Dear Jose,

the inits often need to be adjusted to your specific data set and model. In the SMR model particularly, you have to make sure that the inits of y correspond to your matrix of unidentified records.

Best

Rahel

|  |  |  |
| --- | --- | --- |
| /var/folders/xd/kf6m9lf540j9tvbwpj0wbb140000gn/T/com.microsoft.Word/Content.MSO/6D7DDB47.tmp | Rahel Sollmann | /var/folders/xd/kf6m9lf540j9tvbwpj0wbb140000gn/T/com.microsoft.Word/Content.MSO/DB84ABED.tmp6/28/14 |

Hi Jose,  
yeah, I know that problem. One thing that might improve the inits is to generate initial values for the activity centers outside of the inits function, and have the multinomial cell probabilities in the piece of code you quote above depend on those activity centers (assign unidentified records to the individual that lives closes to where the records were obtained). I have never tried if this helps, but is the only thing I can think of, off the top of my head.

Cheers

Rahel

hi Joe,

 This is a standard error which unfortunately does not have a standard solution.  It can arise due to bad starting values for the parameters but most often it arises because one of the latent variables is inconsistent with the observed data. For example: if you have a random starting value for s[i,] that is extremely far away from where an individual was captured or if you have , for the data augmentation variable, z[i] = 0 for a captured individuals (b/c it is known that z[i] = 1).

 So often you can fix this problem by thinking hard about the starting values.

 If you can't figure this out by tweeking starting values please send me your script exactly as it is generating the error and I'll work on it.

regards

andy

# Initial values for y

yin<-array(0, c(M,J,K))  
for(j in 1:J) {  
  for(k in 1:K) {  
    yin[1:M, j, k] <-rmultinom(1, n[j,k], rep(1/M, M))  
    }}

# Initial values for the s and sm

sm<-cbind(runif(m, xlims[1], xlims[2]), runif(m, ylims[1]))  
s<-cbind(runif(M, xlims[1], xlims[2]), runif(M, ylims[1], ylims[2]))    
inits<-function() {list(baseline.p=runif(1),alpha1=runif(1,0,1),  
  alpha2=rnorm(1), sm=sm, s=s, zm=rep(1,max), z=rep(1,M),yu=yin)}

Hi.

n<-datn-apply(datYknown, 2:3, sum)

n is a matrix created by difference: datn is a matrix [j,k] (traps,occasion) for all captures (marked+unmarked) and datYknown is a array for marked individuals

Regards