

Supplement 3. Code accompanying the article:

Exposing changing phenology of fish larvae by modeling climate effects on temporal early life-stage shifts

Benjamin Weigel*, Jussi Mäkinen, Meri Kallasvuo, Jarno Vanhatalo

Table of Contents

Building the HMSC models	2
Post-processing of model results	3
Build hurdle model output and process data for plots	4
Define thresholds and plot examples for figure 4 and 5 in the manuscript.....	4

This document contains the R script for performed analysis in Weigel et al. 2021
<https://doi.org/10.3354/meps13676>.

The herring larval data underlying the analysis are available upon request and a metadata summary with contact information is given at the *Research Data Descriptions Discovery Service* of the **Natural Resources Institute Finland (LUKE)**
<http://radar.luke.fi/catalog/search/resource/details.page?uuid=%7B7FB77F7B-87E3-4583-BBD6-5EDC21997573%7D>

```
#load required libraries
library(Hmsc)
library(BayesLogit)
library(coda)
library(plyr)
library(corrplot)
library(abind)
library(ggpubr)
library(viridis)
library(tidyr)
library(dplyr)

library(ggplot2)
```

We implement the the Ricker model for the *abundance* data (Yabu) as defined in the manuscript, applying multivariate generalized linear mixed models using HMSC. We use the abundance conditional on presence and *presence absence* (Ypa) data to fit separate models as parts of the Hurdle model framework described in the manuscript.

Building the HMSC models

```
#Defining the regression function, including second order polynomial response
s
#for chlorophyll_a and temperature

XFormula= ~ salt + poly(chl_t, degree=2, raw=TRUE) +
poly(temp, degree=2, raw=TRUE) + site:SSB

#Study design
studyDesign = matrix(NA,nrow(Ypa),2)
studyDesign[,1] = sprintf('%.2s',newRE$site) # Site
studyDesign[,2] = sprintf('%.10s',newRE$RL) # spatio temporal random level
studyDesign = as.data.frame(studyDesign)
colnames(studyDesign) = c("site","Year_day_site")
studyDesign[,1]=as.factor(studyDesign[,1])
studyDesign[,2]=as.factor(studyDesign[,2])

#Set random effects
rL1=HmscRandomLevel(units=unique(studyDesign$site))
rL2=HmscRandomLevel(units=unique(studyDesign$Year_day_site))

###
#Define presence absence model
mpa = Hmsc(Y = 1*(Ypa>0), XData = X, XFormula = XFormula, studyDesign = study
Design, ranLevels = list(site=rL1, Year_day_site=rL2), distr = "probit")

#Run HMSC for presence absence data
samples = 500
thin = 10
transient = .5*(samples*thin)
adaptNf = .4*(samples*thin)
nChains = 4

mpa = sampleMcmc(mpa, samples = samples, thin = thin,
transient = transient, adaptNf = rep(adaptNf, mpa$nr),
nChains = nChains, nParallel = nChains)
###

###
#Define abundance model
mabu = Hmsc(Y = Yabu, XData = X, XFormula = XFormula, studyDesign =
studyDesign, ranLevels = list(site=rL1,Year_day_site=rL2), distr = "normal")

#Run HMSC for abundance data
samples = 500
thin = 10
transient = .5*(samples*thin)
adaptNf = .4*(samples*thin)
nChains = 4
```

```
mabu = sampleMcmc(mabu, samples = samples, thin = thin,  
transient = transient, adaptNf = rep(adaptNf, mabu$nr),  
nChains = nChains, nParallel = nChains)  
###
```

Post-processing of model results

```
#make coda object for MCMC check  
postpa = convertToCodaObject(mpa) # (presence absence)  
postabu = convertToCodaObject(mabu) # (abundance)  
  
#examine MCMC and effective sample size  
summary(effectiveSize(postpa$Beta))  
summary(effectiveSize(postabu$Beta))  
gelman.diag(postpa$Beta)  
gelman.plot(postpa$Beta)  
gelman.diag(postabu$Beta)  
gelman.plot(postabu$Beta)  
  
#get beta parameter estimates  
betapa = getPostEstimate(mpa, "Beta")  
betaabu = getPostEstimate(mabu, "Beta")  
  
#compute predictions for Ypa and Yabu  
predYpa = computePredictedValues(mpa, expected = FALSE, thin = 1)  
predYabu = computePredictedValues(mabu, expected = TRUE, thin = 1)  
  
#Evaluate model fit  
MFpa = evaluateModelFit(hM = mpa, predY = predYpa)  
MFabu = evaluateModelFit(hM = mabu, predY = predYabu)  
  
# Variance Partitioning  
# Set groups  
group=c(1,1,2,2,3,3,4,4,4,4,4,4,4)  
groupnames = c("Salinity", "Chlorophyll a", "Temperature", "SSB at area")  
#Compute and plot VP  
VPabu = computeVariancePartitioning(hM = mabu, group = group,  
groupnames = groupnames)  
plotVariancePartitioning(mabu, VPabu, col=viridis(7))  
  
VPpa = computeVariancePartitioning(hM = mpa, group = group,  
groupnames = groupnames)  
plotVariancePartitioning(mpa, VPpa, col=viridis(7))
```

Build hurdle model output and process data for plots

```
# Combine predicted values from abundance (predYabu) and
# presence absence (predYpa) model

predYhurdle<- predYpa*(exp(predYabu))

# For loop through all site-day combinations
predFracHurde = matrix(nrow=nrow(predYhurdle),ncol=2000)
probOver10 = matrix(nrow=nrow(predYhurdle),ncol=1)
for (i in 1:nrow(predYhurdle)){
  x = predYhurdle[i,,]
  x2 =x[3,]/colSums(x) #large size class/all larvae
  predFracHurde[i,] = x2
  probOver10[i]<-sum(na.omit(predFracHurde[i,])>0.1)/
  length(na.omit(predFracHurde[i,]))
}

#predFracHurde is site specific propability distribution for the fraction
#of large larvae

#probOver10 is the probability for large larvae making up >10 % of the pool

#propmeans is mean fraction of large larvae across all MCMC samples

propmeans<-rowMeans(predFracHurde,na.rm = TRUE)

# For loop for probability of small larvae being there
probthere = matrix(nrow=nrow(predYpa),ncol=2000)
for (i in 1:nrow(predYpa)){
  y = predYpa[i,,]
  y2 =temp[1,]
  probthere[i,] = y2
}
probtheremean<-rowMeans(probthere,na.rm = TRUE)
```

Define thresholds and plot examples for figure 4 and 5 in the manuscript

```
#Fig 4. mean probability of small larvae being there with different threshold
s
###
threshold90<-data%>%
  dplyr::filter(probtheremean >=0.90)%>%
  dplyr::group_by(Year, site)%>%
  dplyr::slice(which.min(yday)) #yday is the julian day of the year

threshold75<-data%>%
  dplyr::filter(probtheremean >=0.75)%>%
```

```

dplyr::group_by(Year, site)%>%
dplyr::slice(which.min(yday))

area1<-filter(threshold90, site == 1)
area2<-filter(threshold90, site == 2)
area3<-filter(threshold90, site == 3)
area4<-filter(threshold90, site == 4)
area5<-filter(threshold90, site == 5)
area6<-filter(threshold90, site == 6)
area7<-filter(threshold90, site == 7)

area1_75<-filter(threshold75, site == 1)
area2_75<-filter(threshold75, site == 2)
area3_75<-filter(threshold75, site == 3)
area4_75<-filter(threshold75, site == 4)
area5_75<-filter(threshold75, site == 5)
area6_75<-filter(threshold75, site == 6)
area7_75<-filter(threshold75, site == 7)

####plot first day of small larvae present with mean prob. of 90% and 75%

ggplot(threshold90)+
  aes(x=Year, y=yday)+
  geom_point(data=threshold75, color="slategray4")+
  geom_point(color="black")+
  xlim(1974,1996)+
  ylim(121,210)+
  # geom_jitter(color="black", position=position_jitter(1,1))+
  theme_bw()+
  theme(axis.title = element_text(size = 16),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14))+
  labs(title = "All areas combined", y = "day of year")+
  stat_smooth(method=lm, color="black") +
  stat_smooth(data=threshold75,method=lm, color="slategray4") +
  theme(panel.grid.major = element_line(linetype = "blank"), panel.grid.minor
= element_line(linetype = "blank"))+
  stat_cor(label.x.npc = "middle", label.y.npc = "top")+
  stat_regline_equation(label.x.npc = "left", label.y.npc = "top")+
  stat_cor(data=threshold75,label.x.npc = "middle", label.y.npc = "bottom")+
  stat_regline_equation(data=threshold75,label.x.npc = "left",
label.y.npc = "bottom")

#Fig. 5: Define threshold: All days of the year where mean proportion of large
e
#larvae is at least 10% of total larvae pool

threshold<-data %>%

```

```
dplyr::filter(propmeans >= 0.1)%>%
  dplyr::group_by(Year, site)

threshold$site<-as.factor(threshold$site)
threshold$Year<-as.numeric(as.character(threshold$Year))

area1<-filter(threshold, site == 1)
area2<-filter(threshold, site == 2)
area3<-filter(threshold, site == 3)
area4<-filter(threshold, site == 4)
area5<-filter(threshold, site == 5)
area6<-filter(threshold, site == 6)
area7<-filter(threshold, site == 7)

#plot
ggplot(threshold)+
  aes(x=Year, y=yday, palette = c("black"))+
  geom_point(color="black", size =1)+
  ylim(150,250)+
  xlim(1974,1996)+
  geom_jitter(color="black",size=1, position=position_jitter(0.5,0.5))+
  theme_bw()+
  theme(axis.title = element_text(size = 16),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14))+
  labs(title = "All areas combined", y = "day of year")+
  stat_smooth(method=lm, color="red") +
  theme(panel.grid.major = element_line(linetype = "blank"), panel.grid.minor
= element_line(linetype = "blank"))+
  stat_cor(label.x.npc = "middle", label.y.npc = "top")+
  stat_regline_equation(label.x.npc = "left", label.y.npc = "top")
```