

```

library(raster)
library(dismo)
library(ROCR)
library(sp)
work <- getwd()

BNG <- CRS("+init=epsg:27700")

models <- 30
iterations <- 50
p.trans <- 0.25
n.points <- 20

f.prob <- function(x, p.trans){
  x*p.trans
}

f.bin <- function(x){
  if(!is.na(x)) x <- rbinom(1,1,x)
  else x <- NA
}

f.correct <- function(x){
  if(!is.na(x) && x > 1) x <- 1
  else x
}

### Function to grow new habitat

f.Exp <- function(map, base.map, p.trans){
  mat.0 <- as.matrix(map)
  map.t <- focal(map, w=matrix(1, nr=3, nc=3),sum, na.rm=T, pad=T)
  map.t <- map.t + base.map
  mat.t <- as.matrix(map.t)
  mat.prob <- apply(mat.t, c(1,2), f.prob, p.trans=p.trans)
  mat.cor <- mat.0 + mat.prob
  mat.cor <- apply(mat.cor, c(1,2), f.correct)
  mat.new <- apply(mat.cor, c(1,2), f.bin)
  map.new <- raster(mat.new, xmn=xmin(map), xmx=xmax(map), ymn=ymin(map), ymx=ymax(map),
crs=paste(projection(map)))
}

# Model of black grouse habitat suitability in 1994

BK.pres <- read.csv("BK_pres.csv")

env.l.base <- stack(raster(paste(getwd(), "Proportion/hab_1.asc", sep="/"), crs=BNG), raster(paste
(getwd(), "Proportion/hab_2.asc", sep="/"), crs=BNG), raster(paste(getwd(), "Proportion/hab_3.asc",
sep="/"), crs=BNG), raster(paste(getwd(), "Proportion/hab_4.asc", sep="/"), crs=BNG), raster(paste
(getwd(), "Proportion/hab_5.asc", sep="/"), crs=BNG), raster(paste(getwd(), "Proportion/hab_6.asc",
sep="/"), crs=BNG), raster("studyareadem.asc", crs=BNG))

max_base <- maxent(env.l.base, BK.pres)

base.map <- predict(max_base, env.l.base)

writeRaster(base.map, filename=(paste(getwd(), "base_map.grd", sep="/")), overwrite=T)

BK.sp <- SpatialPoints(BK.pres, proj4string=BNG)
BK.scores <- extract(base.map, BK.sp)
cutoff.1 <- summary(BK.scores)[[2]]
cutoff.med <- summary(BK.scores)[[3]]
cutoff.3 <- summary(BK.scores)[[5]]

#bg <- randomPoints(env.l.base, 1000)
#pres.pp <- extract(base.map, BK.pres)
#bg.pp <- extract(base.map, bg)
#combined <- c(pres.pp, bg.pp)
#label <- c(rep(1,length(pres.pp)),rep(0,length(bg.pp)))

#pred <- prediction(combined, label)
#perf <- performance(pred, "tpr", "fpr")

```

```
#fpr <- perf@x.values[[1]]
#tpr <- perf@y.values[[1]]
#sum.v <- tpr + (1-fpr)
#index <- which.max(sum.v)
#cutoff <- perf@alpha.values[[1]][[index]]

rcl.1 <- matrix(c(0, cutoff.1-0.0001, 0, cutoff.1, 1, 1), nrow=3, ncol=3, byrow=T)
rcl.med <- matrix(c(0, cutoff.med-0.0001, 0, cutoff.med, 1, 1), nrow=3, ncol=3, byrow=T)
rcl.3 <- matrix(c(0, cutoff.3-0.0001, 0, cutoff.3, 1, 1), nrow=3, ncol=3, byrow=T)

pa.1 <- reclass(base.map, rcl.1)
pa.med <- reclass(base.map, rcl.med)
pa.3 <- reclass(base.map, rcl.3)
rm(base.map)
max_scores <- data.frame("Model"=c("1st", "Median", "3rd"), "Proportion"=numeric(3), "Cutoff"=numeric
(3))
max_scores[1,3] <- cutoff.1
max_scores[2,3] <- cutoff.med
max_scores[3,3] <- cutoff.3
max_scores[1,2] <- cellStats(pa.1, mean)
max_scores[2,2] <- cellStats(pa.med, mean)
max_scores[3,2] <- cellStats(pa.3, mean)
rm(pa)

write.csv(max_scores, "max_models.csv", row.names=F)

source("Sim_scenario_1.R", echo=T, verbose=T)
source("Sim_scenario_2.R", echo=T, verbose=T)
source("Sim_scenario_3.R", echo=T, verbose=T)
source("Sim_scenario_6.R", echo=T, verbose=T)
source("Sim_scenario_7.R", echo=T, verbose=T)
```