

Software code for Bayesian Risk Assessment of Phytoplankton Diversity Loss

Malay Naskar Soma Das Sarkar, S.K. Sahu, Pranab Gogoi, B.K. Das

Summary:

The software code presented is the computational code for estimating relative risk of species loss induced by barge movement. The theoretical development can be found in Naskar et al. (2021), which is a Bayesian Generalized Linear Mixed Model. Computational algorithm used ‘JAGS’ (Plummer, 2003), which implements BUGS, for fitting the model via R (R Core Team, 2019) using ‘R2jags’(Su and Yajima, 2012)—which is an R package that allows fitting JAGS models from within R. The codes for model-fitting are provided in ‘*R_code S1*’ and ‘*BUG_code S2*’).

References:

- Naskar, Malay, Soma Das Sarkar, S.K. Sahu, Pranab Gogoi, B.K. Das (2021) Impact of barge movement on phytoplankton diversity in a river: A Bayesian risk estimation framework, *Journal of Environmental Management*, Volume 296, <https://doi.org/10.1016/j.jenvman.2021.113227>
- Plummer, M. 2003 JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling, pp. 1-10, Vienna, Austria.
- Su, Y.-S. and Yajima, M. 2012. R2jags: A Package for Running jags from R. R package version 0.03-08, URL <http://CRAN.R-project.org/package=R2jags>.

#####

R code: S1 (Main program)

```
#####bayesian analysis ###
```

```
library(readxl)
```

```
library(ggplot2)
```

```
library(R2jags) ###library for interfacing R and JAGS
```

```
##### The data #####
```

```
#dat<-read.csv(file="species_ns_s.csv",header=TRUE)
```

```
#dat$T<-dat$NS+dat$S
```

```
#dat<-dat[,2:8]
```

```
dat<- as.data.frame(read_excel("dat-plankton-abund.xlsx"))
```

```
dat$Barge<-as.factor(dat$Barge)
```

```
dat$Site<-as.factor(dat$Site)
```

```

dat$Zone<-as.factor(dat$Zone)
dat.NS.S<-function(dat,type="presence")
{
  sites<-unique(dat$Site)
  state<-unique(dat$Barge)
  n.sites<-length(sites)
  n.state<-length(state)
  d1<-matrix(data=NA,nrow=1,ncol=4)
  d1<-as.data.frame(d1)
  colnames(d1)<-c("Site","Assemblage","NS","S")
  for(s in sites)
  {
    comm<-as.matrix(dat[dat$Site==s,5:56])
    n.row<-nrow(comm)
    for(i in 1:(n.row-1))
    {
      samp1<-comm[i,]
      for(j in (i+1):n.row)
      {
        samp2<-comm[j,]
        pair.samp<-rbind(samp1,samp2)
        PA<-ifelse(pair.samp>0,1,0)
        if (type=="presence") {S<-PA%*%t(PA)}
        if(type=="abundance") {S<-pair.samp%*%t(PA)}
        NS<-diag(S)-S
        d1<-rbind(d1,c(s,1,NS[1,2],S[1,2]),c(s,0,NS[2,1],S[2,1]))
      }
    }
  }
  d1<-d1[-1,]
  d1<-cbind(d1,Pair=c("BD","BA","DA"))
  Zone<-
  ifelse(d1$Site%in%c("BKP","BRNGR"),"Z1",ifelse(d1$Site%in%c("TRVN","BLGR"),
  "Z2","Z3"))
  d1<-cbind(d1,Zone=Zone)
  d1$Site<-as.factor(d1$Site)
}

```

```

    d1$Assemblage<-as.factor(d1$Assemblage)
    d1$NS<-as.numeric(d1$NS)
    d1$S<-as.numeric(d1$S)
    return(d1)
}
d<-dat.NS.S(dat,type="abundance")
#d<-dat.NS.S(dat,type="presence")
d$Pair<-as.factor(d$Pair)
d$Zone<-as.factor(d$Zone)
##### Setting Data #####
bayes.data<-list(N=nrow(d),NS=d$NS, Site=rep(c(1:6),rep(6,6)),T=d$NS+d$S,
Barge=as.numeric(d$Assemblage)-1,Pair=as.numeric(d$Pair)-2)
bayes.data$S<-length(unique(dat$Site))
#####Initial Values ###
inits1<-list(alpha=-1.0,beta=0.8,gamma=0)
inits2<-list(alpha=-0.9,beta=0.47,gamma=0.19)
inits3<-list(alpha=-0.5,beta=-0.47,gamma=-0.19)
bayes.init<-list(inits1,inits2,inits3)
param.monitor<-c("rr","u","BA","BD","DA")
jags.fit<-jags(data=bayes.data,inits =
bayes.init,param.monitor,n.iter=50000,model.file="bug_binomial_log.txt")
jags.mcmc<-as.mcmc(jags.fit)
conv.test<-gelman.diag(jags.mcmc)

##### Require
require(ggmcmc)
jags.gg<-ggs(jags.mcmc)
ggs_autocorrelation(jags.gg,family="rr")
ggs_traceplot(jags.gg,family="rr")
ggs_density(jags.gg,family="rr")
jpeg(file="site-BA-HPD.jpg",width=4,height=3,units="in",quality = 100,res=200)
g<-ggs_caterpillar(jags.gg,family="BA",sort=FALSE)
g<-g+geom_vline(xintercept = 1, col="red",linetype="dashed",size=1)
g<-g+scale_y_discrete(labels=unique(d$Site))
g<-g+ylab("Site")
g+geom_text(x=1.6,y=6,label="(a)")

```

```

dev.off()
jpeg(file="site-BD-HPD.jpg",width=4,height=3,units="in",quality = 100,res=200)
g<-ggs_caterpillar(jags.gg,family="BD",sort=FALSE)
g<-g+geom_vline(xintercept = 1, col="red",linetype="dashed",size=1)
g<-g+scale_y_discrete(labels=unique(d$Site))
g<-g+ylab("Site")
g+geom_text(x=1.6,y=6,label="(b)")
dev.off()
jpeg(file="site-DA-HPD.jpg",width=4,height=3,units="in",quality = 100,res=200)
g<-ggs_caterpillar(jags.gg,family="DA",sort=FALSE)
g<-g+geom_vline(xintercept = 1, col="red",linetype="dashed",size=1)
g<-g+scale_y_discrete(labels=unique(d$Site))
g<-g+ylab("Site")
g+geom_text(x=1.7,y=6,label="(c)")
dev.off()
##### Comparison with Frequentist
require(lme4)
require(nlme)
d$Assemblage<- abs(d$Assemblage -1)
glmm.fit<-
glmer(cbind(NS,S)~Assemblage+Pair+(1|Site),data=d,family=binomial(link="log"))

```

BUG code: S2("bug_binomial_log.txt")

Model

```

{
# N observations
for (i in 1:N)
  {
    p[i]<-exp(alpha+beta*Barge[i]+gamma*Pair[i])+u[Site[i]]
    NS[i] ~ dbin(p[i],T[i])
    ##logit(p[i]) <- alpha + beta*Barge[i]+ gamma*Pair[i]+u[Site[i]]
  }
for(j in 1:S)
  {
    u[j]~dnorm(0,tau)
  }
# Non-informative Priors
alpha ~ dnorm(0.0,1.0E-6)

```

```

beta ~ dnorm(0.0,1.0E-6)
gamma ~ dnorm(0.0,1.0E-6)
#### Informative priors
#alpha~dunif(-5,5)
#beta~dunif(-5,5)
#gamma~dunif(-5,5)
# Hyperprior
tau ~ dgamma(0.001,0.001)
rr[1]<-exp(beta) #####Before vs During
rr[2]<-exp(beta-gamma) ##### Before vs After
rr[3]<-exp(beta+gamma) ##### During vs After
for(j in 1:S)
  {
    BA[j]<-exp(beta+u[j])
    BD[j]<-exp(beta-gamma+u[j])
    DA[j]<-exp(beta+gamma+u[j])
  }
}

#####

```

Plummer, M. (2003). *JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling*. Paper presented at the Proceedings of the 3rd international workshop on distributed statistical computing.