# with the presence/absence matrices per density, we can use the RN occupany

# model to examine whether it provides a significan\_intervals difference between

# pairs of simulations with differen\_intervals densities.

# We can do this with differen\_intervals parameter settings:

# - density-combinations

# - time in\_intervalservals

# - study durations

# - number of camera traps

# do this for 10 simulation runs, using 10 parallel sessions...

# TODO iteraties eruit halen en parameter combinaties inlezen uit file!

royle\_nichols\_stats <- function(sim\_nr) {

library(unmarked)

x <- vector(length = 0)

data <- data.frame(

dens1 = x,

dens2 = x,

time\_in\_intervalserval = x,

study\_duration = x,

n\_cams = x,

z = x,

P = x,

p\_presence = x)

outputfile <- paste("./data/processed/PresenceAbsence/RoyleNicholsStats",

sim\_nr,

".txt")

if (!file.exists(outputfile)) {

write.table(data, outputfile, col.names = TRUE, row.names = FALSE)

}

for (i1 in 1:18){

for (i2 in (i1 + 1):19){

study\_duration <- 365

n\_cams <- 25

print(paste(i1, i2))

df1 <- as.matrix(

read.table(paste("./data/processed/PresenceAbsence/PresAbs",

i1, sim\_nr, ".txt")))

dens1 <- df1[1, 1]

df1 <- df1[, 2:366]

df2 <- as.matrix(

read.table(paste("./data/processed/PresenceAbsence/PresAbs",

i2, sim\_nr, ".txt")))

dens2 <- df2[1, 1]

df2 <- df2[, 2:366]

cov <- data.frame(

location = factor(c(rep("A", 25), rep("B", 25))),

density = c(rep(dens1, 25), rep(dens2, 25))

)

dets <- rbind(df1, df2)

for (n\_cams in 25) { # 25:5)

for (study\_duration in 50) { # round(exp(log(365)\*(10:30/30))))

for (time\_in\_intervalserval in 1) { # unique(round(1.1^(4:54))))

if (time\_in\_intervalserval <= study\_duration / 2) {

sd <- study\_duration

n\_intervals <- floor(sd / time\_in\_intervalserval)

sd <- n\_intervals \* time\_in\_intervalserval

dets2 <- dets[, 1:sd]

dets3 <- matrix(0, 50, n\_intervals)

# voor iedere rij in dets2, presence/absence samenvoegen per

# tijdsintervalserval...

x <- sort(rep(1:n\_intervals, time\_in\_intervalserval))

for (i3 in 1:(n\_cams \* 2)) {

dets3[i3, ] <- tapply(dets2[i3, ], x, max)

}

# randomly select cameras to use...

cams\_used <- sort(rank(runif(25))[1:n\_cams])

# using n\_cams random camera traps.

dets3 <- dets3[c(cams\_used, cams\_used + 25), ]

cov2 <- cov[c(cams\_used, cams\_used + 25), ]

p\_presence <- mean(rowSums(dets3) > 0)

umf <- unmarkedFrameOccu(y = dets3, siteCovs = cov2)

m1 <- occuRN(~1 ~ location, umf)

s1 <- summary(m1)

data <- data.frame(

dens1 = dens1,

dens2 = dens2,

time\_in\_intervalserval = 1,

study\_duration = sd,

n\_cams = n\_cams,

z = s1$state$z[2],

P = s1$state$`P(>|z|)`[2],

p\_presence = p\_presence)

write.table(

data, outputfile,

append = TRUE,

col.names = FALSE,

row.names = FALSE)

}

}

}

}

}

}

}

# library needed for parallel runs:

library(parallel)

# create the local cluster:

cl <- makeCluster(10)

# run the simulations:

results <- parSapply(cl, 1:10, royle\_nichols\_stats)

# stop the cluster:

stopCluster(cl)