

# Annotation Activity

*Salamander Life History: Grant M. Connette*

Life history determines salamander recovery rate from timber harvest in Southern Appalachian Forests

```
# Load Libraries #
library(unmarked)
library(AICcmodavg)

# Import Data
RNCOUNTS <- read.csv("Occupancy.csv", header = TRUE)

# Sort and make long form count data into single row per survey point (Plethodon shermani,
# Desmognathus wrighti, and Stream-Breeding Salamanders Separated)
RNCOUNTS <- RNCOUNTS[with(RNCOUNTS, order(Survey, Point)), ]
YStream <- matrix(RNCOUNTS$AllStream, nrow = 136, ncol = 5, byrow = FALSE)
YPlsh <- matrix(RNCOUNTS$AllPlsh, nrow = 136, ncol = 5, byrow = FALSE)
YDewr <- matrix(RNCOUNTS$AllDewr, nrow = 136, ncol = 5, byrow = FALSE)

#####
# Bundling/Formatting Data For Unmarked
#####

# List Possible Observation Covariates
ObsCovs <- list(Time = matrix(RNCOUNTS$Time1, 136, 5, byrow = FALSE),
                Time2 = matrix((RNCOUNTS$Time1^2), 136, 5, byrow = FALSE),
                Surftemp = matrix(RNCOUNTS$Surftemp, 136, 5, byrow = FALSE),
                Rain7 = matrix(RNCOUNTS$Rain7, 136, 5, byrow = FALSE),
                Rain3 = matrix(RNCOUNTS$Rain3, 136, 5, byrow = FALSE),
                Rain1 = matrix(RNCOUNTS$Rain1, 136, 5, byrow = FALSE),
                Date = matrix(RNCOUNTS$Date1, 136, 5, byrow = FALSE))

# Import Site Covariates
SiteCovs <- read.csv("OccuSiteCovs.csv", header = TRUE)
SiteCovs <- data.frame(Elev=SiteCovs[, "Elevation"],
                      Stand1=SiteCovs[, "StandAge"],
                      Stand2=(exp(SiteCovs[, "StandAge"])/100),
                      Stand3=log(SiteCovs[, "StandAge"]),
                      North=SiteCovs[, "Northness"],
                      East=SiteCovs[, "Eastness"],
                      TPI30=SiteCovs[, "TPI30"],
                      TPI100=SiteCovs[, "TPI100"],
                      Dist=SiteCovs[, "Dist"],
                      DistWt=SiteCovs[, "DistWt"],
                      TWI=SiteCovs[, "TWI"],
                      Solar=SiteCovs[, "Solar"],
                      Slope=SiteCovs[, "Slope"],
                      Point=SiteCovs[, "Point"],
                      Stream=SiteCovs[, "Stream_Dist"],
                      Litter=SiteCovs[, "Litter"])
```

```

# Create Unmarked Dataframe for each "species group"
Plsh.umf <- unmarkedFramePCount(y = YPlsh, siteCovs = SiteCovs, obsCovs = ObsCovs)
Stream.umf <- unmarkedFramePCount(y = YStream, siteCovs = SiteCovs, obsCovs = ObsCovs)
Dewr.umf <- unmarkedFramePCount(y = YDewr, siteCovs = SiteCovs, obsCovs = ObsCovs)

# Standardize Covariates
obsCovs(Plsh.umf) <- scale(obsCovs(Plsh.umf))
siteCovs(Plsh.umf) <- scale(siteCovs(Plsh.umf))
obsCovs(Stream.umf) <- scale(obsCovs(Stream.umf))
siteCovs(Stream.umf) <- scale(siteCovs(Stream.umf))
obsCovs(Dewr.umf) <- scale(obsCovs(Dewr.umf))
siteCovs(Dewr.umf) <- scale(siteCovs(Dewr.umf))

#####
## Fit Models - Plethodon shermani ##
#####

# Compare Models with complex stand age relationships
#####

fm13.PS <- pcount(~Rain7 + Time + I(Time^2) + Date + Surftemp + I(Surftemp^2) ~ North + East + Elev + S
fm14.PS <- pcount(~Rain7 + Time + I(Time^2) + Date + Surftemp + I(Surftemp^2) ~ North + East + Elev + S
fm15.PS <- pcount(~Rain7 + Time + I(Time^2) + Date + Surftemp + I(Surftemp^2) ~ North + East + Elev + S
fm16.PS <- pcount(~Rain7 + Time + I(Time^2) + Date + Surftemp + I(Surftemp^2) ~ North + East + Elev + S
fm17.PS <- pcount(~Rain7 + Time + I(Time^2) + Date + Surftemp + I(Surftemp^2) ~ North + East + Elev + S
fm18.PS <- pcount(~Rain7 + Time + I(Time^2) + Date + Surftemp + I(Surftemp^2) ~ North + East + Elev + S

# Note: We explored a range of Ks for integration (100-3000) in these NegBin models. AIC values increas
# but parameter estimates were not greatly affected, only the relative scales for estimated abundance a
# analysis was based on the default (max count + 100) which seemed reasonable given the typical densiti

# Generate AIC Table #
# Name Models
mspart3.PS <- fitList(
  "Linear" = fm13.PS,
  "Exponential" = fm14.PS,
  "Logarithmic" = fm15.PS,
  "Linear Distance" = fm16.PS,
  "Exponential Distance" = fm17.PS,
  "Logarithmic Distance" = fm18.PS)

(ms3.PS <- modSel(mspart3.PS))
print(coef(ms3.PS), digits=2)

# Generate AICc Table #
Cand.models <- list(fm13.PS, fm14.PS, fm15.PS, fm16.PS, fm17.PS, fm18.PS)
Modnames <- c("Linear", "Exponential", "Logarithmic", "Linear Distance",
  "Exponential Distance", "Logarithmic Distance")

print(aictab(cand.set = Cand.models, modnames = Modnames,
  second.ord = TRUE), digits = 4)

```

```

# Goodness-of-fit for Intercept-only and Top-ranked Models
#####

# Fit stats function
fitstats <- function(fm) {
  observed <- getY(fm@data)
  expected <- fitted(fm)
  resids <- residuals(fm)
  sse <- sum(resids^2, na.rm=TRUE)
  chisq <- sum((observed - expected)^2 / expected, na.rm=TRUE)
  freeTuke <- sum((sqrt(observed) - sqrt(expected))^2, na.rm=TRUE)
  out <- c(SSE=sse, Chisq=chisq, freemanTukey=freeTuke)
  return(out)
}

(fmPSNull.pb <- parboot(fm1.PS, fitstats, nsim=500, report=1))
(fmPSFull.pb <- parboot(fm14.PS, fitstats, nsim=500, report=1))

# Check residuals for spatial autocorrelation of points within stands
#####

Residuals<-t(residuals(fm18.AB))
Corr<-cor(Residuals,Residuals)
CorrMatrix<-(abs(Corr))
MeanCorr<-mean(CorrMatrix)
SDcorr<-sd(as.vector(CorrMatrix))
StandPairs<-matrix(1:136, nrow=2)

PairCorr<-seq(1:68)
for (i in 1:68){
  PairCorr[i]<-abs(Corr[((i*2)-1),(i*2)])
}

```