Supplemental code 1. BOAM_R+JAGS_code.docx. R script with the sample code to run JAGS and R routines to perform the Bayesian Ordinal Animal Model

```
# Bayesian Animal Models for Brown Rot Susceptibility in Peach, using an
Ordered Logistic Model

setwd("~/BOAM_BR/") # setting working directory
library(rjags)
dat <- read.delim("BR_NomOrd.txt", header=TRUE, stringsAsFactors = FALSE,
na.strings = "NA", strip.white = TRUE)

# names(dat)

set.seed(100307) # Setting a seed for same random number generation

# Giving name to the data to use in JAGS
Ntrees <- length(unique(dat$anim)) # number of trees
nrec <- length(dat[,1])  # count size of data set
y <- dat$BRSuc
ncat <- max(y, na.rm=T)
ncat1 <- ncat - 1
sq_D   <- dat$Sq_D # Mendielian sampling term
pid <- dat$sire # father tree ID
sid <- dat$dam # mother tree ID
id <- dat$animal # tree ID
trt <- dat$Treatment # effect of treatment
yr <- dat$Year # effect of year (it is being imputed in JAGS for NAs)

# Building the data file for JAGS as usual, normal for var.a and everything
else
jag.dat <- list("id"=id, "sid"=sid, "pid"=pid, "sq_D"=sq_D, "nrec"=nrec,
"ncat"=ncat, "ncat1"=ncat1, "Ntrees"=Ntrees, "y"=y, "trt"=trt, "yr"=yr)
inits <- list("b.tr"=c(NA,0.1), "b.yr"=c(NA,0.1,-0.1), "alpha"=c(-
0.5,0,0.5,1))  #  alpha must have ncat - 1 elements

cat(
  "#   Program for the Bayesian Animal Model (applied to peaches) using
dcat()
  #   Brown Rot Suceptibility (BRSuc) in fruits
  #    Pedigree of 221 trees (Ntrees), 1034 observations (nrec)
  #   Treatments: 1 and 2 (non-wounded and wounded); Years: 1,2 and 3 (2007,
2008 and 2009)
  #   Very unbalanced dataset, several NA in covariates for trees upstream of
pedigree
  #   Year will be imputed.
  #   This is an application of the Ordered Logistic Model with 5 categories:
1,2,3,4,5,
  #   meaning: resistant, tolerant, low tolerant, susceptible and highly
susceptible
  #   Missing data for years

  #Specification of the reparameterized sampling distribution

  model{

  #   Loop for pyear, one vector for levels of the missing covariate (3).
  #   We use dcat to randomly generate a year class (2 or 3, since year 1 =
0; with equal frequencies).

  for(y in 1:3){
```

```
pyear[y] <- 1/3
}

# Generating year class
for (i in 1:nrec){
yr[i] ~ dcat(pyear[])
}
# Likelihood
for (i in 1:nrec){
for (j in 1:ncat1){
logit(theta[i,j]) <- alpha.s[j] - mu[i]
}
y[i] ~ dcat(p[i,1:ncat])

p[i,1]<- theta[i,1]
for(k in 2:ncat1){
p[i,k] <- theta[i,k] - theta[i,k-1]
}   #   end of k loop
p[i,ncat] <- 1-theta[i,ncat1]

mu[i] <- b.tr[trt[id[i]]] + b.yr[yr[id[i]]] + v[id[i]]*tau.a
v[i] <- tree[id[i]] * sq_D[id[i]] + ((v[pid[id[i]]] + v[sid[id[i]]])/2)
rr[i] <- (y[i] - mu[i])^2
}

# Phantom tree parents breeding values
v[1035] <- 0.0
v[1036] <- 0.0

# Randoms effect for each tree
for(j in 1:Ntrees){
tree[j] ~ dnorm(0.0,1.0) # a standard normal
}

# Priors
for (f in 1:ncat1) {
alpha[f] ~ dnorm(0,1.0E-3)
} #   end of f loop
alpha.s[1:ncat1] <- sort(alpha)
normal ~ dnorm(0,1.0E-3)
tau.a <- abs(normal) #Normal
b.tr[1] <- 0.0
b.tr[2] ~ dnorm(0,1.0E-3)
b.yr[1] <- 0.0
for(j in 2:3){
b.yr[j] ~ dnorm(0,1.0E-3)
}   #  end of j loop

# Calculating the additive genetic variance
  var.a <- log(tau.a)^2     # Additive genetic variance

# Breeding Values
for(k in 1:Ntrees){
EBV[k] <- v[k]*(tau.a)
}

# Heritability
  h2 <-  var.a/(var.a^2+((3.1416^2/3))

}  #   end of model loop"
```

```
  , file="BOAM_Ordnormal.jag")

  mod.JAGSn <- jags.model(file="BOAM_Ordnormal.jag", data=jag.dat,
inits=inits, n.chains=3)
  update(mod.JAGSn,10000)
  par.monn <- c("var.a","var.e","var.p","EBV","h2","b.tr","b.yr","alpha")
  post.sampn <- coda.samples(mod.JAGSn,par.monn,n.iter=170000)
  subsampn <- window(post.sampn,start=20001,end=170000, thin=30)
  summary(subsampn)
  BVsn<-summary(subsampn)
  dic.pDn <- dic.samples(mod.JAGSn, 1000, "pD") # Deviance Information
Criterion
  dic.poptn <- dic.samples(mod.JAGSn, 1000, "popt") # Penalized expected
deviance

  # Convergence Diagnostics Plots
  #plot(subsampn)
  trcdenn<-plot(subsampn[,231:234]) # Heritability, var.a, var.e and var.p
  #autocorrn.plot(subsampn)
  autcorn<-autocorr.plot(subsampn[,231:234]) # Heritability, var.a, var.e and
var.p
  #gelman.plot(subsampn)
  gplotn<-gelman.plot(subsampn[,231:234]) # Heritability, var.a, var.e and
var.p
  # Tabular diagnostics
  #gelman.diag(subsampn, confidence=0.95, multivariate=TRUE) # Above is 1.1
bad
  gelman.diag(subsampn[,231:234], confidence=0.95, multivariate=TRUE) # Above
1.1 is bad
  #raftery.diag(subsampn, q = 0.025, r = 0.005, s = 0.95, converge.eps=0.001)
# Above 5 is bad
  raftery.diag(subsampn[,231:234], q = 0.025, r = 0.005, s = 0.95,
converge.eps=0.001) # Above 5 is bad
  #heidel.diag(subsampn)
  heidel.diag(subsampn[,231:234])

  #   Building the data file for JAGS as usual, gamma for var.a and normal
everything else
  jag.dat <- list("id"=id, "sid"=sid, "pid"=pid, "sq_D"=sq_D,"nrec"=nrec,
"ncat"=ncat, "ncat1"=ncat1, "Ntrees"=Ntrees, "y"=y,"trt"=trt, "yr"=yr)
  inits <- list("b.tr"=c(NA,0.1), "b.yr"=c(NA,0.1,-0.1), "alpha" = c(-
0.5,0,0.5,1))  #  alpha must have ncat - 1 elements

  cat(
  "#   Program for the Bayesian Animal Model (applied to peaches) using
dcat()
  #   Brown Rot Suceptibility (BRSuc) in fruits
  #    Pedigree of 221 trees (Ntrees), 1034 observations (nrec)
  #   Treatments: 1 and 2 (non-wounded and wounded); Years: 1,2 and 3 (2007,
2008 and 2009)
  #   Very unbalanced dataset, several NA in covariates for trees upstream of
pedigree
  #   Year will be imputed.
  #    This is an application of the Ordered Logistic Model with 5 categories:
1,2,3,4,5,
  #   meaning: resistant, tolerant, low tolerant, susceptible and highly
susceptible
  #   Missing data for years

  # Specification of the reparameterized sampling distribution
  model{

    # Loop for pyear, one vector for levels of the missing covariate (3).
    # We use dcat to randomly generate a year class (2 or 3, since year 1 =
0; with equal frequencies).

    for(y in 1:3){
      pyear[y] <- 1/3
    }

    # Generating year class
    for (i in 1:nrec){
      yr[i] ~ dcat(pyear[])
    }
    # Likelihood
    for (i in 1:nrec){
      for (j in 1:ncat1){
        logit(theta[i,j]) <- alpha.s[j] - mu[i]
      }
      y[i] ~ dcat(p[i,1:ncat])

      p[i,1]<- theta[i,1]
      for(k in 2:ncat1){
        p[i,k] <- theta[i,k] - theta[i,k-1]
      }   #   end of k loop
      p[i,ncat] <- 1-theta[i,ncat1]

      mu[i] <- b.tr[trt[id[i]]] + b.yr[yr[id[i]]] + v[id[i]]*tau.a
      v[i] <- tree[id[i]] * sq_D[id[i]] + ((v[pid[id[i]]] + v[sid[id[i]]])/2)
      rr[i] <- (y[i] - mu[i])^2
    }

    # Phantom tree parents breeding values
    v[1035] <- 0.0
    v[1036] <- 0.0

    # Randoms effect for each tree
    for(j in 1:Ntrees){
      tree[j] ~ dnorm(0.0,1.0) # a standard normal
    }

    # Priors
    for (f in 1:ncat1) {
      alpha[f] ~  dnorm(0,1.0E-3)
    } #   end of f loop
    alpha.s[1:ncat1] <- sort(alpha)
    tau.a ~ dgamma(0.001,0.001) #gamma
    b.tr[1] <- 0.0
    b.tr[2] ~  dnorm(0,1.0E-3)
    b.yr[1] <- 0.0
    for(j in 2:3){
      b.yr[j] ~ dnorm(0,1.0E-3)
    }   #  end of j loop

    # Calculating the additive genetic variance
    var.a <- log(tau.a)^2     # Additive genetic variance

  # Breeding Values
  for(k in 1:Ntrees){
```

```
  EBV[k] <- v[k]*(tau.a)
  }

  # Heritability
    h2 <-  var.a/(var.a^2+((3.1416^2/3))


  }  #   end of model loop"
  , file="BOAM_Ordgamma.jag")

mod.JAGSg <- jags.model(file="BRAM_Ordgamma.jag", data=jag.dat, inits=inits,
n.chains=3)
update(mod.JAGSg,10000)
par.mong <- c("var.a", "var.e", "var.p", "EBV", "h2", "b.tr", "b.yr",
"alpha")
post.sampg <- coda.samples(mod.JAGSg,par.mong,n.iter=170000)
subsampg <- window(post.sampg,start=20001,end=170000, thin=30)
summary(subsampg)
BVsg<-summary(subsampg)
dic.pDg <- dic.samples(mod.JAGSg, 1000, "pD") # Deviance Information
Criterion
dic.poptg <- dic.samples(mod.JAGSg, 1000, "popt") # Penalized expected
deviance

# Convergence Diagnostics Plots
plot(subsampg)
trcdeng<-plot(subsampg[,231:234]) # Heritability, var.a, var.e and var.p
#autocorrg.plot(subsampg)
autcorg<-autocorr.plot(subsampg[,231:234]) # Heritability, var.a, var.e and
var.p
#gelman.plot(subsampg)
gplotg<-gelman.plot(subsampg[,231:234]) # Heritability, var.a, var.e and
var.p
# Tabular diagnostics
#gelman.diag(subsampg, confidence=0.95, multivariate=TRUE) # Above is 1.1 bad
gelman.diag(subsampg[,231:234], confidence=0.95, multivariate=TRUE) # Above
1.1 is bad
#raftery.diag(subsampg, q = 0.025, r = 0.005, s = 0.95, converge.eps=0.001) #
Above 5 is bad
raftery.diag(subsampg[,231:234], q = 0.025, r = 0.005, s = 0.95,
converge.eps=0.001) # Above 5 is bad
#heidel.diag(subsampg)
heidel.diag(subsampg[,231:234])

#   Building the data file for JAGS as usual, Half-Cauchy for var.a and
normal for everything else
jag.dat <- list("id"=id, "sid"=sid, "pid"=pid, "sq_D"=sq_D,
"nrec"=nrec,"ncat"=ncat, "ncat1"=ncat1, "Ntrees"=Ntrees, "y"=y, "trt"=trt,
"yr"=yr)
inits <- list("b.tr"=c(NA,0.1), "b.yr"=c(NA,0.1,-0.1), "alpha" = c(-
0.5,0,0.5,1))  #  alpha must have ncat - 1 elements

cat(
  "#   Program for the Bayesian Animal Model (applied to peaches) using
dcat()
  #   Brown Rot Suceptibility (BRSuc) in fruits
  #   Pedigree of 221 trees (Ntrees), 1034 observations (nrec)
  #   Treatments: 1 and 2 (non-wounded and wounded); Years: 1,2 and 3 (2007,
2008 and 2009)
```

```
  #   Very unbalanced dataset, several NA in covariates for trees upstream of
pedigree
  #   Year will be imputed.
  #    This is an application of the Ordered Logistic Model with 5 categories:
1,2,3,4,5,
  #   meaning: resistant, tolerant, low tolerant, susceptible and highly
susceptible
  #   Missing data for years

  #Specification of the reparameterized sampling distribution

  model{

  #   Loop for pyear, one vector for levels of the missing covariate (3).
  #   We use dcat to randomly generate a year class (2 or 3, since year 1 =
0; with equal frequencies).

  for(y in 1:3){
  pyear[y] <- 1/3
  }

  # Generating year class
  for (i in 1:nrec){
  yr[i] ~ dcat(pyear[])
  }
  # Likelihood
  for (i in 1:nrec){
  for (j in 1:ncat1){
  logit(theta[i,j]) <- alpha.s[j] - mu[i]
  }
  y[i] ~ dcat(p[i,1:ncat])

  p[i,1]<- theta[i,1]
  for(k in 2:ncat1){
  p[i,k] <- theta[i,k] - theta[i,k-1]
  }   #   end of k loop
  p[i,ncat] <- 1-theta[i,ncat1]

  mu[i] <- b.tr[trt[id[i]]] + b.yr[yr[id[i]]] + v[id[i]]*tau.a
  v[i] <- tree[id[i]] * sq_D[id[i]] + ((v[pid[id[i]]] + v[sid[id[i]]])/2)
  rr[i] <- (y[i] - mu[i])^2
  }

  # Phantom tree parents breeding values
  v[1035] <- 0.0
  v[1036] <- 0.0

  # Randoms effect for each tree
  for(j in 1:Ntrees){
  tree[j] ~ dnorm(0.0,1.0) # a standard normal
  }

  # Priors
  for (f in 1:ncat1) {
  alpha[f] ~ dnorm(0,1.0E-3)
  } #   end of f loop
  alpha.s[1:ncat1] <- sort(alpha)
  cauchy ~ dt(0,1,1)
  tau.a <- abs(cauchy) #half-Cauchy
  b.tr[1] <- 0.0
  b.tr[2] ~ dnorm(0,1.0E-3)
  b.yr[1] <- 0.0
  for(j in 2:3){
  b.yr[j] ~ dnorm(0,1.0E-3)
  }   #  end of j loop

  # Calculating the additive genetic variance
    var.a <- log(tau.a)^2     # Additive genetic variance

  # Breeding Values
  for(k in 1:Ntrees){
  EBV[k] <- v[k]*(tau.a)
  }

  # Heritability
    h2 <-  var.a/(var.a^2+((3.1416^2/3))


  }  #   end of model loop"
  , file="BOAM_Ordcauchy.jag")

  mod.JAGSc <- jags.model(file="BOAM_Ordcauchy.jag", data=jag.dat,
inits=inits, n.chains=3)
  update(mod.JAGSc,10000)
  par.monc <- c("var.a","var.e","var.p","EBV","h2","b.tr","b.yr","alpha")
  post.sampc <- coda.samples(mod.JAGSc,par.monc,n.iter=170000)
  subsampc <- window(post.sampc,start=20001,end=170000, thin=30)
  summary(subsampc)
  BVsc<-summary(subsampc)
  dic.pDc <- dic.samples(mod.JAGSc, 1000, "pD") # Deviance Information
Criterion
  dic.poptc <- dic.samples(mod.JAGSc, 1000, "popt") # Penalized expected
deviance

  # Convergence Diagnostics Plots
  #plot(subsampc)
  trcdenc<-plot(subsampc[,231:234]) # Heritability, var.a, var.e and var.p
  #autocorrn.plot(subsampc)
  autcorc<-autocorr.plot(subsampc[,231:234]) # Heritability, var.a, var.e and
var.p
  #gelman.plot(subsampc)
  gplotc<-gelman.plot(subsampc[,231:234]) # Heritability, var.a, var.e and
var.p
  # Tabular diagnostics
  #gelman.diag(subsampc, confidence=0.95, multivariate=TRUE) # Above is 1.1
bad
  gelman.diag(subsampc[,231:234], confidence=0.95, multivariate=TRUE) # Above
1.1 is bad
  #raftery.diag(subsampc, q = 0.025, r = 0.005, s = 0.95, converge.eps=0.001)
# Above 5 is bad
  raftery.diag(subsampc[,231:234], q = 0.025, r = 0.005, s = 0.95,
converge.eps=0.001) # Above 5 is bad
  #heidel.diag(subsampc)
  heidel.diag(subsampc[,231:234])

  #   Building the data file for JAGS as usual, uniform (1/sqrt(pi
squared/3)) for var.a and normal for everything else
  jag.dat <- list("id"=id, "sid"=sid, "pid"=pid, "sq_D"=sq_D, "nrec"=nrec,
"ncat"=ncat, "ncat1"=ncat1, "Ntrees"=Ntrees, "y"=y, "trt"=trt, "yr"=yr)
```

```
  inits <- list("b.tr"=c(NA,0.1), "b.yr"=c(NA,0.1,-0.1), "alpha" = c(-
0.5,0,0.5,1))  #  alpha must have ncat - 1 elements

  cat(
  "#   Program for the Bayesian Animal Model (applied to peaches) using
dcat()
  #   Brown Rot Suceptibility (BRSuc) in fruits
  #   Pedigree of 221 trees (Ntrees), 1034 observations (nrec)
  #   Treatments: 1 and 2 (non-wounded and wounded); Years: 1,2 and 3 (2007,
2008 and 2009)
  #   Very unbalanced dataset, several NA in covariates for trees upstream of
pedigree
  #   Year will be imputed.
  #   This is an application of the Ordered Logistic Model with 5 categories:
1,2,3,4,5,
  #   meaning: resistant, tolerant, low tolerant, susceptible and highly
susceptible
  #   Missing data for years

  #Specification of the reparameterized sampling distribution

  model{

    # Loop for pyear, one vector for levels of the missing covariate (3).
    # We use dcat to randomly generate a year class (2 or 3, since year 1 =
0; with equal frequencies).

    for(y in 1:3){
      pyear[y] <- 1/3
    }

    # Generating year class
    for (i in 1:nrec){
      yr[i] ~ dcat(pyear[])
    }
    # Likelihood
    for (i in 1:nrec){
      for (j in 1:ncat1){
        logit(theta[i,j]) <- alpha.s[j] - mu[i]
      }
      y[i] ~ dcat(p[i,1:ncat])

      p[i,1]<- theta[i,1]
      for(k in 2:ncat1){
        p[i,k] <- theta[i,k] - theta[i,k-1]
      }   #   end of k loop
      p[i,ncat] <- 1-theta[i,ncat1]

      mu[i] <- b.tr[trt[id[i]]] + b.yr[yr[id[i]]] + v[id[i]]*tau.a
      v[i] <- tree[id[i]] * sq_D[id[i]] + ((v[pid[id[i]]] + v[sid[id[i]]])/2)
      rr[i] <- (y[i] - mu[i])^2
    }

    # Phantom tree parents breeding values
    v[1035] <- 0.0
    v[1036] <- 0.0

    # Randoms effect for each tree
    for(j in 1:Ntrees){
      tree[j] ~ dnorm(0.0,1.0) # a standard normal
    }

    # Priors
    for (f in 1:ncat1) {
      alpha[f] ~ dnorm(0,1.0E-3)
    } #   end of f loop
    alpha.s[1:ncat1] <- sort(alpha)
    cauchy ~ dt(0,1,1)
    tau.a <- abs(cauchy)    #half-Cauchy
    b.tr[1] <- 0.0
    b.tr[2] ~ dnorm(0,1.0E-3)
    b.yr[1] <- 0.0
    for(j in 2:3){
      b.yr[j] ~ dnorm(0,1.0E-3)
    }   #  end of j loop

    # Calculating the additive genetic variance
    var.a <- log(tau.a)^2     # Additive genetic variance

  # Breeding Values
  for(k in 1:Ntrees){
  EBV[k] <- v[k]*(tau.a)
  }

  # Heritability
    h2 <-  var.a/(var.a^2+((3.1416^2/3))


  }  #   end of model loop"
  , file="BOAM_Orduniform.jag")

mod.JAGSu <- jags.model(file="BOAM_Orduniform.jag", data=jag.dat,
inits=inits, n.chains=3)
update(mod.JAGSu,10000)
par.monu <- c("var.a", "var.e", "var.p", "EBV", "h2", "b.tr", "b.yr",
"alpha")
post.sampu <- coda.samples(mod.JAGSu,par.monu,n.iter=170000)
subsampu <- window(post.sampu,start=20001,end=170000, thin=30)
summary(subsampu)
BVsu<-summary(subsampu)
dic.pDu <- dic.samples(mod.JAGSu, 1000, "pD") # Deviance Information
Criterion
dic.poptu <- dic.samples(mod.JAGSu, 1000, "popt") # Penalized expected
deviance

# Convergence Diagnostics Plots
#plot(subsampu)
trcdenu<-plot(subsampu[,231:234]) # Heritability, var.a, var.e and var.p
#autocorrn.plot(subsampu)
autcoru<-autocorr.plot(subsampu[,231:234]) # Heritability, var.a, var.e and
var.p
#gelman.plot(subsampu)
gplotu<-gelman.plot(subsampu[,231:234]) # Heritability, var.a, var.e and
var.p
# Tabular diagnostics
#gelman.diag(subsampu, confidence=0.95, multivariate=TRUE) # Above is 1.1 bad
gelman.diag(subsampu[,231:234], confidence=0.95, multivariate=TRUE) # Above
1.1 is bad
#raftery.diag(subsampu, q = 0.025, r = 0.005, s = 0.95, converge.eps=0.001) #
Above 5 is bad
```

```
raftery.diag(subsampu[,231:234], q = 0.025, r = 0.005, s = 0.95,
converge.eps=0.001) # Above 5 is bad
#heidel.diag(subsampu)
heidel.diag(subsampc[,231:234])

# Comparing DIC among priors
diffdic(dic.pDn, dic.pDg) # For model comparison, if negative, normal is
better than gamma
diffdic(dic.pDn, dic.pDc) # For model comparison, if negative, normal is
better than cauchy
diffdic(dic.pDn, dic.pDu) # For model comparison, if negative, normal is
better than uniform
diffdic(dic.pDc, dic.pDg) # For model comparison, if negative, cauchy is
better than gamma
diffdic(dic.pDc, dic.pDu) # For model comparison, if negative, cauchy is
better than uniform
diffdic(dic.pDg, dic.pDu) # For model comparison, if negative, gamma is
better than uniform

save.image("BOAM.RData")
```