Supplementary Material S1. *OpenBUGS* software code used to fit an auto-logistic parameterization[1] of the dynamic (multi-season) occupancy model through Bayesian inference to study habitat and urbanization covariates of juvenile blue crab *Callinectes sapidus* occupancy dynamics in eight North Carolina (USA) tidal creeks. The full model (all covariates) is presented here. Effects not considered in the final model run are commented out with a single pound sign (#). Comments for the modeler are denoted by multiple pound signs.

```
model {

  psi ~    dunif(0,1)          ### initial occupancy probability

### Prior distributions for occupancy model auto-logistic parameters

      for(t in 1:(n.seasons-1)){
         a[t] ~ dnorm(0,0.01)
         b[t] ~ dnorm(0,0.01)
      }

### Prior distributions for ecological model parameters: effects on colonization

#     alpha1 ~ dnorm(0,0.01)    ### depth
        alpha2 ~ dnorm(0,0.01)    ### imperviousness
#     alpha3 ~ dnorm(0,0.01)    ### marsh percent
#     alpha4 ~ dnorm(0,0.01)    ### edge percent
        alpha5 ~ dnorm(0,0.01)    ### salinity

#     culud[1] <- 0        ### culvert location
#     culud[2] ~ dnorm(0,0.01)

### Prior distributions for ecological model parameters: partial effects on survival

#     alpha1S ~ dnorm(0,0.01)      ### depth
        alpha2S ~ dnorm(0,0.01)      ### imperviousness
#     alpha3S ~ dnorm(0,0.01)      ### marsh percent
#     alpha4S ~ dnorm(0,0.01)      ### edge percent
        alpha5S ~ dnorm(0,0.01)      ### salinity

### Prior distributions for observation model parameters

        beta0 ~ dnorm(0,0.01)       ### intercept
        beta1 ~ dnorm(0,0.01)       ### water temperature
#     beta2 ~ dnorm(0,0.01)       ### salinity
#     beta3 ~ dnorm(0,0.01)       ### dissolved oxygen

###### Ecological model fitted to standardized values for covariates

      for(i in 1:n.sites){
         z[i,1] ~ dbern(psi)
           for(t in 2:n.seasons){
```

```
                logit(muZ[i,t]) <-
            a[t-1] +    b[t-1]*z[i,t-1]
#         +    alpha1 * ((depth[i] - MeanDepth)/StDevDepth)
           +     alpha2 * ((imperv[i]    - MeanImperv)/StDevImperv)
#         +    alpha3 * ((marshpercent[i] - MeanMarshPercent)/StDevMarshPercent)
#         +    alpha4 * ((edgepercent[i] - MeanEdgePercent)/StDevEdgePercent)
           +       alpha5 *((OccSalt[i] - MeanOccSalt)/StDevOccSalt)
#         +      culud[culvertupdown[i]]

#         +    alpha1S * ((depth[i] - MeanDepth)/StDevDepth) * z[i,t-1]
           +     alpha2S * ((imperv[i]    - MeanImperv)/StDevImperv) * z[i,t-1]
#         +    alpha3S * ((marshpercent[i] - MeanMarshPercent)/StDevMarshPercent)      * z[i,t-1]
#         +    alpha4S * ((edgepercent[i] - MeanEdgePercent)/StDevEdgePercent)    * z[i,t-1]
           +     alpha5S * ((OccSalt[i] - MeanOccSalt)/StDevOccSalt) * z[i,t-1]

            Impute_Occ_Salt[i] ~ dunif(min_Occ_Salt, max_Occ_Salt) # impute missing values

      z[i,t] ~ dbern(muZ[i,t])      ### state model

       } #t
     } #i

### Calculated values for colonization and survival for covariates in final ecological model

     colImperv <- alpha2         ### colonization imperviousness
     colSalt <- alpha5         ### colonization salinity
     survImperv <- alpha2 + alpha2S        ### survival imperviousness
     survSalt <- alpha5 + alpha5S        ### survival salinity

###### Observation model fitted to standardized values for covariates

       for(i in 1:n.sites) {
                for(j in 1:n.reps) {
                        for(t in 1:n.seasons){
              Temp[i,j,t] ~ dunif(min_Temp, max_Temp)    ### impute missing current values
 #           Salt[i,j,t] ~ dunif(min_Salt, max_Salt)                  ### impute missing current values
 #           DO[i,j,t] ~ dunif(min_DO, max_DO)                    ### impute missing current values

              logit.p[i,j,t] <-    beta0
              + beta1 * ((Temp[i,j,t] - mean_Temp)/sd_Temp)
  #          + beta2 * ((Salt[i,j,t] - mean_Salt)/sd_Salt)
  #          + beta3 * ((DO[i,j,t]    - mean_DO)/sd_DO)

                 muY[i,j,t] <- z[i,t] * p[i,j,t]            ### observation model
                 y[i,j,t] ~ dbern(muY[i,j,t])
                 } #j
          } #i
     } #t

###### Calculated values

    for(t in 1:9){           ### each interval between seasons
```

```
    intcol[t] <-      a[t]                          ### rename colonization intercept by season
    intsurv[t] <- a[t] + b[t]           ### survival intercept by season
        }

###compute average intercepts for intervals between celestial seasons

intColSummerFall <- (intcol[1] + intcol[5] + intcol[9])/3                   ### colonization
intSurvSummerFall <-    (intsurv[1] + intsurv[5] + intsurv[9])/3    ###    survival

intColFallWinter <- (intcol[2] + intcol[6])/2                 ### colonization
intSurvFallWinter <-    (intsurv[2]+intsurv[6])/2         ###    survival

intColWinterSpring <- (intcol[3] + intcol[7])/2                ### colonization
intSurvWinterSpring <-    (intsurv[3]+intsurv[7])/2       ###    survival

intColSpringSummer <- (intcol[4] + intcol[8])/2            ### colonization
intSurvSpringSummer <-    (intsurv[4]+intsurv[8])/2       ###    survival

###### Calculations of colonization and survival by season
###### Calculate across range of values of each covariate retained in the ecological model
###### Transformed and back-calculated values

### imperviousness

for(i in 1:SampleSizePredictedImperv){
    for(t in 1:9){         ### each interval between seasons
        transColImp[i,t]          <-     intcol[t]          +     alpha2          *     ((PredictedImperv[i]     -
        MeanPredictedImperv)/StDevPredictedImperv)
      backtransColImp[i,t]     <- (exp(transColImp[i,t]))/(1+ exp(transColImp[i,t]))
      transSurvImp[i,t]          <-     intsurv[t]     +          survImp          *          ((PredictedImperv[i]     -
MeanPredictedImperv)/StDevPredictedImperv)
      backtransSurvImp[i,t]          <- (exp(transSurvImp[i,t]))/(1+ exp(transSurvImp[i,t]))
        }}

### salinity

for(i in 1:SampleSizePredictedSalt){
    for(t in 1:9){      ### each interval between seasons
            transColSalt[i,t]     <-     intcol[t]                    +     alpha5          *     ((PredictedSalt[i]     -
MeanPredictedSalt)/StDevPredictedSalt)
            backtransColSalt[i,t]          <- (exp(transColSalt[i,t]))/(1+ exp(transColSalt[i,t]))
            transSurvSalt[i,t]     <-     intsurv[t]          +     survSalt               *     ((PredictedSalt[i]     -
MeanPredictedSalt)/StDevPredictedSalt)
            backtransSurvSalt[i,t]    <- (exp(transSurvSalt[i,t]))/(1+ exp(transSurvSalt[i,t]))
          }}

###### Transformed and back-calculated values for detection probability
###### Calculate across the range of values of the covariate retained in the observation model

for(i in 1:SampleSizePredictedWaterTemp){
            transDetProb[i]    <- beta0    + beta1    * SampleStandardizedPredictedWaterTemp[i]
            backtransDetProb[i]      <- (exp(transDetProb[i]))/(1+ exp(transDetProb[i]))
```

```
    }
}    ### end model
```

[1] Royle, J.A.; Dorazio, R.M. *Hierarchical modeling and inference in ecology: the analysis of data from populations, metapopulations and communities.* Academic Press, Oxford. 2008.