R code for implementation of predator-prey system models

Contents

[#A: Loading data phase 1](#_Toc116837748)

[#B: Loading essential libraries in R and initializing parameters phase 1](#_Toc116837749)

[#C: Choose among ecologic ode models phase 2](#_Toc116837750)

[#E: Run HMC method via R-STAN 2](#_Toc116837751)

[#E: Create posterior predictive plots 3](#_Toc116837752)

[#F: Calculate and record information criteria and elapsed time 5](#_Toc116837753)

# #A: Loading data phase

loc\_dir = "C:\\..." #local working directory that data files are located

setwd(loc\_dir) #set the working directory

# Chrysoperla-Myzus (C-M) system data:#

# read sort time data e.g. 2 hr in our data#

newdata <- read.xlsx(file = "Chrysopidae.xlsx",sheetName = "2hr")

#read long time data e.g. 24 hr in our data #

newdata <- read.xlsx(file = "Chrysopidae.xlsx",sheetName = "24hr")

# Propylea–Aphis (P-A) system data:#

newdata <- read.table("newdata.txt", header = T)

# #B: Loading essential libraries in R and initializing parameters phase

#Load R-libraries

library(ggplot2) # essential for plotting

library(xlsx) # access to excel files

library(rstan) # STAN library in R

#Markov Chains (MC) paramaters#

nch<-3 #number of chains

iter<-30000 #number of iterations

bin<-20000 #size of burn-in

nth<-1 #size of thinning

# Deterministic or Stochastic approach indicator#

type<-"stochastic” # or "deterministic"

#Model parameters initialization#

inum<-2 # number of predators as integer, e.g. 1,2, 3 etc.

N<-6 # hours prey exposed to predator

NN<- nrow(newdata) #data size derived from data rows number

Me<-1 #number of ord. diff. equations in the system as integer

n\_trials<- nrow(newdata) #number of trials

t0<-0 #initial time value

t <- 1:N #time vector

n\_pars<-3 #number of parameters in the ode equation

Pe <- as.array(inum) #number of predators in the model. Plug in as array constant

Ne <- as.integer(newdata[,2\*inum]) #Prey density at the end of the study as integer

A<- as.integer(newdata[,(2\*inum-1)]) #initial values of densities of prey as integer

parms<- c("theta","dev") #parameters of interest vector

#create input vector readable in R-STAN#

my\_data <-

list(t=t,n\_trials=n\_trials,n\_pars=n\_pars,Ne=Ne,A=A,t0=t0,M=M,N=N,NN=NN,pe=pe)

#create initial values using function for parameters of interest for R-STAN

brfun<- function (num\_id=1) {list(kappa=matrix(data = runif(NN\*2,0,1),2,NN),

theta=c(rexp(n\_pars,0.1)),phi=rgamma(1,0.1,0.1),s\_sq=1/rgamma(1,10,10))}

inits<-lapply(1:nch, function(id) brfun(num\_id=id))

# #C: Choose among ecologic ode models phase

####H2 MODEL####

if (type=="deterministic") cfile<-"hol2.stan" else cfile<-"hol\_ou\_twoinstant.stan"

model\_obj<- paste0("mod",inum,"\_hol")

####BD MODEL####

cfile<-"Beddington.stan"

model\_obj<- paste0("mod",inum,"\_BD")

####CM MODEL####

cfile<-"CrowelyMartin.stan.stan"

model\_obj<- paste0("mod",inum,"\_CM")

####HV MODEL####

cfile<-" Hassell.stan"

model\_obj<- paste0("mod",inum,"\_HV")

####SSS MODEL####

if(type=="deterministic") cfile<-"Sss.stan" else cfile<-"SS\_binom\_ou\_twoinstant.stan"

model\_obj<- paste0("mod",inum,"\_sss")

# #E: Run HMC method via R-STAN

#run R-STAN algorithm

assign(model\_obj ,

stan(file = cfile,

data = my\_data,

init = inits,

control = list(adapt\_delta = 0.9),

chains = nch,

warmup = bin,

iter = iter,

thin=nth))

#save and print summary statistics

result<- as.data.frame(summary(eval(parse(text= model\_obj)),pars=param)$summary)

print(result, d=3)

# #E: Create posterior predictive plots

# access posterior predictive estimates#

pnames<-c(parms,"y\_fit") #inlcude predictive values in parameters of interest

# summary statistics#

param.sample<-as.data.frame(summary(eval(parse(text=model\_obj)),pars=pnames)$summary)

# vector of predictive values#

yypred<-as.matrix(eval(parse(text= model\_obj)),pars="y\_fit")

xv<-newdata$Density # vector of predictor variable values

#data frames of 2.5, 25, 50, 75 and 97.5 percentiles of predictive values#

yquant1data<-data.frame(x = xv, y = apply(yypred,2, function(x) quantile(x, c(0.025)) ))

yquant3data<-data.frame(x = xv, y = apply(yypred,2, function(x) quantile(x, c(0.25)) ))

yquant4data<-data.frame(x = xv, y = apply(yypred,2, function(x) quantile(x, c(0.75)) ))

ymeddata<-data.frame(x = xv, y = apply(yypred,2, median))

yquant2data<-data.frame(x = xv, y = apply(yypred,2, function(x) quantile(x, c(0.975)) ))

#real data values dataframe#

ndata<-data.frame(x=newdata$Density.,y=newdata$Aphids..Eaten.)

#create new dataframe to include augmented results#

yndata<-ymeddata

yndata$Legend<-as.factor(yndata$x)# convert grouping variables to factor

yndata$y1 <- yquant1data$y # 2.5 percentile of predictive values

yndata$median <- ymeddata$y # 50 percentile of predictive values

yndata$y2 <- yquant2data$y # 97.5 percentile of predictive values

yndata$y3 <- yquant3data$y # 25 percentile of predictive values

yndata$y4 <- yquant4data$y # 75 percentile of predictive values

Legend<-factor(yndata$x) #the prey density levels

#define color of graph and other features#

#define color gradient for graph points and lines#

colval<-colorRampPalette(c("royalblue","red"))

#define color palette for graph grid lines#

scale\_color\_brewer(palette="Dark2")

#adjust text horizontal justification#

theme\_update(plot.title = element\_text(hjust = 0.5))

#center the output#

ggplot() + ggtitle("Default is now set to centered")

#plot structure#

if (N==2){

ggplot(data = yndata, aes(x = x, y = y)) +

geom\_point(data=ndata, size = 1.5, aes(x = x, y = y, colour = "#000000"),alpha=1) +

geom\_line(aes(x = x, y = median, colour="median"), size=1)+

geom\_ribbon(aes(x = x,ymin=y1,ymax=y2),alpha=0.25, fill="#81DAF5")+

geom\_ribbon(aes(x = x,ymin=y3,ymax=y4),alpha=0.25, fill="#145fa7")+

labs(

x = expression(atop("Prey abundance", paste("for short term response (2h)"))),

y = expression("Prey consumed"),

title = "Fitness of Chrysoperla agilis on Myzus persicae: Posterior predictive Cr.I. vs data",

subtitle = paste0(type," Holling-II model"))+

scale\_colour\_manual("Legend",labels = c("Data","Median"),

values = c("black","red"),

guide = guide\_legend(override.aes = list(

linetype = c(rep("blank", 1), "solid"),

shape = c(16, NA,NA),

fill=c("#81DAF5", "#81DAF5"))))+

guides( color = guide\_legend(

order = 1,

override.aes = list(

color = c("black", "red"),

fill = c("white", "white"),

linetype = c("blank", "solid"),

shape = c(16, NA)))) +

theme\_bw() + # remove legend key border color & background

theme(text = element\_text(size=16),

plot.title = element\_text(size=14),

plot.subtitle = element\_text(size=14,hjust = 0.5),

legend.key = element\_rect(colour = NA, fill = NA),

legend.box.background = element\_blank())+

geom\_point(aes(x = x, y = y, size = "95% Cr.I.", shape = NA)) +

geom\_point(aes(x = x, y = y, size = "50% Cr.I.", shape = NA)) +

guides(size = guide\_legend(NULL,

order = 2,

override.aes = list(shape = c(15,15),

color = c("#145fa7","#81DAF5"),

size = c(6,6) )))

}else

{ggplot(data = yndata, aes(x = x, y = y)) +

geom\_point(data=ndata, size = 1.5, aes(x = x, y = y, colour = "#000000"),alpha=1) +

geom\_line(aes(x = x, y = median, colour="median"), size=1)+

geom\_ribbon(aes(x = x,ymin=y1,ymax=y2),alpha=0.25, fill="#81DAF5")+

geom\_ribbon(aes(x = x,ymin=y3,ymax=y4),alpha=0.25, fill="#145fa7")+

labs(

x = expression(atop("Prey abundance", paste("for long term response (24h)"))),

y = expression("Prey consumed"),

title = "Fitness of Chrysoperla agilis on Myzus persicae: Posterior predictive Cr.I. vs data",

subtitle = paste0(type," Holling-II model"))+

scale\_colour\_manual("Legend",labels = c("Data","Median"),

values = c("black","red"),

guide = guide\_legend(override.aes = list(

linetype = c(rep("blank", 1), "solid"),

shape = c(16, NA,NA),

fill=c("#81DAF5", "#81DAF5"))))+

guides( color = guide\_legend(

order = 1,

override.aes = list(

color = c("black", "red"),

fill = c("white", "white"),

linetype = c("blank", "solid"),

shape = c(16, NA)))) +

theme\_bw() +

# remove legend key border color & background

theme(text = element\_text(size=16),

plot.title = element\_text(size=14),

plot.subtitle = element\_text(size=14,hjust = 0.5),

legend.key = element\_rect(colour = NA, fill = NA),

legend.box.background = element\_blank())+

geom\_point(aes(x = x, y = y, size = "95% Cr.I.", shape = NA)) +

geom\_point(aes(x = x, y = y, size = "50% Cr.I.", shape = NA)) +

guides(size = guide\_legend(NULL,

order = 2,

override.aes = list(shape = c(15,15),

color = c("#145fa7","#81DAF5"),

size = c(6,6) )))}

# #F: Calculate and record information criteria and elapsed time

library(loo) #load essential library

#get elapsed time

ttime<-get\_elapsed\_time(eval(parse(text= model\_obj)))

#merge results from different MC chains#

array.stan <- as.array(eval(parse(text= model\_obj)))

for( i in 1:ncol(array.stan) ){

tempo <- as.data.frame(array.stan[,i,])

tempo$chain <- i

tempo$iteration <- 1:nrow(tempo)

if ( i==1 ) df.res <- tempo

else df.res <- rbind(df.res, tempo)

}

rm(tempo, array.stan, i)

mean.Deviance <- mean(df.res$dev) # deviance posterior mean

var.Deviance <- var(df.res$dev) # deviance posterior variance

df.input<-as.data.frame(eval(parse(text= model\_obj))) # stan fit as data frame

# values of number of prey consumed and initial densities of prey

Ne <- as.integer(newdata[,2\*inum])

A<- as.integer(newdata[,(2\*inum-1)])

#find summary statistics for probability of being consumed and calculate deviance

if (type=="deterministic") res<-summary(eval(parse(text= model\_obj)),pars="p")$summary else

res<-summary(eval(parse(text= model\_obj)),pars="lambda")$summary[(NN+1):(2\*NN),]

lp<-as.vector(res[,"mean"]) # probability of being consumed mean

Deviance.mean <- -2\*sum(with(df.input, dbinom(Ne, A, lp, log = T) ))

#calculate DIC, AIC, BIC, WAIC, LooIC

pD <- mean.Deviance - Deviance.mean

DIC\_res<-c(mean.Dev=mean.Deviance, Dev.mean=Deviance.mean, pD=mean.Deviance-Deviance.mean, DIC=pD+mean.Deviance)

param<-ifelse((parms==c("theta","dev")),n\_pars+1,n\_pars+1+2)[1] #deterministic case uses +1 params for p, deterministic case uses +3 params for p,k,mu

AIC\_BIC<-c(AIC=mean(df.res$dev)+2\*param,

BIC=mean(df.res$dev)+param\*log(length(Ne)))

lik\_sss<-extract\_log\_lik( eval(parse(text= paste0("mod",inum,"\_sss"))), parameter\_name = "log\_lik",

merge\_chains = TRUE)

criteria<-rbind(as.data.frame(loo(lik\_sss)$estimates),as.data.frame(waic(lik\_sss)$estimates))

# model details and information criteria results saved in data frames

result1<-as.data.frame(summary(eval(parse(text= paste0("mod",inum,"\_sss"))),pars=parms)$summary) # write the summary R-STAN output

result2<-data.frame(c(model\_obj,nch,iter,bin,nth,nch\*(iter-bin)/nth), row.names = c("Model Name","Number of chains","iterrations","Burn in","Thinning ratio","sample size"))

colnames(result2)="data"

# choose file name and folder to save output

fold<-"C:\\Users\\...\\"

if (type=="deterministic") ccfile<-paste0(fold, model\_obj,N,"\_no\_",format(Sys.Date(), "%d%m%Y"),".xls") else ccfile<-paste0(fold, model\_obj,N," \_ou\_",format(Sys.Date(), "%d%m%Y"),".xls")[1]

# write output in excel file

write.xlsx(result2, ccfile)

write.xlsx(result1, ccfile, append = TRUE, sheetName="results")

write.xlsx(ttime/60,ccfile, append = TRUE, sheetName="elapsed\_time")

write.xlsx(AIC\_BIC, ccfile, append = TRUE, sheetName="AIC\_BIC")

write.xlsx(DIC\_res, ccfile, append = TRUE, sheetName="DIC")

write.xlsx(criteria, ccfile, append = TRUE, sheetName="criteria")

write.xlsx(newdata, ccfile, append = TRUE, sheetName="data")