

Supplement S5. The R-code used to fit the modified model (1)

```
library(nimble)
library(coda)
library(MCMCpack)
library(MCMCvis)

trial<-read.table("C:/starykomp/programy/old/orgconv/orgvscon32.txt",h=TRUE)

N<-length(trial$yield)
n<-N/2
ne<-12
nv<-7
nr<-3

men1<-aggregate(yield/10~env,data=trial[1:n,],na.rm=TRUE,FUN="mean")
vae1<-var(men1[,2])
men2<-aggregate(yield/10~env,data=trial[(n+1):N,],na.rm=TRUE,FUN="mean")
vae2<-var(men2[,2])

Senv<-matrix(nrow=2,ncol=2)
Senv[1,1]<-vae1/2
Senv[1,2]<-0
Senv[2,1]<-0
Senv[2,2]<-vae2/2

y<-trial$yield/10#;y
sys<-trial$sys
env<-trial$env
rep<-trial$rep
v<-trial$var
exper<-trial$exper
m1<-mean(y[1:n])
m2<-mean(y[(n+1):N])
ma<-c(m1,m2);ma

#### Model definition
model<-nimbleCode({
  for(i in 1:N){
    y[i]~dnorm(mu[i],tau=tau[exper[i]])
    mu[i]<-
  Env[env[i],sys[i]]+b[rep[i],env[i],sys[i]]+va[v[i],sys[i]]+ge[env[i],v[i],sys[i]]
  }

  for(s in 1:24){tau[s]~dgamma(1,0.001)
    sig2[s]<-1/tau[s]}
  for(l in 1:2){
    a[l]~dnorm(ma[l],tau=0.001)
    sigR[l]~T(dt(0,pow(10,-2),1),0,)
    sig2R[l]<-pow(sigR[l],2)
    sigV[l]~T(dt(0,pow(10,-2),1),0,)
    sig2V[l]<-pow(sigV[l],2)
    ksige[l]~dgamma(0.5,0.0005)
  }
  cc[1]<-1
  cc[2]<-1
  S4[1:2,1:2]<-4*diag(ksige[1:2])
  covE[1:2,1:2]~dinvwish(S=Senv[1:2,1:2],df=4)
  covEV[1:2,1:2]~dinvwish(S=S4[1:2,1:2],df=3)
  mul[1]<-0
  mul[2]<-0
  for(r in 1:nr){
    for(j in 1:ne){
      for(l in 1:2){
        b[r,j,l]~dnorm(0,var=sig2R[l])
      }
    }
  }
}
```

```

for(j in 1:ne){
  Env[j,1:2]~dmnorm(a[1:2],cov=covE[1:2,1:2])
}

for(k in 1:nv){
  for(l in 1:2){
    va[k,l]~dnorm(0,var=sig2V[l])
  }
}

for(k in 1:nv){
  for(l in 1:2){
    MV[k,l]<-va[k,l]+a[l]}
}

for(j in 1:ne){
  for(k in 1:nv){
    ge[j,k,1:2]~dmnorm(mul[1:2],cov=covEV[1:2,1:2])
  }
}
rhoE<-covE[1,2]/(sqrt(covE[1,1]*covE[2,2]))
rhoEV<-covEV[1,2]/(sqrt(covEV[1,1]*covEV[2,2]))
for(k in 1:nv){
  for(l in 1:2){
    for(j in 1:ne){
      Vg[k,j,l]<-Env[j,l]+va[k,l]+ge[j,k,l]
      RVE[k,j,l]<-(Vg[k,j,l]/Env[j,l])
      diff[k,j,l]<-Vg[k,j,l]-max(Vg[1:nv,j,l])
    }
    HMRPVG[k,l]<-ne/(sum(1/RVE[k,1:ne,l]))
    Sup[k,l]<-sum((diff[k,1:ne,l])^2)/(2*ne)
  }
}
}
)

X<-riwish(4,diag(2))
Z1<-riwish(3,diag(2))
Z2<-riwish(3, diag(c(0.001,500),2));Z2
Z3<-riwish(3,diag(c(500,0.1),2))
pro<-c(0.1,0.45,0.45)
pro2<-c(0.01,0.65,0.34)
z<-c(100,1000,0.001)
xx<-sample(z, size=24,replace = TRUE,prob=pro)
w<-c(100,0.001,1000)
yy<-sample(w,size=24,replace=TRUE,prob=pro2)
pp<-rinvgamma(24, shape=0.5, scale=0.5)

#### Data and constants

modelConsts<-list(N=N,ne=ne,nv=nv,nr=nr,rep=c(rep),env=c(env),
  sys=c(sys),v=c(v),exper=c(exper))
modelData<-list(y=c(y),Senv=Senv,ma=ma)

### initial values for covariance matrices

inits1<-list(sigR=c(10,10),sigV=c(15,15),
  covE=diag(2),ksige=rgamma(2,0.5,0.0005),covEV=diag(2),tau=c(yy))
inits2<-list(sigR=c(2,15),sigV=c(2,18),
  covE=diag(c(0.001,0.001),2),ksige=rgamma(2,0.5,0.0005),
  covEV=Z2,tau=c(xx))
inits3<-list(sigR=c(5,20),sigV=c(20,5),
  covE=diag(c(1500,1500),2),ksige=rgamma(2,0.5,0.0005),
  covEV=Z3,tau=c(pp))

Inits<- list(inits1,inits2,inits3)

```

```

#### MCMC setup
Rmodel <- nimbleModel(code = model, constants = modelConsts,data =
modelData,inits=inits1)

myModel <- compileNimble(Rmodel)
myMCMC <- buildMCMC(Rmodel)
myMCMCconf<-configureMCMC(Rmodel,enableWAIC=TRUE,print=TRUE)
myMCMCconf$addMonitors(c("a","sig2","sigV","sig2V","covE","covEV","tau",
                        "b","sigR","sig2R","va","ge","Env","MV",
                        "HMRPVG","Sup","rhoE","rhoEV"))
CmyModel <- buildMCMC(myMCMCconf)
CmyMCMC <- compileNimble(CmyModel,project=myModel)

#### MCMC
mcmc.out <- runMCMC(CmyMCMC, niter = 2500000, nburnin=1250000,thin=40,
nchains = 3,inits = Inits,
summary = TRUE, progressBar=TRUE,
samples=TRUE,samplesAsCodaMCMC=TRUE)
print(mcmc.out$summary$all.chains)

##### Samples
mcmc1<-as.mcmc(mcmc.out$samples$chain1)#
mcmc2<-as.mcmc(mcmc.out$samples$chain2)
mcmc3<-as.mcmc(mcmc.out$samples$chain3)
mcmcb<-mcmc.list(mcmc1,mcmc2,mcmc3)

#### MCMC summary statistics and WAIC

calculateWAIC(CmyMCMC)
MCMCsummary(mcmcb,params=c('a','MV','Env','sig2V','sig2R','covE',
                           'covEV','sig2','HMRPVG','Sup',
                           'rhoE','rhoEV'),
            n.eff=FALSE)

##### Diagnostic plots
gelman.diag(mcmcb,multivariate=FALSE)
gelman.plot(mcmcb,ask=TRUE)
plot(mcmcb,ask=TRUE)
MCMCplot(mcmcb,params=c('covV'),horiz=FALSE,HPD=TRUE)
MCMCplot(mcmcb,params=c('covE'),horiz=FALSE,HPD=TRUE)
MCMCplot(mcmcb,params=c('covEV'),horiz=FALSE,HPD=TRUE)

##### Generated quantities

means<-MCMCchains(mcmcb,params='MV')
mew1<-MCMCchains(mcmcb,params=c('MV[1, 1]','MV[5, 1]'),ISB=FALSE,exact=TRUE)
mean1<-rowMeans(mew1)
mew2<-MCMCchains(mcmcb,params=c('MV[1, 1]','MV[2, 1]','MV[3, 1]','
                                'MV[4, 1]','MV[5, 1]','MV[6, 1]','MV[7, 1]'),
                                ISB=FALSE,exact=TRUE)
mean2<-rowMeans(mew2)
mew3<-MCMCchains(mcmcb,params=c('MV[1, 2]','MV[2, 2]','MV[3, 2]','
                                'MV[4, 2]','MV[5, 2]','MV[6, 2]','MV[7, 2]'),
                                ISB=FALSE,exact=TRUE)
mean3<-rowMeans(mew3)
ind1<-matrix(nrow=length(means[,1]),ncol=7)
ind2<-matrix(nrow=length(means[,1]),ncol=7)
ind3<-matrix(nrow=length(means[,1]),ncol=7)
for(i in 1:length(means[,1])){
  for(j in 1:7){
    ind1[i,j]<-ifelse(means[i,j]>mean1[i],1,0)
    ind2[i,j]<-ifelse(means[i,j]>mean2[i],1,0)
    ind3[i,j]<-ifelse(means[i,j+7]>mean3[i],1,0)
  }
}
colMeans(ind1)
colMeans(ind2)

```

```
colMeans(ind3)
```