   AICc  K  n
## 1 43.14 48.029 11 66
```

```
sem.model.fits(modelList5)
```

```
##   Class   Family     Link  n R.squared
## 1    lm gaussian identity 66 0.3338261
## 2    lm gaussian identity 66 0.2738298
## 3    lm gaussian identity 66 0.4861029
```

```
# Results suggest that the direct effect should be included.
# Results also indicate that the effects of biodiversity are not
# significant, so we can remove them from the equation.

modelList6 <- list(
    lm(ldensityC ~ npp + pathos, data = data.high),
    lm(biodiv ~ npp, data=data.high),
    lm(pathos ~ npp, data=data.high))

sem.coefs(modelList6, data=data.high, standardize = "none")
```

```
##    response predictor      estimate    std.error p.value
## 1 ldensityC    pathos -3.0849956996 3.908940e-01       0 ***
## 2 ldensityC       npp  0.0015818781 3.318333e-04       0 ***
## 3    biodiv       npp  0.0001203376 2.449572e-05       0 ***
## 4    pathos       npp  0.0005815285 7.730527e-05       0 ***
```

```
sem.coefs(modelList6, data=data.high, standardize = "scale")
```

```
##    response predictor   estimate  std.error p.value
## 1 ldensityC    pathos -0.9648849 0.12225874       0 ***
## 2 ldensityC       npp  0.5828180 0.12225874       0 ***
## 3    biodiv       npp  0.5232874 0.10651953       0 ***
## 4    pathos       npp  0.6850306 0.09106429       0 ***
```

```
sem.model.fits(modelList6)
```

```
##   Class   Family     Link  n R.squared
```

```
## 1    lm gaussian identity 66 0.5002227
## 2    lm gaussian identity 66 0.2738298
## 3    lm gaussian identity 66 0.4692669
```

```
# Analysis for spatial autocorrelation.
# First, effects on population density.
spatialCorrectL(modelList6,data.high$longitude,data.high$latitude)[[1]][[2]]
```

```
##                 Estimate n.eff      Std.err    Z-value      P(>|z|)
## (Intercept) -0.503313913    66 0.7273633629 -0.6919704 4.889559e-01
## npp          0.001581878    66 0.0003318333  4.7670868 1.869088e-06
## pathos      -3.084995700    66 0.3908939672 -7.8921548 2.970130e-15
```

```
# Effects on biodiversity.
spatialCorrectL(modelList6,data.high$longitude,data.high$latitude)[[2]][[2]]
```

```
##                 Estimate    n.eff      Std.err  Z-value      P(>|z|)
## (Intercept) 0.0738451211 37.58252 5.876484e-02 1.256621 0.2088909995
## npp         0.0001203376 37.58252 3.246155e-05 3.707081 0.0002096621
```

```
# Effects on pathogen stress.
spatialCorrectL(modelList6,data.high$longitude,data.high$latitude)[[3]][[2]]
```

```
##                  Estimate    n.eff      Std.err   Z-value      P(>|z|)
## (Intercept) -1.4862861275 25.99877 0.2229733587 -6.665757 2.633054e-11
## npp          0.0005815285 25.99877 0.0001231699  4.721351 2.342829e-06
```

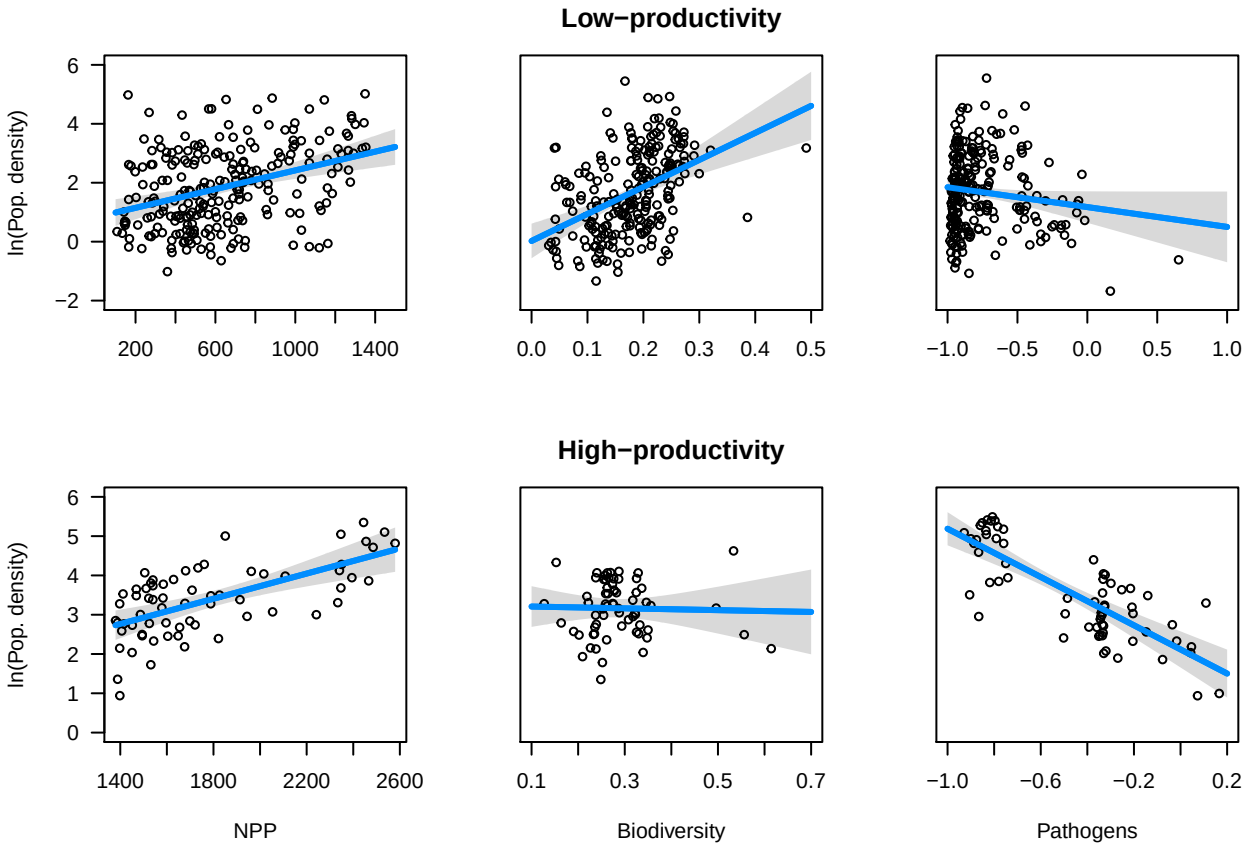
Above model (6) will our final model (coefficients) for high productivity, because analysis of spatial autocorrelation suggests that all paths are significant even when spatial autocorrelation is taken into account.

4.2.3 Partial residuals and predictive power of the SE model

We plotted partial residual plots of the effects of the predictors in the SE-model, and calculated the predictive power of the piecewice model in the whole data (low and high combined).

```
# Fit component models explaining population density
fit.dens.low <- lm(ldensityC ~ npp + biodiv, data=data[data$nrg=="lo",])
fit.dens.high <- lm(ldensityC ~ npp + pathos, data=data[data$nrg=="hi",])

# Fit component models that also include non-significant predictors
fit.dens.low2 <- lm(ldensityC ~ npp + biodiv + pathos, data=data[data$nrg=="lo",])
fit.dens.high2 <- lm(ldensityC ~ npp + biodiv + pathos, data=data[data$nrg=="hi",])

# Partial residual plots for models that include also non-significant predictors.
# The same code can be applied to the models including only significant predictors
par(mfrow=c(2,3), mar=c(4,4,3,0.5))
visreg(fit.dens.low2, "npp", points=list(cex=0.8, pch=1, col="black"), xlab="",
       ylab="ln(Pop. density)", ylim=c(-2,6), xlim=c(100,1500))
visreg(fit.dens.low2, "biodiv", points=list(cex=0.8, pch=1, col="black"),
       main="Low-productivity", xlab="", ylab="", ylim=c(-2,6), xlim=c(0,0.5), yaxt="n")
visreg(fit.dens.low2, "pathos", points=list(cex=0.8, pch=1, col="black"),
       xlab="", ylab="", ylim=c(-2,6), xlim=c(-1,1), yaxt="n")
visreg(fit.dens.high2, "npp", points=list(cex=0.8, pch=1, col="black"),
       xlab="NPP", ylab="ln(Pop. density)", ylim=c(0,6))
visreg(fit.dens.high2, "biodiv", points=list(cex=0.8, pch=1, col="black"),
       main="High-productivity", xlab="Biodiversity", ylab="", ylim=c(0,6),
       xlim=c(0.1,0.7), yaxt="n")
```

```
visreg(fit.dens.high2, "pathos", points=list(cex=0.8, pch=1, col="black"),
       xlab="Pathogens", ylab="", ylim=c(0,6), xlim=c(-1,0.2), yaxt="n")
```




```
# Predictive power

pred <-vector()
for(i in 1:nrow(data)){
        case <- data[i,]
        if(case$nrg == "lo"){
            pred[i] <- predict(fit.dens.low, case)
        } else {
            pred[i] <- predict(fit.dens.high, case)
        }
    return=pred
}

# Adjusted R square

summary(lm(data$ldensityC~pred))$adj.r.squared
```

```
## [1] 0.5291926
```

4.2.4 Direct, indirect, and total effects of predictors on population density

These effects are based on standardised regression coefficients.

```r
# Direct effects (all)

coeffs.low <- sem.coefs(modelList4, data=data.low, standardize = "scale")[,1:3]
coeffs.high <- sem.coefs(modelList6, data=data.high, standardize = "scale")[,1:3]

# For low productivity environments:
# Direct effect of npp on population density.

npp.dir.low <- coeffs.low[2,3]

# Indirect effects of npp on population
# density via biodiversity.

npp.ind.low <- coeffs.low[3,3] * coeffs.low[1,3]

# Total effect of npp on population density.

npp.tot.low <- npp.dir.low + npp.ind.low


# Direct, indirect and total effects of
# biodiversity on population density.

biod.dir.low <- coeffs.low[1,3]
biod.ind.low <- NA
biod.tot.low <- coeffs.low[1,3]

# Direct, indirect and total effects of
# pathogen stress on population density.

path.dir.low <- NA
path.ind.low <- NA
path.tot.low <- NA

# For high productivity environments:
# Direct effect of npp on population density.

npp.dir.high <- coeffs.high[2,3]

# Indirect effects of npp on population
# density via pathogen stress.

npp.ind.high <- coeffs.high[4,3] * coeffs.high[1,3]

# Total effect of npp on population density.

npp.tot.high <- npp.dir.high + npp.ind.high

# Direct, indirect and total effects of
# biodiversity on population density.

biod.dir.high <- NA
biod.ind.high <- NA
```

```
biod.tot.high <- NA


# Direct, indirect and total effects of
# pathogen stress on population density.

path.dir.high <- coeffs.high[1,3]
path.ind.high <- NA
path.tot.high <- coeffs.high[1,3]
```

Tables of effects

Table 1: Effects in low productivity environments

Predictor variable	Direct effect	Indirect effect	Total effect
NPP	0.3300119	0.225205	0.5552169
Biodiversity	0.3406956	NA	0.3406956
Pathogen stress	NA	NA	NA

Table 2: Effects in high productivity environments

Predictor variable	Direct effect	Indirect effect	Total effect
NPP	0.582818	-0.6609757	-0.0781577
Biodiversity	NA	NA	NA
Pathogen stress	-0.9648849	NA	-0.9648849

4.2.5 Linear model with interactions

In order to see whether the same kind of pattern can be observed when the data are not split into low- and high-productivity sets, we performed linear model to explain population density using the whole dataset. In this model we included interactions between NPP and pathogen stress and between NPP and biodiversity. These results show the same pattern and strongly suggest that the differences between low- and high-productivity environments observed in the SEM analyses is not an artefact of splitting the data.

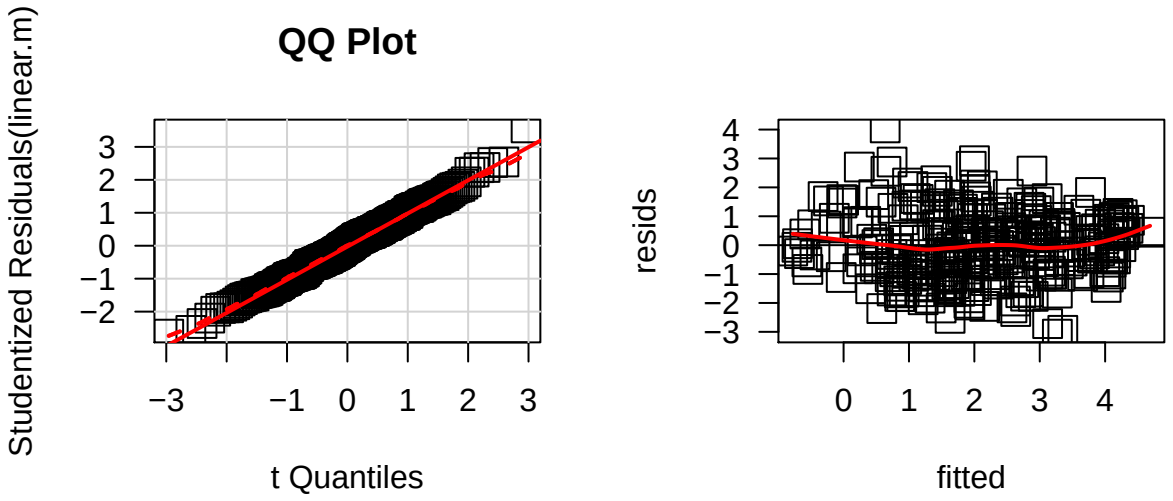
```
linear.m <- lm(ldensityC ~ npp + biodiv + pathos + npp:pathos + npp:biodiv, data=data)

# center variables to get meaningful main effects and p-values
summ(linear.m, vifs=TRUE, center=TRUE)
```

```
## MODEL INFO:
## Observations: 300
## Dependent Variable: ldensityC
##
## MODEL FIT:
## F(5,294) = 70.809, p = 0
## R-squared = 0.546
## Adj. R-squared = 0.539
##
## Standard errors: OLS
##             Est.   S.E.  t val. p         VIF
## (Intercept) 2.425  0.079 30.886 0     ***
```

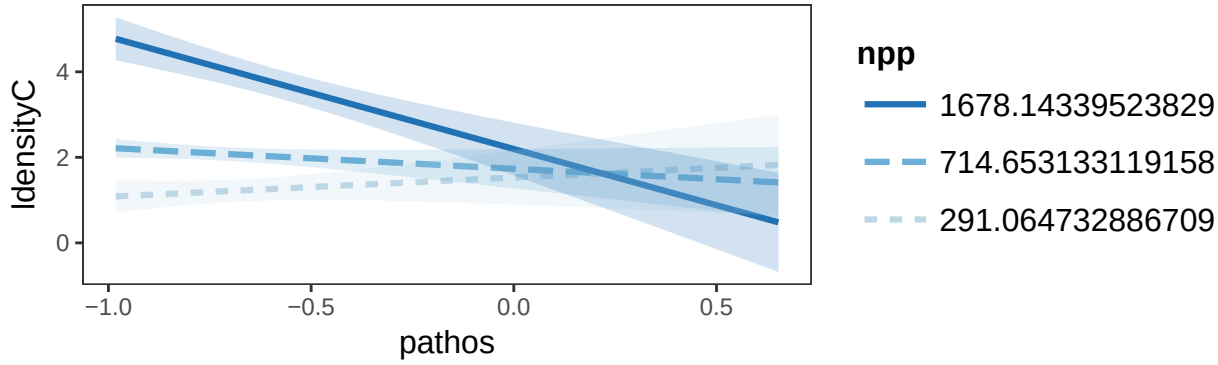
```
## npp          0.002  0     10.366 0     *** 2.738
## biodiv       6.31   1.226 5.148  0     *** 2.364
## pathos       -0.876 0.275 -3.183 0.002 **  1.464
## npp:pathos   -0.002 0.001 -4.249 0     *** 2.855
## npp:biodiv   -0.003 0.002 -1.975 0.049 *   2.344
##
## All continuous variables are mean-centered.
```

Model diagnostics and interactive effects show that the model performs reasonably well and that the relationships between variables resemble to that of the SEM results.

```
par(mfrow=c(1,2))
qqPlot(linear.m, main="QQ Plot", col="black",pch=0,cex=2, col.lines="red",envelope=F,
       las=1,line="quartiles")
resids <- as.vector(linear.m$residuals)
fitted <- as.vector(linear.m$fitted.values)
plot(fitted,resids, las=1, pch=0, cex=2)
abline(lm(resids~fitted))
lines(fitted[order(fitted)],predict(loess(resids~fitted, span=0.75))[order(fitted)],
      lwd=2, col="red")
```



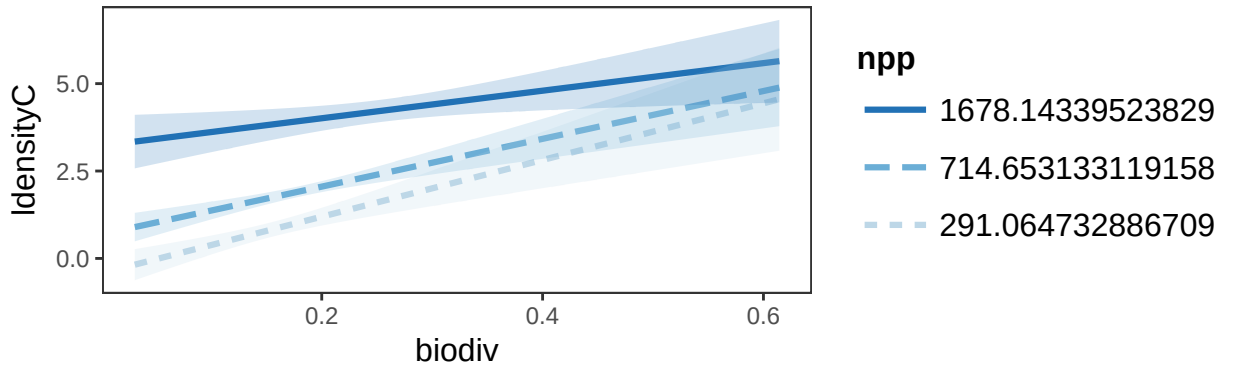

```
interact_plot(linear.m, pred = "pathos", modx = "npp", interval = TRUE,
modxvals=quantile(data$npp, prob=c(0.1,0.5,0.9)))
```



The above figure shows how the effect of pathogen stress on population density changes when NPP changes

(10th, 50th, and 90th percentiles of NPP).

```
interact_plot(linear.m, pred = "biodiv", modx = "npp", interval = TRUE,
modxvals=quantile(data$npp, prob=c(0.1,0.5,0.9)))
```



The Above figure shows how the effect of biodiversity on population density changes when NPP changes (10th, 50th, and 90th percentiles of NPP).

4.3 Global projection of hunter-gatherer population density and its constraints

We produced the global projection using a combination (average) of two modelling techniques: OLS linear regression (LM) and support vector regression (SVR) with linear kernel. We fitted these modelling algorithms to the whole data, i.e., over the entire NPP gradient. We used resulting models to predict global ln(population density) using global environmental variables (see above) as predictor variables. Both of the models include interactions between NPP and biodiversity and between NPP and pathogen stress.

4.3.1 Performance of the prediction model

We evaluated the preformance of the averaged prediction model by *h*-block CV. We used 500 km as a *h*-value in the CV. We adapted the code for *h*-block CV from the supplement of the Salonen et al. (2016) (http://www.helsinki.fi/~ssalonen/txt/Salonen_ea2016-Holocene-Tree_ensembles-v2.pdf).

```
# Hyperparameter values for the SVR model
kernel <- "linear"
cost <- 0.5
epsilon <- 0.1
```

```
distMat <- as.matrix(earth.dist(cbind(data$longitude,data$latitude),dist=F))
h.value=500
preds.ens <- vector(mode = "numeric", length = nrow(data))

for (i in 1:nrow(data)) {
        test.rows <- 1:nrow(data) == i                    # test row indices
                                                          # for iteration i

        train.rows <- !test.rows & distMat[, i] > h.value # train row indices
                                                          # for iteration i

        train.iter <- data[train.rows, ]                 # training data used for
                                                       # the iteration i

        test.iter <- data[test.rows, ]                   # test data used for
```

```
                                                    # the iteration i

    lm.m <- lm(ldensityC ~ npp + biodiv + pathos + npp:pathos + npp:biodiv,
                   data=train.iter)
    svm.m <- svm(ldensityC ~ npp + biodiv + pathos + npp:pathos + npp:biodiv,
                     kernel=kernel, cost=cost, epsilon=epsilon, data = train.iter)

        lm.p <- predict(lm.m, newdata=test.iter)
        svm.p <- predict(svm.m, type="response", newdata=test.iter)

        preds.ens[i] <- rowMeans(cbind(lm.p, svm.p))

}


    rmse <- sqrt(mean((preds.ens - data$ldensityC)^2))
    rmse
```

```
## [1] 1.26164
```

```
    r2 <- summary(lm(data$ldensityC~preds.ens))$adj.r.squared
    r2
```

```
## [1] 0.4615028
```

H-block cross validated RMSE (1.26) and R^2 (0.46) indicate that the ensemble model is performing well with the data that algorithms haven't seen before.

The same approach can be used to evaluate individual techniques:

```
# First for the linear model
distMat <- as.matrix(earth.dist(cbind(data$longitude,data$latitude),dist=F))
h.value=500
preds.lm <- vector(mode = "numeric", length = nrow(data))

for (j in 1:nrow(data)) {
        test.rows <- 1:nrow(data) == j
        train.rows <- !test.rows & distMat[, j] > h.value
        train.iter <- data[train.rows, ]
        test.iter <- data[test.rows, ]
        lm.l <- lm(ldensityC ~ npp + biodiv + pathos + npp:pathos + npp:biodiv,
        data=train.iter)
        preds.lm[j] <- predict(lm.l, newdata = test.iter)
        preds.lm
}

# RMSE
sqrt(mean((preds.lm - data$ldensityC)^2))
```

```
## [1] 1.253505
```

```
# R2
summary(lm(data$ldensityC~preds.lm))$adj.r.squared
```

```
## [1] 0.4642355
```

```
# bias (under- or overestimation)
abs(1-coefficients(lm(data$ldensityC~preds.lm))[[2]])
```

```
## [1] 0.03790814
```

```
# Then for the SVR
distMat <- as.matrix(earth.dist(cbind(data$longitude,data$latitude),dist=F))
h.value=500
preds.svm <- vector(mode = "numeric", length = nrow(data))

for (j in 1:nrow(data)) {
        test.rows <- 1:nrow(data) == j
        train.rows <- !test.rows & distMat[, j] > h.value
        train.iter <- data[train.rows, ]
        test.iter <- data[test.rows, ]
        svm.l <- svm(ldensityC ~ npp + biodiv + pathos + npp:pathos + npp:biodiv,
        data=train.iter, kernel="linear",  cost=0.5, epsilon=0.1)
        preds.svm[j] <- predict(svm.l, newdata = test.iter)
        preds.svm
}

# RMSE
sqrt(mean((preds.svm - data$ldensityC)^2))
```

```
## [1] 1.275245
```

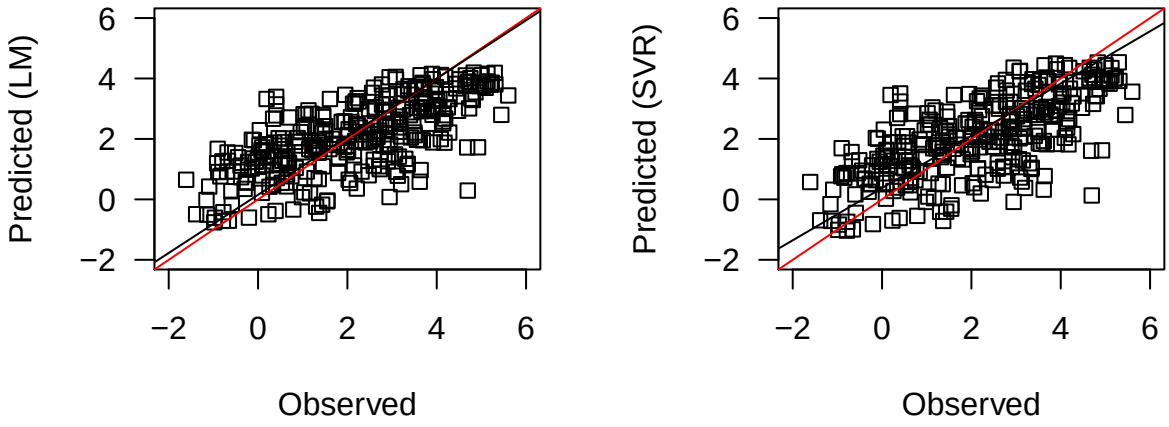
```
# R2
summary(lm(data$ldensityC~preds.svm))$adj.r.squared
```

```
## [1] 0.457514
```

```
# bias (under- or overestimation)
abs(1-coefficients(lm(data$ldensityC~preds.svm))[[2]])
```

```
## [1] 0.1353713
```

```
par(mfrow=c(1,2))
plot(data$ldensityC, preds.lm, pch=0, ylim=c(-2,6), xlim=c(-2,6), las=1,
     ylab="Predicted (LM)", xlab="Observed")
abline(0,1, col="red")
abline(lm(data$ldensityC~preds.lm))

plot(data$ldensityC, preds.svm, pch=0, ylim=c(-2,6), xlim=c(-2,6), las=1,
     ylab="Predicted (SVR)", xlab="Observed")
abline(0,1, col="red")
abline(lm(data$ldensityC~preds.svm))
```

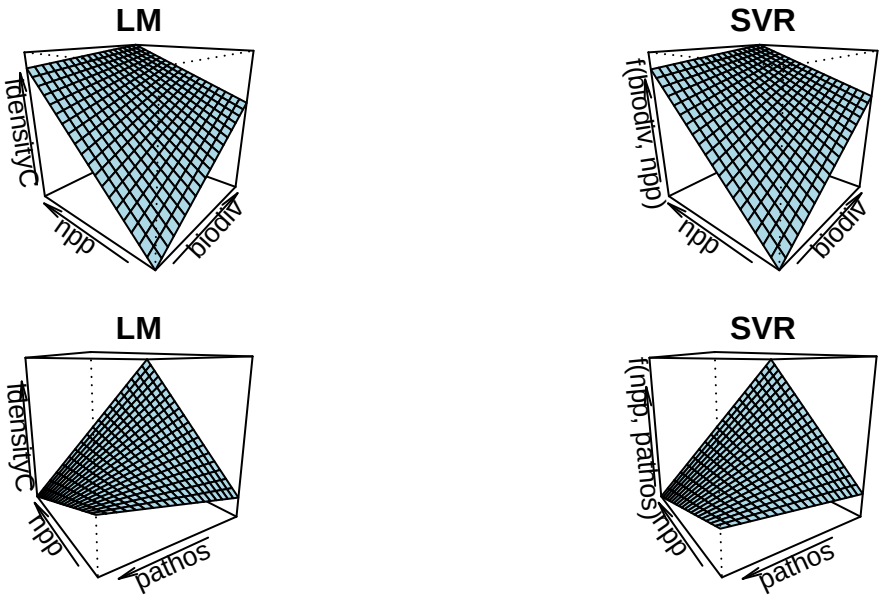


Final models used in the prediction

```
lm.mod <- lm(ldensityC ~ npp + biodiv + pathos + npp:pathos + npp:biodiv, data=data)
svm.mod <- svm(ldensityC ~ npp + biodiv + pathos + npp:pathos + npp:biodiv,
               kernel="linear", cost=0.5, epsilon=0.1, data=data)
```

Plots of the effects of the three predictors on population density

```
par(mfcol=c(2,2), mar=c(1,1,1,1))

visreg2d(lm.mod, "biodiv", "npp", plot.type="persp", col="lightblue",
         theta=-50, phi=20, nn=20, main="LM", ticktype = "simple")
visreg2d(lm.mod, "npp", "pathos", plot.type="persp", col="lightblue",
         theta=-123, phi=14, nn=20, main="LM", ticktype = "simple")

visreg2d(svm.mod, "biodiv", "npp", plot.type="persp",col="lightblue",
         theta=-50, phi=20, nn=20, main="SVR", ticktype = "simple")
visreg2d(svm.mod, "npp", "pathos", plot.type="persp",col="lightblue",
         theta=-123, phi=14, nn=20, main="SVR", ticktype = "simple")
```



4.3.2 Global dataset of predicted hunter-gatherer population density

Here, we use the combination of the two models to construct a global projection hunter-gatherer population density. This projection or prediction assumes that the whole world would be populated by hunter-gatherers only. Global map shows the distribution of hunter-gatherer population density given the effects of NPP, biodiversity, and pathogen stress.

```
# Create global predictor dataset, which is based on the predictor
# raster stack created above

global.data <- as.data.frame(predictors, xy=TRUE)
names(global.data) <- c("x","y","npp","biodiv","pathos")

# Subset NA:s as a separate dataset.

globalNA <- as.data.frame(global.data[!complete.cases(global.data),])
row.names(globalNA)   <- NULL

# add four "empty" variables, i.e., population densities predicted by three different
# modelling techniques.


globalNA$ldensitylm  <- NA
globalNA$ldensitysvm  <- NA

# Subset all the non-NA:s as a separate dataset.

global.data2 <- as.data.frame(global.data[complete.cases(global.data),])
row.names(global.data2) <- NULL

# Predict h-g population density using the fitted model algorithms (see above)

global.data2$ldensitylm <- predict(lm.mod, newdata=global.data2)
```

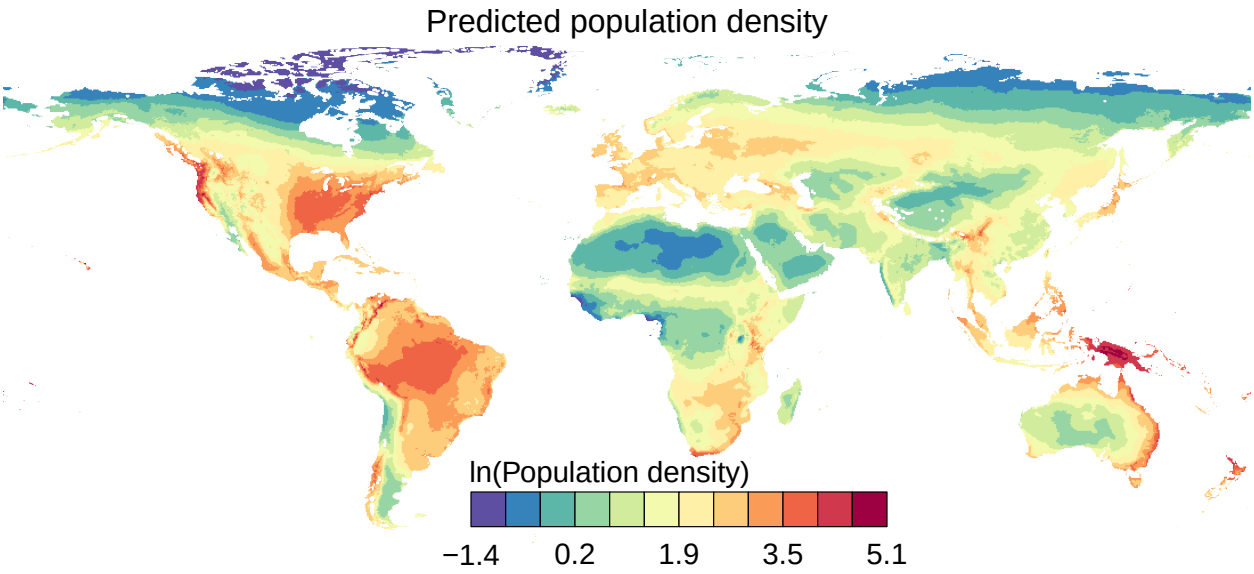
```
global.data2$ldensitysvm <- predict(svm.mod, type="response", newdata=global.data2)

# Combine predictions and NA:s

global.preds <- rbind(global.data2,globalNA)

# Calculate ensemble projection (mean of three predictions)

global.preds$lensemble <- apply(global.preds[,6:7], 1, mean)

# Add variable indicating high population density areas (pop.dens. > 85th percentile)

global.preds$hidens <- ifelse(global.preds$lensemble > quantile(global.preds$lensemble,
                                                                0.85, na.rm=T),1,0)

# Rasterize the global projection

global.stack <- rasterFromXYZ(global.preds)
```

Next, the global map pf predicted population density. The same code can be used for creating maps of LM and SVR predictions individually.

```
# Plot the ensemble prediction
# First, download glaciers data from http://www.naturalearthdata.com/
# (just for the illustration), then define breaks and colors for mapping density,
# and finally plot the map.

# Glaciers data

URL <-paste("http://www.naturalearthdata.com/",
            "http//www.naturalearthdata.com/","download/50m/physical/",
            "ne_50m_glaciated_areas.zip", sep="")

wd <- getwd()
td <- tempdir()
setwd(td)
temp <- tempfile(fileext = ".zip")
download.file(URL, temp, mode="wb")
unzip(temp)
shp <- dir(tempdir(), "*.shp$")
lyr <- sub(".shp$", "", shp)
glaciers <- readOGR(dsn=shp, layer=lyr)
unlink(dir(td))
setwd(wd)

crs(glaciers) <- "+proj=longlat +datum=WGS84"

# Take ensemble projection as a separate layer

lpdensity <- global.stack[[6]]

# Define breaks in the ln(population density)

t.steps <- 0:12
```

```
x.min <- cellStats(lpdensity, stat="min")
step.len <- (cellStats(lpdensity, stat="max")-cellStats(lpdensity, stat="min"))/12
breaks <- vector()
for(i in 1:length(t.steps)){
    breaks[i] <- x.min+t.steps[i]*step.len
    return=breaks
}


# Define colors

col <- c("#5E4FA2", "#3682BA", "#5CB7A9", "#98D5A4", "#D0EC9C", "#F3FAAD", "#FEF0A7",
       "#FDCD7B", "#FA9C58", "#EE6445", "#CF384D", "#9E0142")
```

```
# Finally, the global map itself

par(mar=c(0,0,0,0))
image(lpdensity, col=col, breaks=breaks, asp=1, main="",xlab="",ylab="",
      axes=F, xlim=c(-180,180))
text(0,90, "Predicted population density")
plot(glaciers,col="white",border="white", add=T)
# Legend
legend_cat_width <- 10
legend_height <- 10
legend_xleft <- -45
legend_ybottom <- -55
legend_ytop <- legend_ybottom + legend_height
for (k in 1:length(col)) {
  rect(legend_xleft + (k-1)*legend_cat_width, legend_ybottom,
       legend_xleft + k*legend_cat_width,
  legend_ytop, border="black", col=col[k], lwd=0.6)
}
text(legend_xleft+4*legend_cat_width, legend_ytop+legend_height/2,
labels = "ln(Population density)", family="sans", cex=0.9)
text(c(0,3,6,9,12)*legend_cat_width + legend_xleft,
     rep(legend_ybottom-0.8*legend_height, 3),
     labels=c(round(breaks[1],1), round(breaks[4],1), round(breaks[7],1),
              round(breaks[10],1), round(breaks[13],1)),family="sans", cex=.9)
```

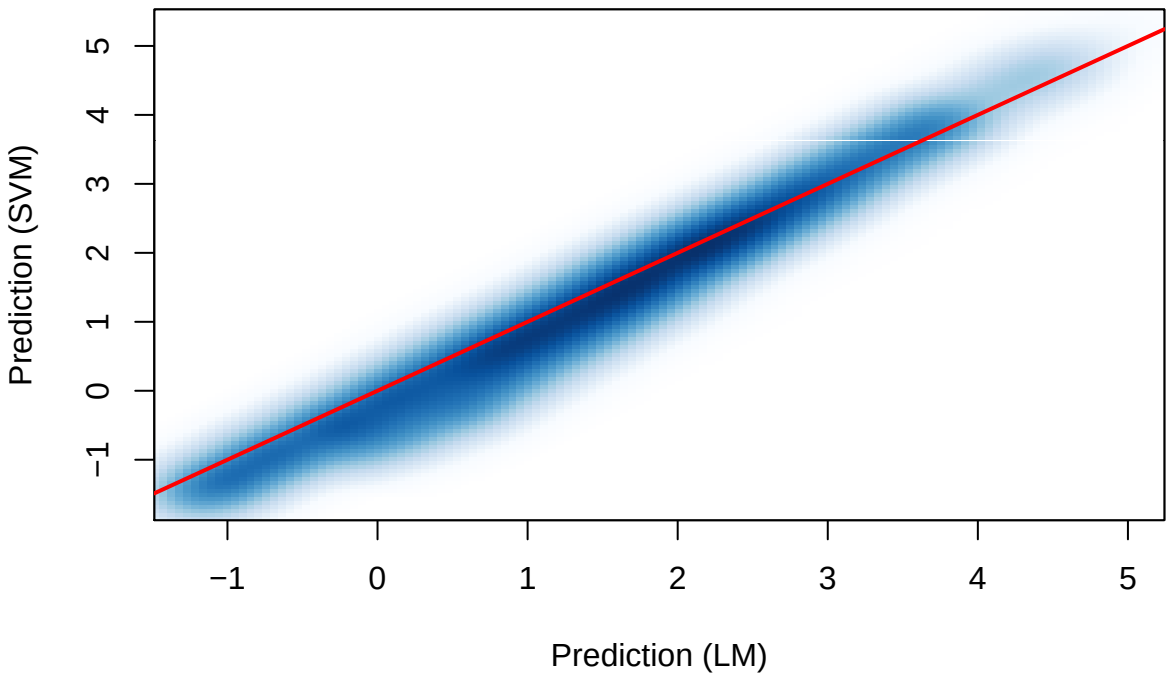


Correlation between LM and SVR global predictions

```
global_lm <- global.stack[[4]]

global_svm <- global.stack[[5]]

lm_svm_stack <- stack(global_lm, global_svm)

layerStats(lm_svm_stack, "pearson", na.rm=TRUE)
```

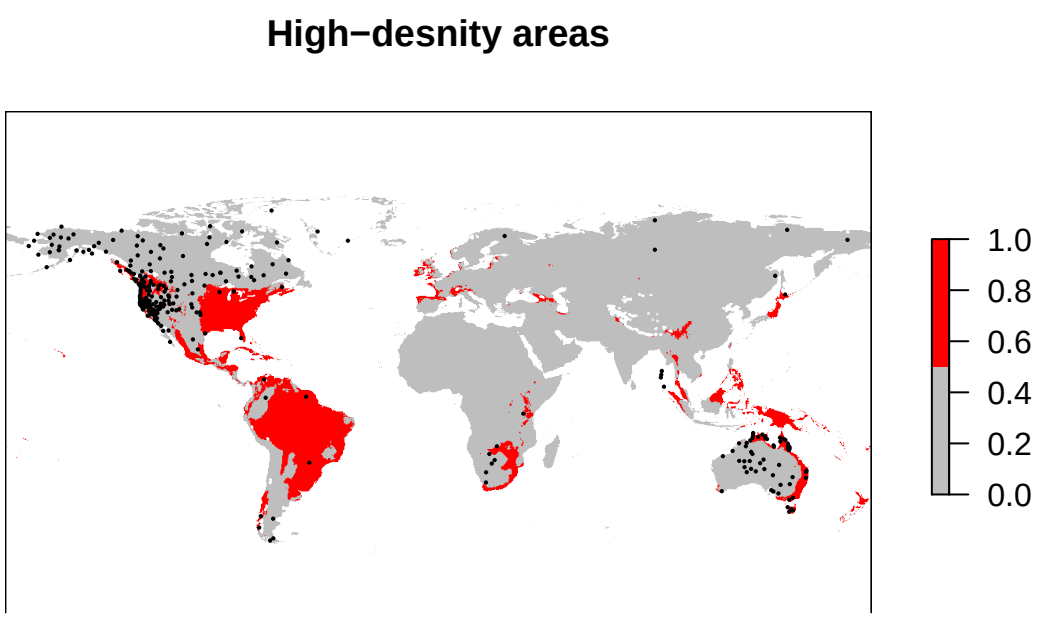
```
## $`pearson correlation coefficient`
##             ldensitylm ldensitysvm
## ldensitylm   1.0000000   0.9922772
## ldensitysvm  0.9922772   1.0000000
##
## $mean
##  ldensitylm ldensitysvm
##    1.365215    1.175980
```

```
smoothScatter(values(global_lm), values(global_svm), nrpoints=0,
              xlab="Prediction (LM)", ylab="Prediction (SVM)")
abline(0,1, col="red", lwd=2)
```



High-desnity areas (population density > 85th percentile)

```
plot(global.stack[[7]], main="High-desnity areas",xlab="",ylab="", col=c("grey","red"), axes=F)
plot(glaciers,col="white",border="white", add=T)
points(data$longitude,data$latitude, pch=16, col="black", cex=0.3)
```



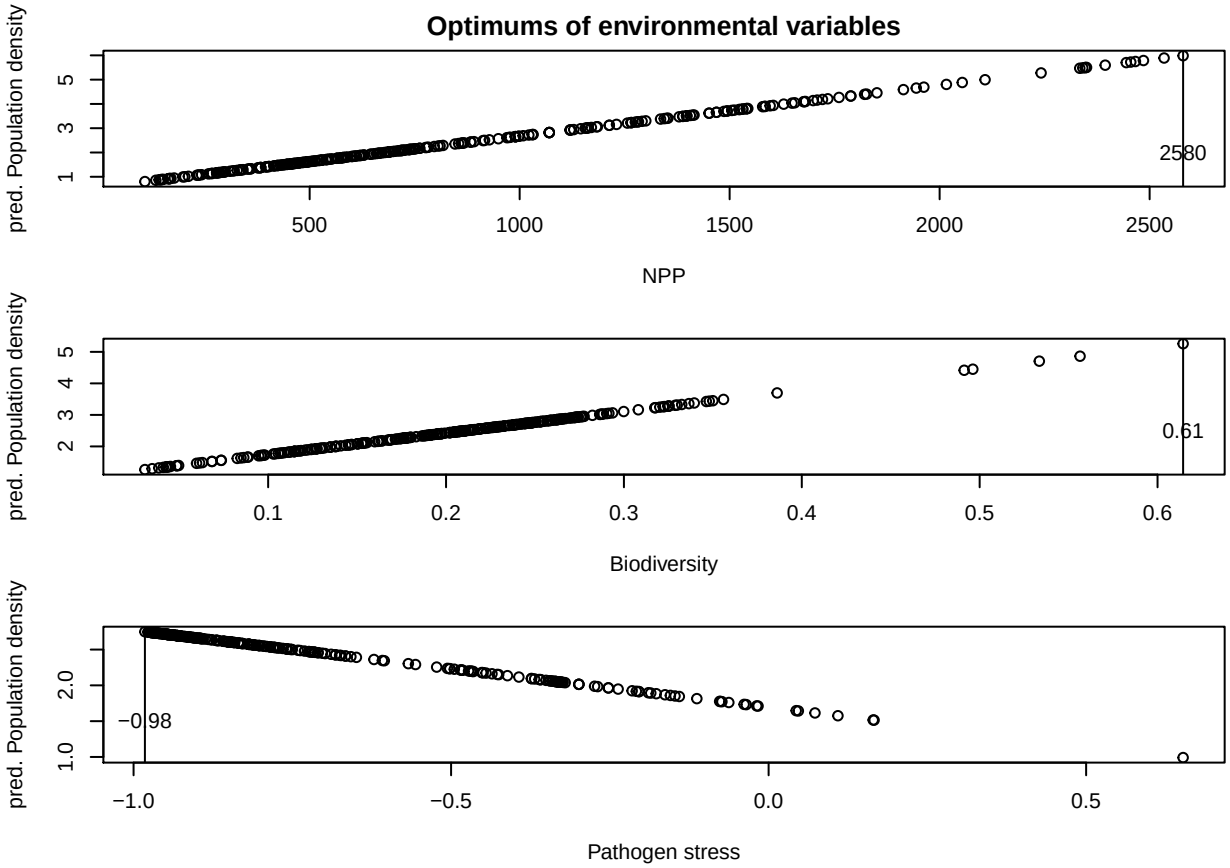
4.3.3 Analysis of limiting factors

To analyse, where NPP, biodiversity, and pathogen stress potentially have the strongest limiting effects on population density, we first determined optimum values of each predictor varibales. Next, we created three global datsets where where each of the predictor variables was set to its optimum, while other two variables remained untouched. Then, we produced three global projections of population densities using these predictor datasets. This resulted in three different predictions of population density. Finally, we subtracted original "true" population density predictions (see above) from these "optimum" predictions. For example, the larger the difference between prediction, where NPP was set to its optimum value, and the original prediction, the larger the limiting effect of NPP. Absolute differences (some were negative) between "optimum" and original predictions were scaled to vary between 0 and 1. For the three-band RGB color map these values were multiplied by 255.

4.3.3.1 Finding optimums of predictors

```
# Finding optimums of predictor variables
# First, create a dataset where the target variable (the one whose optimum
# we are interested in) is allowed to vary and other two variables are set
# to their mean.

dNPPopt <- data[,c("ldensityC","npp")]
dNPPopt$biodiv <- mean(data$biodiv)
dNPPopt$pathos <- mean(data$pathos)
row.names(dNPPopt) <- NULL
```

```r
dBIOopt <- data[,c("ldensityC","biodiv")]
dBIOopt$npp <- mean(data$npp)
dBIOopt$pathos <- mean(data$pathos)
row.names(dBIOopt) <- NULL

dPATopt <- data[,c("ldensityC","pathos")]
dPATopt$npp <- mean(data$npp)
dPATopt$biodiv <- mean(data$biodiv)
row.names(dPATopt) <- NULL

# Next, predict population density to these data.
# NPP optimum

svm.prds.NPPopt <- predict(svm.mod, type = "response", newdata = dNPPopt)
lm.prds.NPPopt <- predict(lm.mod, newdata = dNPPopt)

prds.NPPopt <- rowMeans(cbind(svm.prds.NPPopt, lm.prds.NPPopt))

# Biodiversity optimum
svm.prds.BIOopt <- predict(svm.mod, type = "response", newdata = dBIOopt)
lm.prds.BIOopt <- predict(lm.mod, newdata = dBIOopt)
prds.BIOopt <- rowMeans(cbind(svm.prds.BIOopt, lm.prds.BIOopt))

# Pathogen stress optimum
svm.prds.PATopt <- predict(svm.mod, type = "response", newdata = dPATopt)
lm.prds.PATopt <- predict(lm.mod, newdata = dPATopt)
prds.PATopt <- rowMeans(cbind(svm.prds.PATopt, lm.prds.PATopt))

# Plot optimums

par(mfrow=c(3,1), mar=c(4,4,2,1))
plot(data$npp,prds.NPPopt, xlab="NPP", ylab="pred. Population density",
     main="Optimums of environmental variables")
abline(v=data$npp[which.max(prds.NPPopt)])
text(data$npp[which.max(prds.NPPopt)],2,round(data$npp[which.max(prds.NPPopt)],0))

plot(data$biodiv,prds.BIOopt, xlab="Biodiversity", ylab="pred. Population density")
abline(v=data$biodiv[which.max(prds.BIOopt)])
text(data$biodiv[which.max(prds.BIOopt)],2.5,
     round(data$biodiv[which.max(prds.BIOopt)],2))

plot(data$pathos,prds.PATopt, xlab="Pathogen stress", ylab="pred. Population density")
abline(v=data$pathos[which.max(prds.PATopt)])
text(data$pathos[which.max(prds.PATopt)],1.5,
      round(data$pathos[which.max(prds.PATopt)],2))
```



4.3.3.2 Limiting effect of NPP in the global dataset.

```
# First, create a global dataset based on global predictor data that.
# This is done along the same lines as the original global data.

glb.data.opti.npp <- as.data.frame(predictors, xy=TRUE)
names(glb.data.opti.npp) <- c("x","y","npp","biodiv","pathos")

# Subset NA:s as a separate dataset.

glb.npp.opti.NA <- as.data.frame(glb.data.opti.npp[!complete.cases(glb.data.opti.npp),])
row.names(glb.npp.opti.NA)   <- NULL

# Add three "empty" variables.
glb.npp.opti.NA$ldensitylm  <- NA
glb.npp.opti.NA$ldensitysvm  <- NA


# Subset all the non-NA:s as a separate dataset.

glb.data2.opti.npp <- as.data.frame(glb.data.opti.npp[complete.cases(glb.data.opti.npp),])
row.names(glb.data2.opti.npp) <- NULL

# Set NPP values to its optimum
glb.data2.opti.npp$npp <- data$npp[which.max(prds.NPPopt)]
```

```
# Predict h-g population density using the fitted model algorithms (see above)

glb.data2.opti.npp$ldensitylm <- predict(lm.mod, newdata=glb.data2.opti.npp)

glb.data2.opti.npp$ldensitysvm <- predict(svm.mod, type = "response",
                                          newdata=glb.data2.opti.npp)


# Combine predictions and NA:s

glb.preds.npp.opti <- rbind(glb.data2.opti.npp,glb.npp.opti.NA)

# Calculate ensemble projection (prediction)

glb.preds.npp.opti$lensemble <- apply(glb.preds.npp.opti[,6:7], 1, mean)

# Difference between "real" and optimised

npp.effectL <- glb.preds.npp.opti$lensemble-global.preds$lensemble

# Absolute differences
npp.effectLABS <- abs(npp.effectL)
```

4.3.3.3 Limiting effect of biodiversity in the global dataset.

```
glb.data.opti.bio <- as.data.frame(predictors, xy=TRUE)
names(glb.data.opti.bio) <- c("x","y","npp","biodiv","pathos")

glb.bio.opti.NA <- as.data.frame(glb.data.opti.bio[!complete.cases(glb.data.opti.bio),])
row.names(glb.bio.opti.NA)   <- NULL

glb.bio.opti.NA$ldensitylm  <- NA
glb.bio.opti.NA$ldensitysvm  <- NA

glb.data2.opti.bio <- as.data.frame(glb.data.opti.bio[complete.cases(glb.data.opti.bio),])
row.names(glb.data2.opti.bio) <- NULL

glb.data2.opti.bio$biodiv <- data$biodiv[which.max(prds.BIOopt)]

glb.data2.opti.bio$ldensitylm <- predict(lm.mod, newdata=glb.data2.opti.bio)
glb.data2.opti.bio$ldensitysvm <- predict(svm.mod, type = "response",
                                          newdata=glb.data2.opti.bio)

glb.preds.bio.opti <- rbind(glb.data2.opti.bio,glb.bio.opti.NA)

glb.preds.bio.opti$lensemble <- apply(glb.preds.bio.opti[,6:7], 1, mean)

bio.effectL <- glb.preds.bio.opti$lensemble-global.preds$lensemble
bio.effectLABS <- abs(bio.effectL)
```

4.3.3.4 Limiting effect of pathogens in the global dataset.

```
glb.data.opti.pat <- as.data.frame(predictors, xy=TRUE)
names(glb.data.opti.pat) <- c("x","y","npp","biodiv","pathos")

glb.pat.opti.NA <- as.data.frame(glb.data.opti.pat[!complete.cases(glb.data.opti.pat),])
row.names(glb.pat.opti.NA)   <- NULL

glb.pat.opti.NA$ldensitylm  <- NA
glb.pat.opti.NA$ldensitysvm  <- NA

glb.data2.opti.pat <- as.data.frame(glb.data.opti.pat[complete.cases(glb.data.opti.pat),])
row.names(glb.data2.opti.pat) <- NULL

glb.data2.opti.pat$pathos <- data$pathos[which.max(prds.PATopt)]

glb.data2.opti.pat$ldensitylm <- predict(lm.mod, newdata=glb.data2.opti.pat)

glb.data2.opti.pat$ldensitysvm <- predict(svm.mod, type = "response",
                                          newdata=glb.data2.opti.pat)

glb.preds.pat.opti <- rbind(glb.data2.opti.pat,glb.pat.opti.NA)

glb.preds.pat.opti$lensemble <- apply(glb.preds.pat.opti[,6:7], 1, mean)


pat.effectL <- glb.preds.pat.opti$lensemble-global.preds$lensemble
pat.effectLABS <- abs(pat.effectL)
```

4.3.3.5 Scaling and plotting of the limiting effects

We scaled absolute differences and considered these as measures of the strength of the limiting effect of given environmental variable. Then we created global raster layers of each limiting effects. Finally we plotted the geographical distribution of the limiting effects using `plotRGB` function in `raster` *R* package.

```
# Scaling of absolute differences.
# First, a matrix of limiting effects of NPP, biodiversity, and pathogen stress

effects.mat <- as.data.frame(cbind(npp.effectLABS, bio.effectLABS, pat.effectLABS))

# Scaling function that scales absolute differences between 0 and 1.
standardRow <- function(z) {
rowmax <- apply(z, 1, max) # row max
rv <- sweep(z, 1, rowmax,"/")  # dividing by row maximum
return(rv)
}

effects.mat.strdR <- standardRow(effects.mat)
# Scale to RGB values
effects.mat.strdR.rgb <- effects.mat.strdR*255



# Create rasters of limiting effects
effects.strdR.npp.rast <- rasterFromXYZ(cbind(glb.preds.npp.opti[,c("x","y")],
```

```
                                                effects.mat.strdR.rgb[[1]]))
effects.strdR.bio.rast <- rasterFromXYZ(cbind(glb.preds.bio.opti[,c("x","y")],
                                                effects.mat.strdR.rgb[[2]]))
effects.strdR.pat.rast <- rasterFromXYZ(cbind(glb.preds.pat.opti[,c("x","y")],
                                                effects.mat.strdR.rgb[[3]]))


# Plot the geographical distribution of limiting effects
# Function for for legend
# Coordinates of the triangle
tri <- rbind(sin(0:2*2/3*pi), cos(0:2*2/3*pi))

# Function for calculating the color of a set of points `pt`
# in relation to the triangle
tricol <- function(pt, sharpness=2){
    require(splancs)
    RGB <- sapply(1:3, function(i){
        a <- sweep(pt, 2, tri[,i])
        b <- apply(tri[,-i], 1, mean) - tri[,i]
        sharpness*((a %*% b) / sum(b^2))-sharpness+1
    })
    alpha <- rep(1, nrow(pt))
    RGB <- cbind(RGB, alpha)
    RGB[-inpip(pt,t(tri)),] <- 0    # Color points outside the triangle white
    do.call(rgb, unname(as.data.frame(pmin(pmax(RGB, 0), 1))))
}


par(mar=c(0,0,0,0), mfrow=c(1,1))
image(lpdensity, col=col, breaks=breaks, asp=1, main="",
xlab="",ylab="", axes=F, xlim=c(-180,180))
plotRGB(stack(effects.strdR.npp.rast, effects.strdR.bio.rast, effects.strdR.pat.rast),
        r=3, g=1, b=2, stretch="lin", bgalpha=0, add=T)
text(0,90, "Geographical distribution of limiting effects")
plot(glaciers,col="white",border="white", add=T)
# Plot legend
res <- 1000                         # Resolution
xi <- seq(-1, 1, length=res)        # Axis points
yi <- seq(-.8, 1.2, length=res)
x <- xi[1] + cumsum(diff(xi))       # Midpoints between axis points
y <- yi[1] + cumsum(diff(yi))
xy <- matrix(1:(length(x)*length(y)), length(x))
par(fig=c(0.05,0.25,0.30,0.5), new=T)
image(xi, yi, xy, col=tricol(as.matrix(expand.grid(x,y)),sharpness=1.6),useRaster=T,
      xlab="",ylab="", axes=F)
```

```
## Loading required package: splancs
```

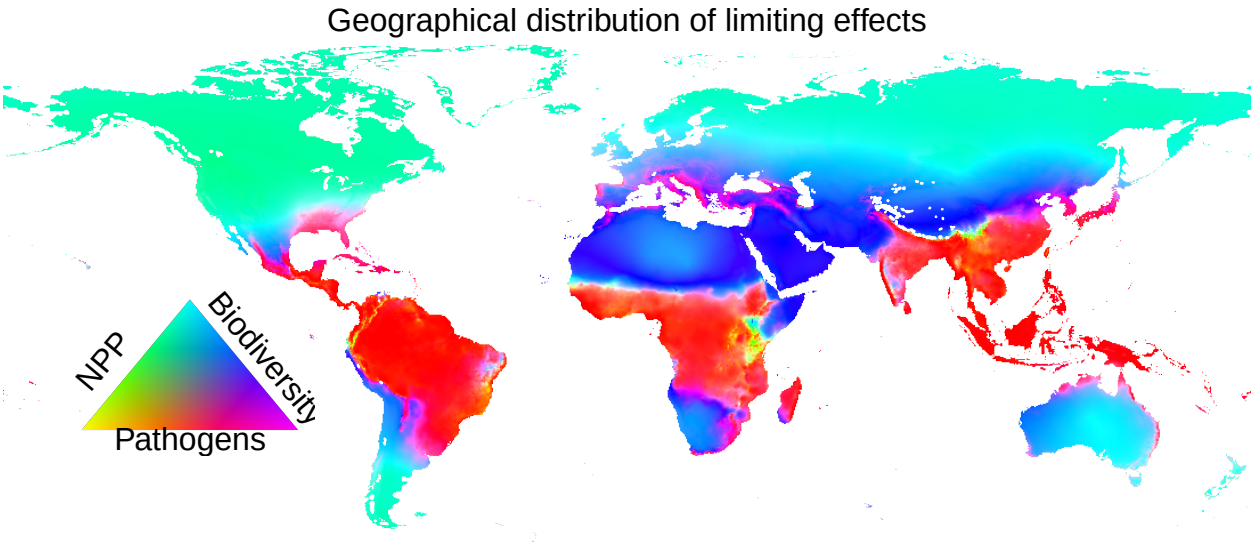
```
##
## Spatial Point Pattern Analysis Code in S-Plus
##
##  Version 2 - Spatial and Space-Time analysis
```

```
##
## Attaching package: 'splancs'
```

```
## The following object is masked from 'package:ape':
##
##     zoom
```

```
## The following object is masked from 'package:raster':
##
##     zoom
```

```
text(0.0,-0.65, "Pathogens")
text(0.6,0.3, "Biodiversity", srt=-50)
text(-0.7,0.3, "NPP", srt=50)
```



4.3.4 Biome-wise analyses

We also wanted to explore biome-wise differences in the predicted hunter-gatherer population density and limiting effects of environmental variables. Information on biomes is extracted from the FAO Global Ecological Zones data (http://www.fao.org/docrep/017/ap861e/ap861e00.pdf). We scaled limiting effects of different predictors to sum to unity (for illustrative purposes), stacked them with ecological zones and population density data. Finally, we calculated mean and maximum values of $\log_e$population density and mean values limiting effects in different biomes and plotted these as boxplots.

```
# Download biome data.
URL <- paste("http://www.fao.org/geonetwork/srv/en/",
             "resources.get?id=47105&fname=gez2010.zip&access=private", sep="")

wd <- getwd()
td <- tempdir()
```

```
setwd(td)
temp <- tempfile(fileext = ".zip")
download.file(URL, temp, mode="wb")
unzip(temp)
shp <- dir(tempdir(), "*.shp$")
lyr <- sub(".shp$", "", shp)
ecozon <- readOGR(dsn=shp, layer=lyr)
unlink(dir(td))
setwd(wd)


# Create a raster mask to rasterize polygon data ##
worldRaster <- nppRast
worldRaster[!is.na(worldRaster)] <- 0

# and rasterize
ecozonRast <- rasterize(ecozon, worldRaster)
```

```
# Scale absolute differences so that they sum to unity
unityRow <- function(z) {
rowsum <- apply(z, 1, sum) # row sum
rv <- sweep(z, 1, rowsum,"/")  # dividing by row sum
return(rv)
}

effects.mat.unity <- unityRow(effects.mat)

effects.unity.npp.rast <- rasterFromXYZ(cbind(glb.preds.npp.opti[,c("x","y")],
                                              effects.mat.unity[[1]]))
effects.unity.bio.rast <- rasterFromXYZ(cbind(glb.preds.bio.opti[,c("x","y")],
                                              effects.mat.unity[[2]]))
effects.unity.pat.rast <- rasterFromXYZ(cbind(glb.preds.pat.opti[,c("x","y")],
                                              effects.mat.unity[[3]]))



# Stack global ensemble projection with the eco-zones raster
# Fix projections.
crs(ecozonRast) <- "+proj=longlat +datum=WGS84"
crs(lpdensity) <- "+proj=longlat +datum=WGS84"
crs(effects.strdR.npp.rast) <- "+proj=longlat +datum=WGS84"
crs(effects.strdR.bio.rast) <- "+proj=longlat +datum=WGS84"
crs(effects.strdR.pat.rast) <- "+proj=longlat +datum=WGS84"
crs(effects.unity.npp.rast) <- "+proj=longlat +datum=WGS84"
crs(effects.unity.bio.rast) <- "+proj=longlat +datum=WGS84"
crs(effects.unity.pat.rast) <- "+proj=longlat +datum=WGS84"


ecozStack <- stack(ecozonRast, lpdensity, effects.unity.npp.rast,
                   effects.unity.bio.rast, effects.unity.pat.rast)


# Make a data frame.
ecozData <- as.data.frame(ecozStack, xy=T)
```

```
# Remove NA:s and eco-zone "Water"

names(ecozData) <- c("x","y","zone","zonecode","zonabr", "ldensity",
                     "nppEffectL", "bioEffectL", "patEffectL")
ecozData <- droplevels(subset(ecozData, subset=zone!="Water"))
ecozData <- ecozData[complete.cases(ecozData),]


# Add new variable to use in the plotting (for ordering the zones)

ecozData$zcode <- NA
ecozData$zcode[ecozData$zone=="Polar"] <- 1
ecozData$zcode[ecozData$zone=="Boreal tundra woodland"] <- 2
ecozData$zcode[ecozData$zone=="Boreal coniferous forest"] <- 4
ecozData$zcode[ecozData$zone=="Boreal mountain system"] <- 3
ecozData$zcode[ecozData$zone=="Temperate desert"] <- 5
ecozData$zcode[ecozData$zone=="Temperate steppe"] <- 6
ecozData$zcode[ecozData$zone=="Temperate continental forest"] <- 8
ecozData$zcode[ecozData$zone=="Temperate oceanic forest"] <- 9
ecozData$zcode[ecozData$zone=="Temperate mountain system"] <- 7
ecozData$zcode[ecozData$zone=="Subtropical desert"] <- 10
ecozData$zcode[ecozData$zone=="Subtropical steppe"] <- 11
ecozData$zcode[ecozData$zone=="Subtropical dry forest"] <- 13
ecozData$zcode[ecozData$zone=="Subtropical humid forest"] <- 14
ecozData$zcode[ecozData$zone=="Subtropical mountain system"] <- 12
ecozData$zcode[ecozData$zone=="Tropical desert"] <- 15
ecozData$zcode[ecozData$zone=="Tropical shrubland"] <- 16
ecozData$zcode[ecozData$zone=="Tropical dry forest"] <- 18
ecozData$zcode[ecozData$zone=="Tropical moist forest"] <- 19
ecozData$zcode[ecozData$zone=="Tropical rainforest"] <- 20
ecozData$zcode[ecozData$zone=="Tropical mountain system"] <- 17

ecozData$zcode <- as.factor(ecozData$zcode)
```

```
# Calculate mean and max population densities per eco zone

means <- aggregate(ldensity ~  zcode, ecozData, mean)
row.names(means) <- unique(ecozData$zone[order(ecozData$zcode)])

# Mean population density
means[order(means$ldensity, decreasing=TRUE),]
```

```
##                             zcode   ldensity
## Temperate oceanic forest        9  2.5465440
## Subtropical humid forest       14  2.4465594
## Temperate continental forest    8  2.3910739
## Tropical rainforest            20  2.3694524
## Subtropical dry forest         13  2.2473843
## Tropical dry forest            18  2.0963834
## Tropical moist forest          19  2.0499230
## Subtropical mountain system    12  1.9936458
## Temperate mountain system       7  1.9117124
## Tropical mountain system       17  1.8793235
```

```
## Temperate steppe                 6  1.8128139
## Subtropical steppe              11  1.6393966
## Tropical shrubland              16  1.5578194
## Boreal coniferous forest         4  1.2779684
## Subtropical desert              10  0.9027423
## Temperate desert                 5  0.8718035
## Boreal mountain system           3  0.8685544
## Tropical desert                 15  0.3342594
## Boreal tundra woodland           2  0.3081925
## Polar                            1 -0.6366659
```

```
maxs <- aggregate(ldensity ~  zcode, ecozData, max)
row.names(maxs) <- unique(ecozData$zone[order(ecozData$zcode)])

# Maximum population density
maxs[order(maxs$ldensity, decreasing=TRUE),]
```

```
##                              zcode ldensity
## Subtropical mountain system     12 5.122169
## Tropical mountain system        17 5.114679
## Temperate mountain system        7 5.072340
## Tropical rainforest             20 4.874803
## Temperate oceanic forest         9 4.596606
## Subtropical humid forest        14 4.432026
## Tropical moist forest           19 4.429815
## Tropical dry forest             18 4.356249
## Temperate continental forest     8 4.227278
## Subtropical dry forest          13 4.226206
## Tropical shrubland              16 4.025632
## Tropical desert                 15 3.988669
## Temperate steppe                 6 3.783219
## Subtropical steppe              11 3.663094
## Temperate desert                 5 3.255535
## Boreal mountain system           3 2.980866
## Subtropical desert              10 2.957178
## Polar                            1 2.771169
## Boreal coniferous forest         4 2.722453
## Boreal tundra woodland           2 2.235171
```

```
# Mean limiting effect of different predictors

meansNPPEF <- aggregate(nppEffectL ~  zcode, ecozData, mean)
row.names(meansNPPEF) <- unique(ecozData$zone[order(ecozData$zcode)])

# Mean limiting effect of NPP
meansNPPEF[order(meansNPPEF$nppEffectL, decreasing=TRUE),]
```

```
##                              zcode nppEffectL
## Boreal tundra woodland           2 0.58014071
## Polar                            1 0.57192078
## Boreal mountain system           3 0.54626769
## Boreal coniferous forest         4 0.54489621
## Temperate steppe                 6 0.46316419
## Temperate desert                 5 0.39509684
## Temperate continental forest     8 0.37692835
```

```
## Temperate oceanic forest         9 0.35844608
## Temperate mountain system        7 0.34877390
## Subtropical desert              10 0.34044197
## Subtropical steppe              11 0.31175355
## Tropical shrubland              16 0.24251452
## Tropical mountain system        17 0.20811590
## Subtropical mountain system     12 0.19695468
## Tropical desert                 15 0.19301416
## Tropical dry forest             18 0.18647440
## Subtropical humid forest        14 0.18031390
## Subtropical dry forest          13 0.16337810
## Tropical moist forest           19 0.13217152
## Tropical rainforest             20 0.06853341
```

```
meansBIOEF <- aggregate(bioEffectL ~  zcode, ecozData, mean)
row.names(meansBIOEF) <- unique(ecozData$zone[order(ecozData$zcode)])

# Mean limiting effect of biodiversity
meansBIOEF[order(meansBIOEF$bioEffectL, decreasing=TRUE),]
```

```
##                              zcode bioEffectL
## Tropical desert                 15  0.7000346
## Subtropical desert              10  0.6169294
## Subtropical steppe              11  0.5867234
## Temperate desert                 5  0.5669007
## Temperate mountain system        7  0.5463824
## Subtropical mountain system     12  0.5442797
## Tropical shrubland              16  0.5403865
## Subtropical dry forest          13  0.5324312
## Temperate steppe                 6  0.4894041
## Temperate continental forest     8  0.4662056
## Temperate oceanic forest         9  0.4421810
## Boreal mountain system           3  0.4420463
## Boreal coniferous forest         4  0.4378011
## Polar                            1  0.4222829
## Boreal tundra woodland           2  0.4148786
## Tropical dry forest             18  0.3185389
## Tropical mountain system        17  0.2869096
## Subtropical humid forest        14  0.2594102
## Tropical moist forest           19  0.1552999
## Tropical rainforest             20  0.0517441
```

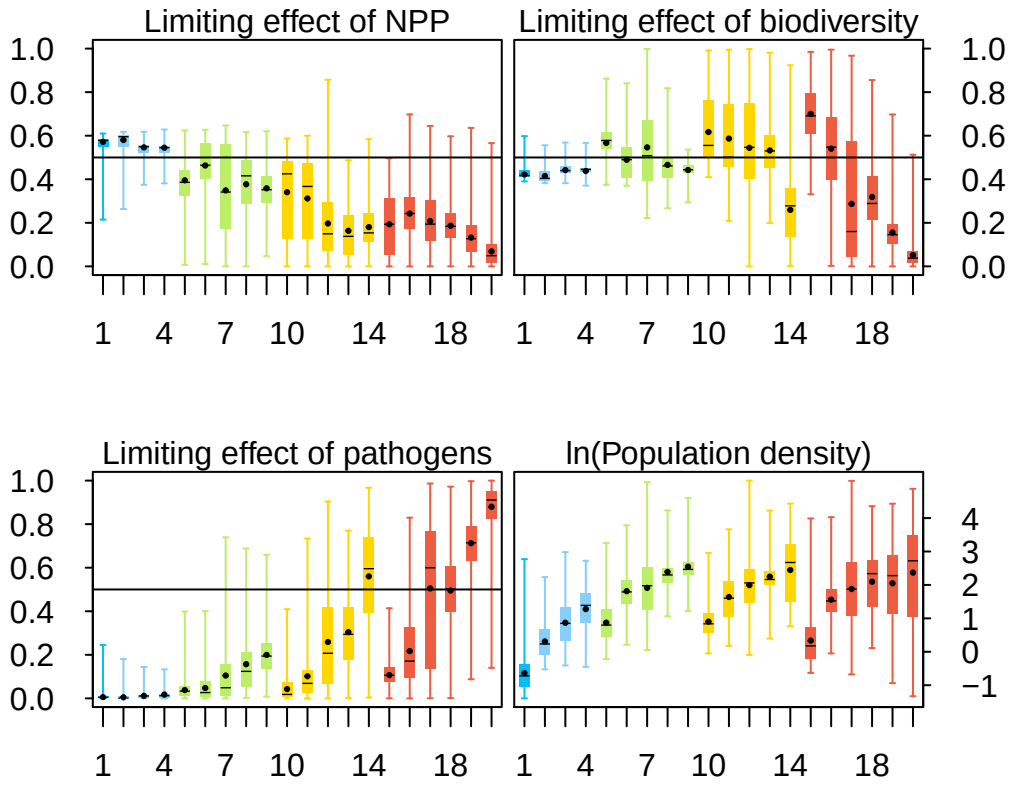
```
meansPATEF <- aggregate(patEffectL ~  zcode, ecozData, mean)
row.names(meansPATEF) <- unique(ecozData$zone[order(ecozData$zcode)])

# Mean limiting effect of pathogens
meansPATEF[order(meansPATEF$patEffectL, decreasing=TRUE),]
```

```
##                              zcode  patEffectL
## Tropical rainforest             20 0.879722497
## Tropical moist forest           19 0.712528621
## Subtropical humid forest        14 0.560275888
## Tropical mountain system        17 0.504974485
## Tropical dry forest             18 0.494986728
## Subtropical dry forest          13 0.304190680
```

```
## Subtropical mountain system     12 0.258765584
## Tropical shrubland              16 0.217098977
## Temperate oceanic forest         9 0.199372891
## Temperate continental forest     8 0.156866044
## Tropical desert                 15 0.106951206
## Temperate mountain system        7 0.104843698
## Subtropical steppe              11 0.101523066
## Temperate steppe                 6 0.047431675
## Subtropical desert              10 0.042628610
## Temperate desert                 5 0.038002421
## Boreal coniferous forest         4 0.017302678
## Boreal mountain system           3 0.011685973
## Polar                            1 0.005796347
## Boreal tundra woodland           2 0.004980677
```

```
# Make a box-plot (see above tables for eco zone codes (zcode))

cols=c("deepskyblue",rep("lightskyblue",3), rep("darkolivegreen2",5),
       rep("gold",5), rep("tomato2",6))

par(xaxs="i")
par(fig=c(0.075,0.495,0.52,0.97))
par(mar=c(2,3,2,0))
boxplot(nppEffectL~zcode, data=ecozData, range=0,boxwex=0.5, col=cols,boxlty=0,
        ylim=c(0,1),medcol="black", whisklty=1,whiskcol=cols,staplecol=cols,
        medlwd=0.7, yaxt="n")
points(1:20, meansNPPEF$nppEffectL, col = "black", pch=19, cex=0.30)
axis(side=2, at=c(0,0.2,0.4,0.6,0.8,1), las=2, tck=-0.03)
abline(h=0.5)
mtext(side=3, "Limiting effect of NPP")

par(fig=c(0.505,0.925,0.52,0.97), new=T)
par(mar=c(2,0,2,3))
boxplot(bioEffectL~zcode, data=ecozData, range=0,boxwex=0.5, col=cols,boxlty=0,
        ylim=c(0,1), medcol="black", whisklty=1,whiskcol=cols,staplecol=cols,
        medlwd=0.7, yaxt="n")
points(1:20, meansBIOEF$bioEffectL, col = "black", pch=19, cex=0.30)
axis(side=4, at=c(0,0.2,0.4,0.6,0.8,1), las=2, tck=-0.03)
mtext(side=3, "Limiting effect of biodiversity")
abline(h=0.5)

par(fig=c(0.075,0.495,0.02,0.47), new=T)
par(mar=c(2,3,2,0))
boxplot(patEffectL~zcode, data=ecozData, range=0,boxwex=0.5, col=cols,boxlty=0,
        ylim=c(0,1), medcol="black", whisklty=1,whiskcol=cols,staplecol=cols,
        medlwd=0.7, yaxt="n")
points(1:20, meansPATEF$patEffectL, col = "black", pch=19, cex=0.30)
axis(side=2, at=c(0,0.2,0.4,0.6,0.8,1), las=2, tck=-0.03)
mtext(side=3, "Limiting effect of pathogens")
abline(h=0.5)

par(fig=c(0.505,0.925,0.02,0.47), new=T)
par(mar=c(2,0,2,3))
boxplot(ldensity~zcode, data=ecozData, range=0,boxwex=0.5, col=cols,boxlty=0,
```

```
medcol="black", whisklty=1,whiskcol=cols,staplecol=cols,medlwd=0.7, yaxt="n")
points(1:20, means$ldensity, col = "black", pch=19, cex=0.30)
axis(side=4, at=c(-2,-1,0,1,2,3,4), las=2, tck=-0.03)
mtext(side=3, "ln(Population density)")
```



References

Binford, Lewis R. 2001. *Constructing Frames of Reference: An Analytical Method for Archaeological Theory Building Using Ethnographic and Environmental Data Sets*. Berkeley: University of California Press.

Cashdan, Elizabeth. 2014. "Biogeography of Human Infectious Diseases: A Global Historical Analysis." Edited by Linda Anne Selvey. *PLoS ONE* 9 (10): e106752. doi:10.1371/journal.pone.0106752.

Ellis, Erle C., Erica C. Antill, and Holger Kreft. 2012. "All Is Not Loss: Plant Biodiversity in the Anthropocene." *PLOS ONE* 7 (1): e30535. doi:10.1371/journal.pone.0030535.

Kelly, Robert L. 2013. *The Lifeways of Hunter-Gatherers: The Foraging Spectrum*. 2 edition. Cambridge: Cambridge University Press.

Kreft, Holger, and Walter Jetz. 2007. "Global Patterns and Determinants of Vascular Plant Diversity." *Proceedings of the National Academy of Sciences* 104 (14): 5925–30. doi:10.1073/pnas.0608361104.

Lieth, Helmut. 1973. "Primary Production: Terrestrial Ecosystems." *Human Ecology* 1 (4): 303–32.

doi:10.1007/BF01536729.

Low, Bobbi S. 1994. "Pathogen Intensity Cross-Culturally." *World Cultures* 8 (1): 24–34.

Salonen, J Sakari, Adrian J Verster, Stefan Engels, Janne Soininen, Mathias Trachsel, and Miska Luoto. 2016. "Calibrating Aquatic Microfossil Proxies with Regression-Tree Ensembles: Cross-Validation with Modern Chironomid and Diatom Data." *The Holocene* 26 (7): 1040–8. doi:10.1177/0959683616632881.