

## Appendix B

Objective 1. regulated versus unregulated sites

```
fish.dat<-read.csv("2005_2016_AMP.csv")
```

```
library(reshape)
```

```
library(reshape2)
```

```
library(tidyverse)
```

```
library(R2jags)
```

```
library(R2WinBUGS)
```

```
head(fish.dat)
```

```
# eto<-read.csv("Eto.28spp.10sites.presence.csv")
```

```
#
```

```
# eto<-Eto_28spp_10sites_presence
```

```
# eto$Site<-as.factor(eto$Site)
```

```
# library("reshape2")
```

```
# alldetections10<-melt(eto)
```

```
# alldetarray10<-array(NA, dim=c(28,10,2,15))
```

```
# alldetarray10[,,,]<-as.array(alldetections10$value)
```

```
# alldetections10$value
```

```

#'
+++++
+++++

#' Format data for occupancy model

#'
+++++
+++++

#' Remove columns citation column

fish.dat <-fish.dat[,-22]

fish.dat$occ<-1

#' Remove rows with no site data

fish.dat<-fish.dat[!fish.dat$Site_Description=="",]

#' Remove rows with no species data

fish.dat<-fish.dat[!fish.dat$Species=="",]

#' Assign column for season based on month

fish.dat$season[fish.dat$AMP_month<9]<-"Spring"

fish.dat$season[fish.dat$AMP_month>8]<-"Fall"

#' Reassign spring to dummy variable == 1, fall == 2

fish.dat$season.dummy[fish.dat$season=="Spring"]<-1

fish.dat$season.dummy[fish.dat$season=="Fall"]<-2

#' Look to see how many unique sites

```

```

length(unique(fish.dat$Site_Description)) #' 25 unique sites

#' Create unique number from 1 - 25 associated with each unique site
fish.dat2 <- transform(fish.dat,site=as.numeric(factor(Site_Description)))

#' Concatenate genus and species and create new 'taxon' column
fish.dat2$taxon<-paste(fish.dat2$Genus,"_",fish.dat2$Species)

#' Take a look at the number of times each species occurs at each site
agg<-aggregate(occ~taxon,data=fish.dat2,sum)

#' I removed species that are extremely rare -- may want a more robust approach to this such as,
removing species

#' that occur at less than x% of sites, or that have less than x number of observations, etc.
sub.ag<-subset(agg,occ>25) # this drops the species count from 65 to 38(MUCH better)

#' This filters the original data to only include the new list of 38 species
fish.dat2<-fish.dat2 %>%
  filter(taxon %in% sub.ag$taxon)

#' Create new column for dummy indicator of species 1-38
fish.dat3<-transform(fish.dat2, dummy.spp=as.numeric(factor(taxon)))

#' Create new column for dummy indicator of year 2005 - 2016

```

```

fish.dat3<-transform(fish.dat3, dummy.year=as.numeric(factor(AMP_year)))

#' Take a look at the unique number of species
length(unique(fish.dat3$taxon)) ##38 species
length(unique(fish.dat3$dummy.year)) ##12 years

melt.dat1<-melt(fish.dat3, id.vars = c("taxon", "dummy.spp", "AMP_year", "dummy.year",
"Site_Description", "site", "season.dummy"), measure.var="occ")

write.csv(melt.dat1,"melt.dat.38spp.csv")

write.csv(list(unique(melt.dat1$taxon)), "list.spps.csv")

#'

+++++
+++++

#' TRUE pres/abs matrix for each site, species, year, and season (i.e., a 4-dimensional matrix)

#'

+++++
+++++

melt.dat1<-melt_dat_38spp ### mary did this

#' Presence matrix for each site~species*year*season

melt.dat1$site<-as.factor(melt.dat1$site)

y=cast(melt.dat1,dummy.spp~site~season.dummy~AMP_year,sum) #TRUE MATRIX

dim(y) #4 dimensions

```

y[y>=2]<-1 #'changes any detection value over 1 to 1, so matrix is only 1's and 0's

ynew<-y

ynew[,1:3,1,1:3]<-NA ### griffin A,B,C summer, 2005-2007 is NA

ynew[,1:3,2,5]<-NA ### griffin A,B,C fall, 2009 is NA

ynew[,2:3,,6:10]<-NA ### griffin B,C summer, fall, 2010-2014 is NA mary changed years to  
6:10

ynew[,1,2,6]<-NA ### griffin A, fall, 2010 is NA

ynew[,1,1,8:10]<-NA ### griffin A, summer, 2012-2014 is NA

ynew[,1:3,2,12]<-NA ### griffin A,B,C fall, 2016 is NA

ynew[,1:3,1,11:12]<-NA ### griffin A,B,C summer, 2015-2016 is NA

ynew[,4:8,1,1]<-NA ### hillabee A,B,C,D,E summer, 2005 is NA

ynew[,6:7,,3]<-NA ### hillabee D,E summer, fall, 2007 is NA mary changed to 6:7, C & D

ynew[,4,2,5]<-NA ### hillabee A,fall, 2009 is NA

ynew[,8,2,5]<-NA ### hillabee E, fall, 2009 is NA

ynew[,6:8,2,7]<-NA ### hillabee D,E, fall, 2007 is NA ## mary changed to 6:8, C,D,E

ynew[,4:8,2,6]<-NA ### hillabee A,B,C,D,E summer, 2005 is NA ## means year 6 2010

ynew[,7:8,,8]<-NA ### hillabee D,E, summer, fall, 2012 is NA

ynew[,8,,8:12]<-NA ### hillabee E, summer, fall, 2012-2016 is NA

ynew[,6:7,2,12]<-NA ### hillabee C,D fall, 2009 is NA

ynew[,10:13,,8]<-NA ### malone B,C,D,E summer, fall, 2005-2007 is NA ### means 2012

ynew[,14:15,1,1:3]<-NA ### peters island D,E summer, 2005-2007 is NA

ynew[,14:15,2,5:6]<-NA ### peters island D,E fall, 2009-2010 is NA

ynew[,15,1,6]<-NA ### peters island E summer, 2010 is NA

ynew[,14:15,2,6]<-NA ### peters island E fall, 2010 is NA

ynew[,15,,7:10]<-NA ### peters island E summer,fall, 2011-2014 is NA

ynew[,14,1,8:10]<-NA ### peters island D summer, 2012-2014 is NA

ynew[,14:15,1,11:12]<-NA ### peters island D,E summer, 2015-2016 is NA

ynew[,16:20,1,1]<-NA ### upper tal A,B,C,D,E summer, 2005 is NA

ynew[,16:18,2,3]<-NA ### upper tal A,B,C, fall, 2007 is NA

ynew[,16:18,,4]<-NA ### upper tal A,B,C summer, fall, 2008 is NA

ynew[,16:18,2,6:7]<-NA ### upper tal A,B,C fall, 2010-2011 is NA

ynew[,16:20,1,8]<-NA ### upper tal A,B,C,D,E summer, 2012 is NA

ynew[,16:18,2,12]<-NA ### upper tal A,B,C fall, 2016 is NA

ynew[,23:25,1,8]<-NA ### wadley C,D,E summer, 2012 is NA

y<-ynew

remove(ynew)

reg<-c(1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,0,0,1,1,1,1,1)

```

annual_covariates <- read.csv("~/Documents/Tallapoosa/annual.covariates.csv")
annual_covariates$deg.days.s<-scale(annual_covariates$cum.deg.days)
annual_covariates$num.gen.s<-scale(annual_covariates$num.gen.events)

# Specify model in BUGS language
sink("Dynocc_multitaxa_reg_unreg.jags")
cat("
  model {

# Specify priors

for (m in 1:ntaxa){
  psi1[m] ~ dunif(0, 1)
  psi1.reg[m] ~ dunif(0, 1)
  for (i in 1:nsite){
    for (k in 1:(nyear-1)){

      logit(phi[m,i,k]) <-beta.phi[m] + beta.phi.reg[m]*reg[i] + epsilon.phi.year[k]+
epsilon.phi.site[i] # phi is persistence

      logit(gamma[m,i,k]) <-beta.gamma[m] + beta.gamma.reg[m]*reg[i]+
epsilon.gamma.year[k] + epsilon.gamma.site[i] # gamma is colonization      # gamma is
colonization

    }}}

```

```
for (m in 1:ntaxa){  
  for (i in 1:nsite){  
    for (j in 1:nrep){  
      for (k in 1:(nyear)){  
        logit(p[m,i,j,k]) <-beta.p[m] + epsilon.p[i,j,k]  
      }  
    }  
  }  
}
```

```
beta.phi.mean~dnorm(0, 0.37)    #intercept for persistence  
beta.gamma.mean~dnorm(0, 0.37) #intercept for colonization  
beta.phi.reg.mean~dnorm(0, 0.37)  
beta.gamma.reg.mean~dnorm(0, 0.37)
```

```
for (m in 1:ntaxa){  
  beta.p[m]~dnorm(0, 0.37)  
  p.sp[m]<-1/(1+exp(-beta.p[m]))  
  beta.phi.reg[m]<-beta.phi.reg.mean + epsilon.phi.reg[m]  
  beta.phi[m]<-beta.phi.mean + epsilon.phi.taxa[m]  
  beta.gamma.reg[m]<-beta.gamma.reg.mean + epsilon.gamma.reg[m]  
  beta.gamma[m]<-beta.gamma.mean + epsilon.gamma.taxa[m]  
}
```

```
for(k in 1:(nyear-1)){
```

```

epsilon.phi.year[k]~dnorm(0, tau.phi.y) ### random effect of year on persistence
epsilon.gamma.year[k]~dnorm(0, tau.gamma.y) ### random effect of year on colonization
}

for (i in 1:nsite){
  epsilon.phi.site[i]~dnorm(0, tau.phi.s) ### random effect of site on persistence
  epsilon.gamma.site[i]~dnorm(0, tau.gamma.s) ### random effect of site on colonization
  for (j in 1:nrep){
    for (k in 1:(nyear)){
      epsilon.p[i,j,k]~dnorm(0, tau.p) ## random effect of survey on overall detection
    }}
}

for (m in 1:ntaxa){
  epsilon.phi.taxa[m]~dnorm(0, tau.phi.taxa) ### random effect of taxon on persistence
  epsilon.gamma.taxa[m]~dnorm(0, tau.gamma.taxa) ### random effect of taxon on
colonization
  epsilon.phi.reg[m]~dnorm(0, tau.phi.reg.taxa)
  epsilon.gamma.reg[m]~dnorm(0, tau.gamma.reg.taxa)
}

sigma.phi.s~dunif(0,10) #random effect for site
tau.phi.s<-pow(sigma.phi.s,-2)
var.phi.s<-pow(sigma.phi.s,2)

```

```
sigma.gamma.s~dunif(0,10)
tau.gamma.s<-pow(sigma.gamma.s,-2)
var.gamma.s<-pow(sigma.gamma.s,2)
sigma.p~dunif(0,10) #random effect for survey on detection
tau.p<-pow(sigma.p,-2)
var.p<-pow(sigma.p,2)
sigma.phi.taxa~dunif(0,10) #random effect for taxon
tau.phi.taxa<-pow(sigma.phi.taxa,-2)
var.phi.taxa<-pow(sigma.phi.taxa,2)
sigma.phi.reg.taxa~dunif(0,10) #random effect for taxon
tau.phi.reg.taxa<-pow(sigma.phi.reg.taxa,-2)
var.phi.reg.taxa<-pow(sigma.phi.reg.taxa,2)
sigma.gamma.taxa~dunif(0,10)
tau.gamma.taxa<-pow(sigma.gamma.taxa,-2)
var.gamma.taxa<-pow(sigma.gamma.taxa,2)
sigma.gamma.reg.taxa~dunif(0,10)
tau.gamma.reg.taxa<-pow(sigma.gamma.reg.taxa,-2)
var.gamma.reg.taxa<-pow(sigma.gamma.reg.taxa,2)
sigma.phi.y~dunif(0,10) #random effect for year
tau.phi.y<-pow(sigma.phi.y,-2)
var.phi.y<-pow(sigma.phi.y,2)
sigma.gamma.y~dunif(0,10)
tau.gamma.y<-pow(sigma.gamma.y,-2)
```

```

var.gamma.y<-pow(sigma.gamma.y,2)

#' ----- END PRIOR SPECIFICATION

for (i in 1:nsite){
  for (k in 1:(nyear-1)){
    ## get year and site-specific rates
    phi.site.yr[i,k]<-1/(1+exp(-beta.phi.mean-beta.phi.reg.mean*reg[i]-epsilon.phi.site[i] -
epsilon.phi.year[k]))
    gamma.site.yr[i,k]<-1/(1+exp(-beta.gamma.mean-beta.gamma.reg.mean*reg[i]-
epsilon.gamma.site[i] - epsilon.gamma.year[k]))
  }
}

for (k in 1:(nyear-1)){
  mean.phi.unreg[k]<-1/(1+exp(-beta.phi.mean-epsilon.phi.year[k]))
  mean.phi.reg[k]<-1/(1+exp(-beta.phi.mean-beta.phi.reg.mean-epsilon.phi.year[k]))
  mean.gamma.unreg[k]<-1/(1+exp(-beta.gamma.mean-epsilon.gamma.year[k]))
  mean.gamma.reg[k]<-1/(1+exp(-beta.gamma.mean-beta.gamma.reg.mean-
epsilon.gamma.year[k]))
}

# Ecological submodel: Define state conditional on parameters

```

```

for (m in 1:ntaxa){
  for (i in 1:nsite){
    z[m,i,1] ~ dbern(psi1[m]*(1-reg[i]) + psi1.reg[m]*reg[i])    #prob of occurrence is constant
across sites, for each species
    for (k in 2:nyear){
      muZ[m,i,k]<- z[m,i,k-1]*phi[m,i,k-1] + (1-z[m,i,k-1])*gamma[m,i,k-1]    #muZ is a
probability of occurrence based on persistence or colonization
      z[m,i,k] ~ dbern(muZ[m,i,k])
    } #k
  } #i
} #m

# Observation model
for (m in 1:ntaxa){
  for (i in 1:nsite){
    for (j in 1:nrep){
      for (k in 1:nyear){
        muy[m,i,j,k] <- z[m,i,k]*p[m,i,j,k]
        y[m,i,j,k] ~ dbern(muy[m,i,j,k])
        ynew[m,i,j,k]~dbern(muy[m,i,j,k]) #simulated observations
      } #k
    } #j
  } #i
}

```

```

} #m

##### stuff for posterior predictive check #####

for (i in 1:nsite){
  for (j in 1:nrep){
    for (k in 1:nyear){

obsrich[i,j,k]<-sum(y[,i,j,k]) ## observed richness by survey

simrich[i,j,k]<-sum(ynew[,i,j,k]) ## simulated richness by survey

exprich[i,j,k]<-sum(muy[,i,j,k]) ## expected richness

depobs[i,j,k]<-pow((pow(obsrich[i,j,k], 0.5)-pow(exprich[i,j,k], 0.5)), 2) # freeman-tukey
measure of departure from expected, observed data

depsim[i,j,k]<-pow((pow(simrich[i,j,k], 0.5)-pow(exprich[i,j,k], 0.5)), 2) # departure from
expected, simulated data

    }

  }

}

fit<- sum(depobs[,,]) #discrepancy, observed data

fit.sim<- sum(depsim[,,]) #discrepancy, simulated data

}

",fill = TRUE)

sink()

# Bundle data

```

```
win.data <- list(y = y, ntaxa=dim(y)[1], nsite = dim(y)[2], nrep = dim(y)[3], nyear = dim(y)[4],  
               reg=reg)
```

```
# Initial values
```

```
ntaxa=38
```

```
nsite=25
```

```
nyear=12
```

```
zst<-array(NA,dim=c(38,25,12))
```

```
for (m in 1:ntaxa){
```

```
  for (i in 1:nsite){
```

```
    for (k in 1:nyear){
```

```
      if(is.na(y[m,i,1,k]) & is.na(y[m,i,2,k]))
```

```
        {zst[m,i,k]<-NA
```

```
        next}
```

```
      zst[m,i,k]<-max(y[m,i,1:2,k],na.rm=T)
```

```
    }
```

```
  }
```

```
}
```

```
inits <- function(){ list(z = zst)}
```

```
params <- c("beta.gamma.mean", "beta.phi.mean", "beta.phi.reg.mean", "beta.gamma.reg.mean",  
"p.sp", "beta.phi.reg", "beta.gamma.reg",  
"var.phi.reg.taxa", "var.gamma.reg.taxa", "phi.site.yr", "gamma.site.yr", "mean.phi.reg",  
"mean.phi.unreg", "mean.gamma.reg", "mean.gamma.unreg",  
"var.phi.taxa", "var.gamma.taxa", "var.p", "var.phi.s", "var.gamma.s", "var.phi.y",  
"var.gamma.y", "fit", "fit.sim",  
"psi1", "psi1.reg")
```

```
#params<-c("obsrich", "exprich", "fit", "fit.sim", "beta.phi.reg.mean", "beta.gamma.reg.mean")
```

```
# MCMC settings
```

```
ni <- 5000
```

```
nt <- 3
```

```
nb <- 500
```

```
nc <- 3
```

```
# Call JAGS from R (BRT 3 min)
```

```
out.tal.reg.unreg.fit<- jags(win.data, inits, params, "Dynocc_multitaxa_reg_unreg.jags", n.chains
```

```
= nc, n.thin = nt, n.iter = ni, n.burnin = nb, working.directory = getwd())
```

```
print(out.tal.reg.unreg.fit, dig = 2)
```

```
#dim(out.tal.reg.unreg.effect.adj$BUGSoutput$summary)
```

```
#out.tal.reg.unreg.effect.adj$BUGSoutput$summary[780:800,] ## did variance terms converge?
```

```

#plot(out.tal.reg.unreg.effect.adj$BUGSoutput$sims.list$fit,
out.tal.reg.unreg.effect.adj$BUGSoutput$sims.list$fit.sim, xlab = "Discrepancy actual data",
ylab = "Discrepancy replicate data", las = 1,
#   bty = "n")
#abline(0, 1, col = "black", lwd = 2)
#mean(out.tal.reg.unreg.effect.adj$BUGSoutput$sims.list$fit.sim >
out.tal.reg.unreg.effect.adj$BUGSoutput$sims.list$fit)

##### species labels
spp.lab<-c("Aari", "Coli", "Ctal", "Ccal", "Cgib", "Cven", "Dpet", "Echu", "Esti", "Etal", "Fbif",
"Foli",
"Gaff", "Hlin", "Heto", "Ipun", "Laur", "Lcya", "Lmac", "Lchr", "Metn", "Mhen", "Mtal",
"Mmel", "Mduq", "Mpoe", "Nbai", "Nsti", "Ntex", "Nfun", "Nlep", "Pkat", "Ppal",
"Psmi", "Pcat",
"Pvig", "Poli", "Satr")

spp.lab.3x3<-c("Amb_ari", "Cam_oli", "Cot_tal", "Cyp_cal", "Cyp_gib", "Cyp_ven",
"Dor_pet", "Eth_chu",
"Eth_sti", "Eth_tal", "Fun_bif", "Fun_oli",
"Gam_aff", "Hyb_lin", "Hyp_eto", "Ict_pun", "Lep_aur", "Lep_cya", "Lep_mac",
"Lux_chr", "Mac_aes", "Mic_hen", "Mic_tal",
"Min_mel", "Mox_duq", "Mox_poe", "Not_bai", "Not_sti", "Not_tex", "Not_fun",
"Not_lep", "Per_kat", "Per_pal", "Per_smi", "Phe_cat",

```

"Pim\_vig", "Pyl\_oli", "Sem\_atr")

spp.lab.long<-c(expression(italic("Ambloplites ariommus")), expression(italic("Campostoma oligolepis")),

expression(italic("Cottus tallapoosae")), expression(italic("Cyprinella callistia")),

expression(italic("Cyprinella gibbsi")), expression(italic("Cyprinella venusta")),

expression(italic("Dorosoma petenense")), expression(italic("Etheostoma chuckwachatte")),

expression(italic("Etheostoma stigmaeum")), expression(italic("Etheostoma tallapoosae")),

expression(italic("Fundulus bifax")), expression(italic("Fundulus olivaceus")),

expression(italic("Gambusia affinis")), expression(italic("Hybopsis lineapunctata")),

expression(italic("Hypentelium etowanum")), expression(italic("Ictalurus punctatus")),

expression(italic("Lepomis auritus")), expression(italic("Lepomis cyanellus")),

expression(italic("Lepomis macrochirus")), expression(italic("Luxilus chrysocephalus")),

expression(italic("Macrhybopsis aestivalis")), expression(italic("Micropterus henshalli")),

expression(italic("Micropterus tallapoosae")), expression(italic("Minytrema melanops")),

expression(italic("Moxostoma duquesnei")), expression(italic("Moxostoma poecilurum")),

expression(italic("Notropis baileyi")), expression(italic("Notropis stilbius")),

```
expression(italic("Notropis texanus")), expression(italic("Noturus funebris")),  
expression(italic("Noturus leptacanthus")), expression(italic("Percina kathae")),  
expression(italic("Percina palmaris")), expression(italic("Percina smithvanizi")),  
expression(italic("Phenacobius catostomus")), expression(italic("Pimephales  
vigilax")),  
expression(italic("Pyloodictis olivaris")), expression(italic("Semotilus atromaculatus")))
```

##### plot species-specific flow-regulation effects on phi & gamma: Figures B6 and B7

```
gamma.reg<-out.tal.reg.unreg.fit$BUGSoutput$mean$beta.gamma.reg
```

```
phi.reg<-out.tal.reg.unreg.fit$BUGSoutput$mean$beta.phi.reg
```

```
gamma.lc<-gamma.uc<-phi.lc<-phi.uc<-numeric()
```

```
for (m in 1:ntaxa){
```

```
  gamma.lc[m]<-quantile(out.tal.reg.unreg.fit$BUGSoutput$sims.list$beta.gamma.reg[,m],  
0.025)
```

```
  gamma.uc[m]<-quantile(out.tal.reg.unreg.fit$BUGSoutput$sims.list$beta.gamma.reg[,m],  
0.975)
```

```
  phi.lc[m]<-quantile(out.tal.reg.unreg.fit$BUGSoutput$sims.list$beta.phi.reg[,m], 0.025)
```

```
  phi.uc[m]<-quantile(out.tal.reg.unreg.fit$BUGSoutput$sims.list$beta.phi.reg[,m], 0.975)
```

```
}
```

```
spprates<-as.data.frame(cbind(gamma.reg,gamma.lc,gamma.uc,phi.reg,phi.lc,phi.uc))
```

```
spprates$labels<-spp.lab
```

```
sppphi<-spprates[order(phi.reg),]
```

```
sppgamma<-spprates[order(gamma.reg),]
```

```
#####
```

```
####
```

```
#### Figure B6 effect of being in the regulated reach on persistence probability, by species (logit scale)
```

```
plot(1:38, sppphi$phi.reg, type="p", pch=16, ylim=c(-10,10),
```

```
family="serif",
```

```
ylab="Flow regulation effect, persistence",
```

```
xlab="Species", bty="n", cex=1, axes=FALSE)
```

```
axis(1, at = 1:38, family="serif",
```

```
labels=sppphi$labels, adj=1, xpd = TRUE, tcl=-0.25, las=2)
```

```
axis(2, at=seq(-10, 10, 2), family="serif",
```

```
labels=c("-10","-8", "-6", "-4", "-2", "0", "2","4","6","8","10"))
```

```
abline(0,0)
```

```
segments((1:38), sppphi$phi.lc, (1:38), sppphi$phi.uc, col="gray")
```

```
# Fig. B6 using spp.lab.3x3
```

```
spprates$labels<-spp.lab.3x3
```

```
sppphi<-spprates[order(phi.reg),]
```

```

sppgamma<-spprates[order(gamma.reg),]

par(mar=c(6, 4.1, 4.1, 2.1)) # set margins (c(bottom, left, top, right))

plot(1:38, sppphi$phi.reg, type="p", pch=16, ylim=c(-10,10),
     family="serif",
     ylab="Flow regulation effect, persistence",xlab="",
     bty="n", cex=1, cex.lab = 1.2, axes=FALSE)

axis(1, at=1:38, tcl=-0.25, labels = FALSE)
axis(2, at=seq(-10, 10, 2), family="serif",
     labels=c("-10","-8", "-6", "-4", "-2", "0", "2","4","6","8","10"))
text(x = seq(1.5, 38.5, by=1), par("usr")[3] - 0.65,
     srt = 60, pos = 2, xpd = TRUE, adj=c(1,1) ,
     labels = sppphi$labels, family="serif")

# add xlabel, "line" argument controls vertical position
mtext("Species", side = 1, line = 4.5, cex = 1.2, family = "serif")

abline(0,0)

segments((1:38), sppphi$phi.lc, (1:38), sppphi$phi.uc, col="gray")

# save at dimension 1200 width

```

```

### Fig. B6 Using spp.lab.long

par(mar = c(12, 4.1, 4.1, 2.1)) # set margins (c(bottom, left, top, right))

spprates$labels<-spp.lab.long

sppphi<-spprates[order(phi.reg),]

sppgamma<-spprates[order(gamma.reg),]

plot(1:38, sppphi$phi.reg, type="p", pch=16, ylim=c(-10,10),
     family="serif",
     ylab="Flow regulation effect, persistence",
     bty="n", cex=1.2, axes=FALSE)

axis(1, at=seq(1, 38, by=1), tcl=-0.25, labels = FALSE)
text(x = seq(1.5, 38.5, by=1), par("usr")[3] - 0.25,
     srt = 60, pos = 2, xpd = TRUE, adj=c(1,1) ,
     labels = sppphi$labels)

# add xlabel, "line" argument controls vertical position
mtext("Species", side = 1, line = 6, cex = 1.2, family = "serif")

par(mar=c(5.1, 4.1, 4.1, 2.1)) # default plot margins

```

```
#####
```

```
####
```

```
#####
```

```
####
```

```
#### Figure B7 effect of being in the regulated reach on colonization probability, by species
```

```
(logit scale)
```

```
plot(1:38, sppgamma$gamma.reg, type="p", pch=16, ylim=c(-7,4), ylab="Flow regulation effect,  
colonization",
```

```
  xlab="Species", bty="n", cex=1, axes=FALSE)
```

```
axis(1, at = 1:38, labels=sppgamma$labels, tcl=-0.25, las=2)
```

```
axis(2, at=seq(-6, 4, 2), labels=c( "-6", "-4", "-2", "0", "2","4"))
```

```
abline(0,0)
```

```
segments((1:38), sppgamma$gamma.lc, (1:38), sppgamma$gamma.uc, col="gray")
```

```
## Fig. B7 using spp.lab.3x3
```

```
par(mar=c(6, 4.1, 4.1, 2.1)) # set margins (c(bottom, left, top, right))
```

```
plot(1:38, sppgamma$gamma.reg, type="p", pch=16, ylim=c(-7,4),
```

```
  family="serif",
```

```

ylab="Flow regulation effect, colonization",xlab="",
bty="n", cex=1, cex.lab = 1.2, axes=FALSE)

axis(1, at=1:38, tcl=-0.25, labels = FALSE)
axis(2, at=seq(-6, 4, 2), family="serif",
      labels=c("-6", "-4", "-2", "0", "2", "4"))
text(x = seq(1.5, 38.5, by=1), par("usr")[3] - 0.65,
      srt = 60, pos = 2, xpd = TRUE, adj=c(1,1) ,
      labels = sppphi$labels, family="serif")

# add xlabel, "line" argument controls vertical position
mtext("Species", side = 1, line = 4.5, cex = 1.2, family = "serif")

abline(0,0)

segments((1:38), sppgamma$gamma.lc, (1:38), sppgamma$gamma.uc, col="gray")

# save at dimension 1200 width

par(mar=c(5.1, 4.1, 4.1, 2.1)) # default plot margins

```

```
#####
```

```
####
```

```
##### plot phi and gamma across years, phi.site.year & gamma.site.year_Figures B4 and B5
```

```
nsites<-25
```

```
#### colonization - figure B5
```

```
par(mar=c(5, 4, 4, 2) + 0.1) # restore plot parameters to default
```

```
par(xpd = T, mar = par()$mar + c(0,0,0,9)) # set parameters to fit a legend..
```

```
plot(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$gamma.site.yr[1,],
```

```
type="b", pch=16, ylim=c(0,0.6), ylab="Colonization probability", xlab="Year",
```

```
family = "serif", bty="n", cex=1, axes=FALSE)
```

```
axis(1, at = 1:11, labels=rep(NA, 11), tcl=-0.25)
```

```
axis(1, at=seq(1,11,2), labels=c("06", "08", "10", "12", "14", "16"), family="serif")
```

```
axis(2, at=seq(0, 0.6, 0.2), labels=c("0", "0.2", "0.4", "0.6"), family="serif")
```

```
for (i in 2:3){
```

```
points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$gamma.site.yr[i,],type="b",
```

```
pch=16)
```

```
}
```

```
for (i in 4:8){
```

```
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$gamma.site.yr[i,],type="b",
pch=16, col="blue")
}
for (i in 9:13){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$gamma.site.yr[i,],type="b",
pch=16, col="red")
}
for (i in 14:15){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$gamma.site.yr[i,],type="b",
pch=16, col="grey")
}
for (i in 16:20){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$gamma.site.yr[i,],type="b",
pch=16, col="green")
}
for (i in 21:25){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$gamma.site.yr[i,],type="b",
pch=16, col="orange")
}

lines(1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$mean.gamma.unreg, type="l",
lwd=2, lty=2)
```

```

lines(1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$mean.gamma.reg, type="l",
lwd=2)

# add legend
par(family = "serif")
legend(11.5, 0.45, title = "EXPLANATION",
      c("Heflin", "Hillabee", "Malone", "Wadley", "Horseshoe Bend", "Peters Island \n
(Horseshoe Bend)"),
      col=c("green", "blue", "red", "orange", "black", "grey"),
      lty=1, cex=0.8)

# saved at 900 X 550 dimensions

par(mar=c(5, 4, 4, 2) + 0.1) # restore plot parameters to default
#####

##### persistence - Figure B4

par(mar=c(5, 4, 4, 2) + 0.1) # restore plot parameters to default
par(xpd = T, mar = par()$mar + c(0,0,0,9)) # set parameters to fit a legend..

plot(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$phi.site.yr[1,],
      family="serif", type="b", pch=16, ylim=c(0.4,1), ylab="Persistence probability",
      xlab="Year", bty="n", cex=1, axes=FALSE)
axis(1, at = 1:11, labels=rep(NA, 11), tcl=-0.25)
axis(1, at=seq(1,11,2), labels=c("06", "08", "10", "12", "14", "16"), family="serif")

```

```
axis(2, at=seq(0.4, 1, 0.2), labels=c("0.4", "0.6", "0.8", "1.0"), family="serif")
##segments((1:(nyear-1)), lower.phi, (1:(nyear-1)), upper.phi, col="gray")
for (i in 2:3){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$phi.site.yr[i,],type="b",
  pch=16)
}
for (i in 4:8){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$phi.site.yr[i,],type="b",
  pch=16, col="blue")
}
for (i in 9:13){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$phi.site.yr[i,],type="b",
  pch=16, col="red")
}
for (i in 14:15){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$phi.site.yr[i,],type="b",
  pch=16, col="grey")
}
for (i in 16:20){
  points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$phi.site.yr[i,],type="b",
  pch=16, col="green")
}
for (i in 21:25){
```

```

points(x=1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$phi.site.yr[i,],type="b",
pch=16, col="orange")
}
lines(1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$mean.phi.unreg, type="l",
lwd=2,lty=2)
lines(1:(nyear-1), y=out.tal.reg.unreg.fit$BUGSoutput$mean$mean.phi.reg, type="l", lwd=2)

# add legend
par(family = "serif")
legend(11.5, 0.85, title = "EXPLANATION",
      c("Heflin", "Hillabee", "Malone", "Wadley", "Horseshoe Bend", "Peters Island \n
(Horseshoe Bend)"),
      col=c("green", "blue", "red", "orange", "black", "grey"),
      lty=1, cex=0.8)
# saves at 950 X 650 dimensions

par(mar=c(5, 4, 4, 2) + 0.1) # restore plot parameters to default

#out.tal.reg.unreg.fit$BUGSoutput$mean$beta.gamma.reg.mean

```

```

#out.tal.reg.unreg.fit$BUGSoutput$mean$beta.phi.reg.mean

##### detection

ntaxa<-38

lower.phi.re<-upper.phi.re<-lower.gamma.re<-upper.gamma.re<-lower.p<-upper.p<-numeric()

for (m in 1:ntaxa){

  lower.p[m]<-quantile(out.tal.reg.unreg.fit$BUGSoutput$sims.list$p.sp[,m], 0.025)

  upper.p[m]<-quantile(out.tal.reg.unreg.fit$BUGSoutput$sims.list$p.sp[,m], 0.975)

}

p.sp.mean<-out.tal.reg.unreg.fit$BUGSoutput$mean$p.sp

detect.set<-as.data.frame(cbind(p.sp.mean, lower.p, upper.p))

detect.set<-cbind(spp.lab, detect.set)

detect.sorted<-detect.set[order(p.sp.mean),]

plot(x=1:ntaxa, y=detect.sorted$p.sp.mean, type="p", pch=16, ylim=c(0,1), ylab="Detection
probability",

      xlab="Taxon", bty="n", cex=1.5, axes=FALSE)

axis(2, at=seq(0, 1, 0.2), labels=c("0", "0.2", "0.4", "0.6", "0.8", "1.0"))

axis(1, at=1:38, labels=detect.sorted$spp.lab, las=2)

segments(1:ntaxa, detect.sorted$lower.p, 1:ntaxa, detect.sorted$upper.p, col="gray")

```

Objective 2. compare sites and relations to hydrothermal variables below the dam

```
fish.dat<-read.csv("2005_2016_AMP.csv")
```

```
library(reshape)
```

```
library(reshape2)
```

```
library(abind) #<--- is this the package for abind function line130?
```

```
library(tidyverse)
```

```
library(R2jags)
```

```
library(R2WinBUGS)
```

```
head(fish.dat)
```

```
# eto<-read.csv("Eto.28spp.10sites.presence.csv")
```

```
#
```

```
# eto<-Eto_28spp_10sites_presence
```

```
# eto$Site<-as.factor(eto$Site)
```

```
# library("reshape2")
```

```
# alldetections10<-melt(eto)
```

```
# alldetarray10<-array(NA, dim=c(28,10,2,15))
```

```
# alldetarray10[,,]<-as.array(alldetections10$value)
```

```
# alldetections10$value
```

```
#'
```

```
+++++
```

```
+++++
```

```
#' Format data for occupancy model
```

```

#'
+++++
+++++

#' Remove columns citation column

fish.dat <- fish.dat[,-22]

fish.dat$occ <- -1

#' Remove rows with no site data

fish.dat <- fish.dat[!fish.dat$Site_Description=="",]

#' Remove rows with no species data

fish.dat <- fish.dat[!fish.dat$Species=="",]

#' Assign column for season based on month

fish.dat$season[fish.dat$AMP_month < 9] <- "Spring"
fish.dat$season[fish.dat$AMP_month > 8] <- "Fall"

#' Reassign spring to dummy variable == 1, fall == 2

fish.dat$season.dummy[fish.dat$season=="Spring"] <- 1
fish.dat$season.dummy[fish.dat$season=="Fall"] <- 2

#' Look to see how many unique sites

length(unique(fish.dat$Site_Description)) #' 25 unique sites

#' Create unique number from 1 - 25 associated with each unique site

fish.dat2 <- transform(fish.dat, site=as.numeric(factor(Site_Description)))

```

```

#' Concatenate genus and species and create new 'taxon' column
fish.dat2$taxon<-paste(fish.dat2$Genus,"_",fish.dat2$Species)

#' Take a look at the number of times each species occurs at each site
agg<-aggregate(occ~taxon,data=fish.dat2,sum)

#' I removed species that are extremely rare -- may want a more robust approach to this such as,
removing species

#' that occur at less than x% of sites, or that have less than x number of observations, etc.
sub.ag<-subset(agg,occ>25) # this drops the species count from 65 to 38(MUCH better)

#' This filters the original data to only include the new list of 38 species
fish.dat2<-fish.dat2 %>%
  filter(taxon %in% sub.ag$taxon)

#' Create new column for dummy indicator of species 1-38
fish.dat3<-transform(fish.dat2, dummy.spp=as.numeric(factor(taxon)))

#' Create new column for dummy indicator of year 2005 - 2016
fish.dat3<-transform(fish.dat3, dummy.year=as.numeric(factor(AMP_year)))

#' Take a look at the unique number of species
length(unique(fish.dat3$taxon)) ##38 species

```

```
length(unique(fish.dat3$dummy.year)) ##12 years
```

```
melt.dat1<-melt(fish.dat3, id.vars = c("taxon", "dummy.spp", "AMP_year","dummy.year",  
"Site_Description", "site", "season.dummy"), measure.var="occ")
```

```
write.csv(melt.dat1,"melt.dat.38spp.csv")
```

```
write.csv(list(unique(melt.dat1$taxon)), "list.spps.csv")
```

```
#'
```

```
+++++
```

```
+++++
```

```
#' TRUE pres/abs matrix for each site, species, year, and season (i.e., a 4-dimensional matrix)
```

```
#'
```

```
+++++
```

```
+++++
```

```
melt.dat1<-melt_dat_38spp ### mary did this
```

```
#' Presence matrix for each site~species*year*season
```

```
melt.dat1$site<-as.factor(melt.dat1$site)
```

```
y=cast(melt.dat1,dummy.spp~site~season.dummy~AMP_year,sum) #TRUE MATRIX
```

```
dim(y) #4 dimensions
```

```
y[y>=2]<-1 #'changes any detection value over 1 to 1, so matrix is only 1's and 0's
```

```
ynew<-y
```

ynew[,1:3,1,1:3]<-NA ### griffin A,B,C summer, 2005-2007 is NA

ynew[,1:3,2,5]<-NA ### griffin A,B,C fall, 2009 is NA

ynew[,2:3,,6:10]<-NA ### griffin B,C summer, fall, 2010-2014 is NA mary changed years to 6:10

ynew[,1,2,6]<-NA ### griffin A, fall, 2010 is NA

ynew[,1,1,8:10]<-NA ### griffin A, summer, 2012-2014 is NA

ynew[,1:3,2,12]<-NA ### griffin A,B,C fall, 2016 is NA

ynew[,1:3,1,11:12]<-NA ### griffin A,B,C summer, 2015-2016 is NA

ynew[,4:8,1,1]<-NA ### hillabee A,B,C,D,E summer, 2005 is NA

ynew[,6:7,,3]<-NA ### hillabee D,E summer, fall, 2007 is NA mary changed to 6:7, C & D

ynew[,4,2,5]<-NA ### hillabee A,fall, 2009 is NA

ynew[,8,2,5]<-NA ### hillabee E, fall, 2009 is NA

ynew[,6:8,2,7]<-NA ### hillabee D,E, fall, 2007 is NA ## mary changed to 6:8, C,D,E

ynew[,4:8,2,6]<-NA ### hillabee A,B,C,D,E summer, 2005 is NA ## means year 6 2010

ynew[,7:8,,8]<-NA ### hillabee D,E, summer, fall, 2012 is NA

ynew[,8,,8:12]<-NA ### hillabee E, summer, fall, 2012-2016 is NA

ynew[,6:7,2,12]<-NA ### hillabee C,D fall, 2009 is NA

ynew[,10:13,,8]<-NA ### malone B,C,D,E summer, fall, 2005-2007 is NA ### means 2012

ynew[,14:15,1,1:3]<-NA ### peters island D,E summer, 2005-2007 is NA

ynew[,14:15,2,5:6]<-NA ### peters island D,E fall, 2009-2010 is NA

```
ynew[,15,1,6]<-NA ### peters island E summer, 2010 is NA
ynew[,14:15,2,6]<-NA ### peters island E fall, 2010 is NA
ynew[,15,,7:10]<-NA ### peters island E summer,fall, 2011-2014 is NA
ynew[,14,1,8:10]<-NA ### peters island D summer, 2012-2014 is NA
ynew[,14:15,1,11:12]<-NA ### peters island D,E summer, 2015-2016 is NA
```

```
ynew[,16:20,1,1]<-NA ### upper tal A,B,C,D,E summer, 2005 is NA
ynew[,16:18,2,3]<-NA ### upper tal A,B,C, fall, 2007 is NA
ynew[,16:18,,4]<-NA ### upper tal A,B,C summer, fall, 2008 is NA
ynew[,16:18,2,6:7]<-NA ### upper tal A,B,C fall, 2010-2011 is NA
ynew[,16:20,1,8]<-NA ### upper tal A,B,C,D,E summer, 2012 is NA
ynew[,16:18,2,12]<-NA ### upper tal A,B,C fall, 2016 is NA
```

```
ynew[,23:25,1,8]<-NA ### wadley C,D,E summer, 2012 is NA
```

```
y<-ynew
```

```
remove(ynew)
```

```
##### take out site 10, rep 1, 2011 = year 7
```

```
#y.adj<-y          only use with original melt_dat_38spp file
```

```
#y.adj[,10,1,7]<-NA
```

```
y.reg<-y
```

```

y.reg.a<-y.reg[,1:3,,]
y.reg.b<-y.reg[,9:15,,]
y.reg.c<-y.reg[,21:25,,]
y.regs<-abind(y.reg.a, y.reg.b, along=2)
y.regsites<-abind(y.regs, y.reg.c, along=2) ### 15 flow-regulated sites only
remove(y.reg, y.reg.a, y.reg.b, y.reg.c, y.regs)
#' y = matrix used in JAGS model for occupancy
#'
#' Define variables for JAGS model

annual_covariates <- read_csv("~/Documents/Tallapoosa/annual.covariates.csv")
### scale covariates after removing 2005
newcovariates<-annual_covariates[2:12,]
newcovariates$deg.days.s<-scale(newcovariates$cum.deg.days)
newcovariates$num.gen.s<-scale(newcovariates$num.gen.events)
distance.ds<-c(rep(61.01,3), 2.56, 8.42, 9.3, 9.68, 9.82, 70.23, 70.23, 11.98, 12.16, 16.06, 16.65,
19.19)
distance.ds.s<-as.numeric(scale(distance.ds))
deg.days.s<-as.numeric(newcovariates$deg.days.s) #2006-2016
num.gen.s<-as.numeric(newcovariates$num.gen.s)

```

```

##### figure 2 #####

plot(num.gen.s, deg.days.s, family="serif",
     xlab="number of generation events (scaled)",
     ylab="number of degress days (scaled)"
)

# saved to dimensions 700x500 for publication

#####

# Specify model in BUGS language

sink("Dynocc_multitaxa_reg_covar.jags")

cat("

model {

# Specify priors

for (m in 1:ntaxa){
  psi1[m] ~ dunif(0, 1)
  for (i in 1:nsite){
    for (k in 1:(nyear-1)){
      logit(phi[m,i,k]) <-beta.phi[m] + beta.phi.distance*distance.ds.s[i] +
        beta.phi.temp*deg.days.s[k] + beta.phi.gen*num.gen.s[k] + epsilon.phi.year[k]+
epsilon.phi.site[i] # phi is persistence

      logit(gamma[m,i,k]) <-beta.gamma[m] + beta.gamma.distance*distance.ds.s[i] +

```

```
beta.gamma.temp*deg.days.s[k] + beta.gamma.gen*num.gen.s[k] +  
epsilon.gamma.year[k] + epsilon.gamma.site[i] # gamma is colonization  
}}}
```

```
for (m in 1:ntaxa){  
for (i in 1:nsite){  
for (j in 1:nrep){  
for (k in 1:(nyear)){  
logit(p[m,i,j,k]) <-beta.p[m] + epsilon.p[i,j,k]  
}}}}}
```

```
beta.phi.mean~dnorm(0, 0.37) #intercept for persistence  
beta.gamma.mean~dnorm(0, 0.37) #intercept for colonization  
beta.phi.distance~dnorm(0, 0.37)  
beta.gamma.distance~dnorm(0, 0.37)  
beta.phi.temp~dnorm(0, 0.37)  
#beta.phi.temp.int~dnorm(0, 0.37)  
beta.gamma.temp~dnorm(0, 0.37)  
#beta.gamma.temp.int~dnorm(0, 0.37)  
beta.phi.gen~dnorm(0, 0.37)  
#beta.phi.gen.int~dnorm(0, 0.37)  
beta.gamma.gen~dnorm(0, 0.37)  
#beta.gamma.gen.int~dnorm(0, 0.37)
```

```

for (m in 1:ntaxa){
beta.p[m]~dnorm(0, 0.37)
p.sp[m]<-1/(1+exp(-beta.p[m]))
beta.phi[m]<-beta.phi.mean + epsilon.phi.taxa[m]
beta.gamma[m]<-beta.gamma.mean + epsilon.gamma.taxa[m]
}

for(k in 1:(nyear-1)){
epsilon.phi.year[k]~dnorm(0, tau.phi.y) ### random effect of year on persistence
epsilon.gamma.year[k]~dnorm(0, tau.gamma.y) ### random effect of year on colonization
}

for (i in 1:nsite){
epsilon.phi.site[i]~dnorm(0, tau.phi.s) ### random effect of site on persistence
epsilon.gamma.site[i]~dnorm(0, tau.gamma.s) ### random effect of site on colonization
for (j in 1:nrep){
for (k in 1:(nyear)){
epsilon.p[i,j,k]~dnorm(0, tau.p) ## random effect of survey on overall detection
}}
}}}

for (m in 1:ntaxa){
epsilon.phi.taxa[m]~dnorm(0, tau.phi.taxa) ### random effect of taxon on persistence

```

```
epsilon.gamma.taxa[m]~dnorm(0, tau.gamma.taxa) ### random effect of taxon on  
colonization
```

```
}
```

```
sigma.phi.s~dunif(0,10) #random effect for site
```

```
tau.phi.s<-pow(sigma.phi.s,-2)
```

```
var.phi.s<-pow(sigma.phi.s,2)
```

```
sigma.gamma.s~dunif(0,10)
```

```
tau.gamma.s<-pow(sigma.gamma.s,-2)
```

```
var.gamma.s<-pow(sigma.gamma.s,2)
```

```
sigma.p~dunif(0,10) #random effect for survey on detection
```

```
tau.p<-pow(sigma.p,-2)
```

```
var.p<-pow(sigma.p,2)
```

```
sigma.phi.taxa~dunif(0,10) #random effect for taxon
```

```
tau.phi.taxa<-pow(sigma.phi.taxa,-2)
```

```
var.phi.taxa<-pow(sigma.phi.taxa,2)
```

```
sigma.gamma.taxa~dunif(0,10)
```

```
tau.gamma.taxa<-pow(sigma.gamma.taxa,-2)
```

```
var.gamma.taxa<-pow(sigma.gamma.taxa,2)
```

```
sigma.phi.y~dunif(0,10) #random effect for year
```

```
tau.phi.y<-pow(sigma.phi.y,-2)
```

```
var.phi.y<-pow(sigma.phi.y,2)
```

```
sigma.gamma.y~dunif(0,10)
```

```

tau.gamma.y<-pow(sigma.gamma.y,-2)

var.gamma.y<-pow(sigma.gamma.y,2)

#' ----- END PRIOR SPECIFICATION

for (i in 1:nsite){
  for (k in 1:(nyear-1)){
    ## get year and site-specific rates

    phi.site.yr[i,k]<-1/(1+exp(-beta.phi.mean-beta.phi.distance*distance.ds.s[i] -
beta.phi.temp*deg.days.s[k]-beta.phi.gen*num.gen.s[k]
    -epsilon.phi.site[i] - epsilon.phi.year[k]))

    gamma.site.yr[i,k]<-1/(1+exp(-beta.gamma.mean-beta.gamma.distance*distance.ds.s[i]-
    beta.gamma.temp*deg.days.s[k] - beta.gamma.gen*num.gen.s[k]
    -epsilon.gamma.site[i] - epsilon.gamma.year[k]))

  }
}

# Ecological submodel: Define state conditional on parameters

for (m in 1:ntaxa){
  for (i in 1:nsite){
    z[m,i,1] ~ dbern(psi1[m])    #prob of occurrence is constant across sites, for each species

```

```

    for (k in 2:nyear){
      muZ[m,i,k]<- z[m,i,k-1]*phi[m,i,k-1] + (1-z[m,i,k-1])*gamma[m,i,k-1] #muZ is a
probability of occurrence based on persistence or colonization
      z[m,i,k] ~ dbern(muZ[m,i,k])
    } #k
  } #i
} #m

# Observation model
for (m in 1:ntaxa){
  for (i in 1:nsite){
    for (j in 1:nrep){
      for (k in 1:nyear){
        muy[m,i,j,k] <- z[m,i,k]*p[m,i,j,k]
        y[m,i,j,k] ~ dbern(muy[m,i,j,k])
        ynew[m,i,j,k]~dbern(muy[m,i,j,k]) #simulated observations
      } #k
    } #j
  } #i
} #m

##### stuff for posterior predictive check #####
for (i in 1:nsite){
  for (j in 1:nrep){

```

```

    for (k in 1:nyear){
obsrich[i,j,k]<-sum(y[,i,j,k]) ## observed richness by survey
simrich[i,j,k]<-sum(ynew[,i,j,k]) ## simulated richness by survey
exprich[i,j,k]<-sum(muy[,i,j,k]) ## expected richness
depobs[i,j,k]<-pow((pow(obsrich[i,j,k], 0.5)-pow(exprich[i,j,k], 0.5)), 2) # freeman-tukey
measure of departure from expected, observed data

    depsim[i,j,k]<-pow((pow(simrich[i,j,k], 0.5)-pow(exprich[i,j,k], 0.5)), 2) # departure from
expected, simulated data
    }
}
}

fit<- sum(depobs[,,]) #discrepancy, observed data
fit.sim<- sum(depsim[,,]) #discrepancy, simulated data

}

",fill = TRUE)

sink()

# Bundle data

win.data <- list(y = y.regsites, ntaxa=dim(y.regsites)[1], nsite = dim(y.regsites)[2], nrep =
dim(y.regsites)[3], nyear = dim(y.regsites)[4],

    deg.days.s=deg.days.s, num.gen.s=num.gen.s, distance.ds.s=distance.ds.s)

```

```

# Initial values

ntaxa=38

nsite=15

nyear=12

zst<-array(NA,dim=c(38,15,12))

for (m in 1:ntaxa){

  for (i in 1:nsite){

    for (k in 1:nyear){

      if(is.na(y.regsites[m,i,1,k]) & is.na(y.regsites[m,i,2,k]))

        {zst[m,i,k]<-NA

          next}

      zst[m,i,k]<-max(y.regsites[m,i,1:2,k],na.rm=T)

    }

  }

}

inits <- function(){ list(z = zst)}

params <- c("beta.gamma.mean", "beta.phi.mean", "beta.phi.distance", "beta.gamma.distance",

"p.sp", "beta.phi.temp", "beta.gamma.temp",

"phi.site.yr", "gamma.site.yr", "beta.phi.gen", "beta.gamma.gen",

"var.phi.taxa", "var.gamma.taxa", "var.p", "var.phi.s", "var.gamma.s", "var.phi.y",

"var.gamma.y", "fit", "fit.sim")

```

```

# MCMC settings

ni <- 5000

nt <- 3

nb <- 500

nc <- 3

# Call JAGS from R (BRT 3 min)

out.tal.reg<- jags(win.data, inits, params, "Dynocc_multitaxa_reg_covar.jags", n.chains = nc,
n.thin = nt, n.iter = ni, n.burnin = nb, working.directory = getwd())

print(out.tal.reg, dig = 2)

dim(out.tal.reg$BUGSoutput$summary)

out.tal.reg$BUGSoutput$summary[370:386,] ## did variance terms converge?

plot(out.tal.reg$BUGSoutput$sims.list$fit, out.tal.reg$BUGSoutput$sims.list$fit.sim, xlab =
"Discrepancy actual data", ylab = "Discrepancy replicate data", las = 1,
      bty = "n")

abline(0, 1, col = "black", lwd = 2)

mean(out.tal.reg$BUGSoutput$sims.list$fit.sim > out.tal.reg$BUGSoutput$sims.list$fit)

##### plot phi and gamma across years, phi.site.year & gamma.site.year

nsites<-15

```

```
#####

#### colonization figure 10

par(mar=c(5, 4, 4, 2) + 0.1) # restore plot parameters to default

par(xpd = T, mar = par()$mar + c(0,0,0,9)) # set parameters to fit a legend..

plot(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[1,], type="b", pch=16,
     family="serif", ylim=c(0,0.6), ylab="Colonization probability",
     xlab="Year", bty="n", cex=1, axes=FALSE)

axis(1, at = 1:11, labels=rep(NA, 11), tcl=-0.25)

axis(1, at=seq(1,11,2), labels=c("06", "08", "10", "12", "14", "16"), family="serif")

axis(2, at=seq(0, 0.6, 0.2), labels=c("0", "0.2", "0.4", "0.6"), family="serif")

#segments((1:(nyear-1)), lower.gamma, (1:(nyear-1)), upper.gamma, col="gray")

for (i in 2:3){

  points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="b", pch=16)

}

for (i in 4:8){

  points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="b", pch=16,
        col="red")

}

for (i in 9:10){
```

```

points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="b", pch=16,
col="grey")
}
for (i in 11:15){
points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="b", pch=16,
col="orange")
}
# add legend
par(family = "serif")
legend(11.5, 0.45, title = "EXPLANATION",
c("Malone", "Wadley", "Horseshoe Bend", "Peters Island \n (Horseshoe Bend)"),
col=c("red", "orange", "black", "grey"),
lty=1, cex=0.8)
# saved at 950x610 dimensions

```

```
##### persistence figure 9
```

```

plot(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$phi.site.yr[1,],
family="serif", type="b", pch=16, ylim=c(0.4,1), ylab="Persistence probability",

```

```

    xlab="Year", bty="n", cex=1, axes=FALSE)
axis(1, at = 1:11, labels=rep(NA, 11), tcl=-0.25)
axis(1, at=seq(1,11,2), labels=c("06", "08", "10", "12", "14", "16"))
axis(2, at=seq(0.4, 1, 0.2), labels=c("0.4", "0.6", "0.8", "1.0"))
##segments((1:(nyear-1)), lower.phi, (1:(nyear-1)), upper.phi, col="gray")
for (i in 2:3){
  points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="b", pch=16)
}
for (i in 4:8){
  points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="b", pch=16,
col="red")
}
for (i in 9:10){
  points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="b", pch=16,
col="grey")
}
for (i in 11:15){
  points(x=1:(nyear-1), y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="b", pch=16,
col="orange")
}
# add legend
par(family = "serif")
legend(11.5, 0.82, title = "EXPLANATION",

```

```

c("Malone", "Wadley", "Horseshoe Bend", "Peters Island \n (Horseshoe Bend)",
col=c("red", "orange", "black", "grey"),
lty=1, cex=0.8)
# saved at 950x610 dimensions

### figure 11
plot(num.gen.s, out.tal.reg$BUGSoutput$mean$phi.site.yr[1,],
family="serif", type="p", pch=16,ylim=c(0.4,1), ylab="Persistence probability",
#xlab="# annual generation events (standardized)",
bty="n", cex=1, xlab= "Number of annual generation events (standardized)" )
for (i in 2:3){
points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16)
}
for (i in 4:8){
points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16,
col="red")
}
for (i in 9:10){

```

```

points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16,
col="grey")
}
for (i in 11:15){
points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16,
col="orange")
}
par(family = "serif")
legend(2.15, 0.8, title = "EXPLANATION",
c("Malone", "Wadley", "Horseshoe Bend", "Peters Island \n (Horseshoe Bend)",
col=c("red", "orange", "black", "grey"),
lty=1, cex=0.8)
#saved at 950x610 dimensions

```

```

### figure 13

```

```

plot(deg.days.s, out.tal.reg$BUGSoutput$mean$phi.site.yr[1,],
family="serif", type="p", pch=16,ylim=c(0.4,1), ylab="Persistence probability",
xlab="Number of annual days meeting temperature criterion (standardized)",
bty="n", cex=1 )
for (i in 2:3){

```

```
points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16)
}
for (i in 4:8){
  points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16,
col="red")
}
for (i in 9:10){
  points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16,
col="grey")
}
for (i in 11:15){
  points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$phi.site.yr[i,],type="p", pch=16,
col="orange")
}
par(family = "serif")
legend(1.15, 0.8, title = "EXPLANATION",
      c("Malone", "Wadley", "Horseshoe Bend", "Peters Island \n (Horseshoe Bend)",
col=c("red", "orange", "black", "grey"),
lty=1, cex=0.8)
```

```
### figure 14
```

```
plot(deg.days.s, out.tal.reg$BUGSoutput$mean$gamma.site.yr[1,],
     family="serif", type="p", pch=16,ylim=c(0,0.5), ylab="Colonization probability",
     xlab="Number of annual days meeting temperature criterion (standardized)", bty="n", cex=1
)
for (i in 2:3){
  points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="p", pch=16)
}
for (i in 4:8){
  points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="p", pch=16,
col="red")
}
for (i in 9:10){
  points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="p", pch=16,
col="grey")
}
for (i in 11:15){
  points(x=deg.days.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="p", pch=16,
col="orange")
}

par(family = "serif")
legend(1.15, 0.35, title = "EXPLANATION",
```

```
c("Malone", "Wadley", "Horseshoe Bend", "Peters Island \n (Horseshoe Bend)",  
col=c("red", "orange", "black", "grey"),  
lty=1, cex=0.8)
```

```
### figure 12
```

```
plot(num.gen.s, out.tal.reg$BUGSoutput$mean$gamma.site.yr[1,],  
family="serif", type="p", pch=16, ylim=c(0,0.5), ylab="Colonization probability",  
xlab="Number of annual generation events (standardized)", bty="n", cex=1 )  
for (i in 2:3){  
  points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,], type="p", pch=16)  
}  
for (i in 4:8){  
  points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,], type="p", pch=16,  
col="red")  
}  
for (i in 9:10){  
  points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,], type="p", pch=16,  
col="grey")  
}  
for (i in 11:15){
```

```

points(x=num.gen.s, y=out.tal.reg$BUGSoutput$mean$gamma.site.yr[i,],type="p", pch=16,
col="orange")
}
par(family = "serif")
legend(2.2, 0.35, title = "EXPLANATION",
      c("Malone", "Wadley", "Horseshoe Bend", "Peters Island \n (Horseshoe Bend)"),
      col=c("red", "orange", "black", "grey"),
      lty=1, cex=0.8)

```

### figure 8 - note that i put the values in from the model output

##### regression parameters

```

parnames<-c("reg.phi", "reg.gamma", "distance.phi", "distance.gamma", "gen.phi",
"gen.gamma", "temp.phi", "temp.gamma")

```

```

parmean<-c(-1.64, -1.05, 0.92, 0.06, -0.63, -0.29, -0.09, 1.04)

```

```

parlci<-c(-2.57, -1.85,0.40, -0.43, -1.31, -1.61, -0.69, 0.07) #90% credible intervals

```

```

paruci<-c(-0.73, -0.33, 1.54, 0.54, 0.05, 0.88, 0.51, 2.08)

```

```

par(mar=c(5, 4, 4, 2) + 0.1) # restore plot parameters to default

```

```

par(xpd = T, mar = par()$mar + c(0,0,0,9)) # set parameters to fit a legend..

```

```
plot(1:8, parmean, type="p", pch=c(16, 15), ylab="Model parameter estimates, logit
scale", bty="n",
     family="serif", ylim=c(-3,3), cex=1.5,
     xlab="", axes=FALSE)
```

```
axis(1, at = 1:8, labels=rep(NA, 11), tcl=-0.25)
```

```
axis(1, at=c(1.5, 3.5, 5.5, 7.5), labels=c("Flow regulation", "Distance below dam", "Generation
events", "Degree days"), tick = FALSE)
```

```
axis(2, at=seq(-3, 3, 0.5), labels=c("-3", "-2.5", "-2", "-1.5", "-1", "-0.5", "0", "0.5", "1",
                                     "1.5", "2", "2.5", "3"))
```

```
abline(0,0, xpd=FALSE)
```

```
segments((1:8), parlci, (1:8), paruci, col="gray")
```

```
# saved 950x550
```

```
legend(8.5, 0.75, title = "EXPLANATION",
```

```
      c("phi", "gamma"),
```

```
      pch = c(16, 15))
```

