BayesGLM Summary

This document is meant to provide a summary of the modeling and code for analyzing FRAM trawl survey data that was developed over the summer of 2012 ("bayesGLM"). The objective of this project was to develop a flexible suite of tools that (1) calculates annual abundance indices for groundfish species from zero-inflated data (delta-GLM), (2) provides model selection tools to enable comparison of CPUE models of varying complexity, (3) provide diagnostic tools and plots for stock assessors developing trend indices.

The code provided here is platform-independent, and relies on a combination of R and JAGS (Just Another Gibbs Sampler, http://mcmc-jags.sourceforge.net/). It has been tested thoroughly, but comes with no warranty or promises of perfection.

Eric J. Ward & James T. Thorson
Northwest Fisheries Science Center
National Marine Fisheries Service
2725 Montlake Blvd., Seattle WA, 98112
email: eric.ward@noaa.gov, james.thorson@noaa.gov

Table of Contents

Data Requirements……3

Estimation and Model Overview……5

Plots and Model Diagnostics……10

Data diagnostics……10

Model diagnostics……11

Model selection……11

Literature Cited……13

Example Code……14

Frequently Asked Questions……15

Data Requirements

Several files are necessary to run the delta-GLMM code, and all files need to be in .csv format. The files are:

defaultLimits.csv
Data.csv
SA3.csv

1. defaultLimits.csv : a file containing the default strata limits. These are read into an R object named **`strata.limits`**. For testing, we used the following values

```
> strata.limits
   STRATA NLat SLat MinDepth MaxDepth
1       A 50.0 47.5   54.864   182.88
2       B 50.0 47.5  182.880   548.64
3       C 50.0 47.5  548.640  1280.16
4       D 47.5 43.0   54.864   182.88
5       E 47.5 43.0  182.880   548.64
6       F 47.5 43.0  548.640  1280.16
7       G 43.0 40.5   54.864   182.88
8       H 43.0 40.5  182.880   548.64
9       I 43.0 40.5  548.640  1280.16
10      J 40.5 34.5   54.864   182.88
11      K 40.5 34.5  182.880   548.64
12      L 40.5 34.5  548.640  1280.16
13      M 34.5 32.0   54.864   182.88
14      N 34.5 32.0  182.880   548.64
15      O 34.5 32.0  548.640  1280.16
```

2. Data.csv : a file containing data for at least 1 species to be run, but this can be the entire trawl database, with species across columns. For consistency of naming variables, the contents of Data.csv (stored in **`masterDat`**) are as follows:

```
> head(masterDat[,1:9])
  X     OP_CODE YEAR VESSEL SURVEY BEST_LAT_DD BEST_DEPTH_M AREA_SWEPT_MSQ arrowtooth
1 1 1.99801e+11 1998      1      1    48.14417     268.9472       16574.28     73.514
2 2 1.99801e+11 1998      1      1    48.14858     444.2620       13820.47     30.516
3 3 1.99801e+11 1998      1      1    47.84708     193.4006       14903.20     14.521
4 4 1.99801e+11 1998      1      1    47.82417     306.7198       22079.41     26.306
5 5 1.99801e+11 1998      1      1    47.82300     914.4000             NA      0.000
6 6 1.99801e+11 1998      1      1    47.79692    1044.7109       24818.36      0.000
```

3. SA3.csv : a file containing the area of each spatial / depth stratum. The areas used in our testing were in hectares, with depths in meters. The header of the file is:

```
> head(SA3)
  SUBAREA_ID MIN_DEPTH_M MAX_DEPTH_M AREA_HECTARES SUBAREA_SET_ID MIN_LAT_DD MAX_LAT_DD
1      80847        1200        1280      2625.175              3       48.0       48.5
2      80848        1200        1280     16735.788              3       47.5       48.0
3      80849        1200        1280     13081.261              3       47.0       47.5
4      80850        1200        1280     11086.655              3       46.5       47.0
5      80851        1200        1280     11102.496              3       46.0       46.5
6      80852        1200        1280      4314.302              3       45.5       46.0
```

For consistency, stock assessors should use the same column names when possible. If other units or headings are used, be aware that some functions for processing output and calculating standardized indices will not function properly. Some of these functions include `MapData()`, called by `doMCMCDiags()`, and the stratum-area calculations done by `doMCMCDiags()`.

Estimation and Model Overview

The core function of the delta-GLMM estimation is named `fitCPUEModel.` This function is sourced automatically via the file "fitCPUEModel vX.X". Arguments for the function, and their default options are as follows

`modelStructure :` this is a list specifying how optional random effects are to be treated. The elements are:

```
StrataYear.positiveTows (default = "random")
VesselYear.positiveTows (default = "random")
Vessel.positiveTows (default = "random")
StrataYear.zeroTows (default = "random")
VesselYear.zeroTows (default = "random")
Vessel.zeroTows (default = "random")
```

By default, each deviation is treated as independent, iid normally distributed random effects. Other options are treat these deviations as fixed effects (`"fixed"`) or omit them entirely (`"zero"`). For cases when random effect variances are close to 0 or deviations are clustered at the posterior mode, we've implemented Gelman's variance expansion model as (`"randomExpansion"`; Gelman 2006, Gelman et al. 2007). Finally, deviations in the positive and binomial models for strata-year or vessel-year interactions may be treated as correlated multivariate normal random effects, but only if both the positive and binomial models are specified as such. For example, to estimate correlated strata-year effects, both `StrataYear.positiveTows` and `StrataYear.zeroTows` are set to `"correlated"`.

```
year.deviations (default = "fixed")
strata.deviations (default = "fixed")
```

Both year and strata deviations are always estimated as fixed effects, and cannot be removed from the model (without directly editing JAGS/BUGS code). For the purposes of stock assessments, the `year.deviations` and `strata.deviations` elements should be left at the defaults. For other applications, users may

wish to estimate these deviations as correlated random effects, and each argument may be specified as `"correlated"`.

`Catchability.positiveTows (default = "one")`, specifies how to treat the offset for effort in the positive model. When set to 1, the offset is just ln(effort), in link space (log link). Other arguments can be `"linear"`, corresponding to an offset estimated as = B_1* ln(effort) or `"quadratic"`, corresponding to = B_1* ln(effort) + B_2* [ln(effort)]^2.

`Catchability.zeroTows (default = "zero")`, specifies how to treat the offset for effort in the binomial model. When set to 0, the effort is not included. Other arguments can be `"one"`, corresponding to an offset of effort in link space (logit) with no additional parameters estimated (= 1*effort), `"linear"`, corresponding to an offset estimated as = B_1*effort or `"quadratic"`, corresponding to = B_1*effort + B_2*(effort^2).

`covariates` – this is a list of 2 elements,

`binomial : boolean (default = "F"),` are covariates being included in the binomial model? If `"T"`, they must be specified in a matrix named `X.bin` in the R workspace, with observations on the rows and variables on the columns.

`positive : boolean (default = "F"),` are covariates being included in the positive model? If `"T"`, they must be specified in a matrix named `X.pos` in the R workspace, with observations on the rows and variables on the columns.

`likelihood` : character string (default = `"gamma"`) specifying the likelihood of the positive model. Other likelihoods available are `"invGaussian"` or `"lognormal"` – in each case, the coefficient of variation (CV) is modeled as a constant. For species with extreme catch events (ECEs), the positive model may be specified as a 2-part mixture, and there are two variants of each of these: either `"lognormalECE"` or `"lognormalECE2"` and either `"gammaECE"` or `"gammaECE2"`. Finally, we have implemented two discrete distributions: the Poisson = `"poisson"`, and Negative Binomial = `"negbin"`.

Gamma likelihood

$\frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x)$, with this distribution $E[x] = \alpha/\beta$, and $CV = 1/\sqrt{\alpha}$. We assume that the CV rather than the variance is constant across tows, and to maintain consistency across likelihoods, we assign a weakly informative gamma prior to $1/CV^2$ (for the gamma, this corresponds to α). For probability models, a log-link function is implemented, so that *log(E[x]) = f(covariates, strata effects, year effects, etc)*.

Lognormal likelihood

$\frac{1}{x\sigma\sqrt{2\pi}} \exp\left(\frac{(\ln(x)-\mu)^2}{2\sigma^2}\right)$, with this distribution $E[x] = \mu + \sigma^2/2$, and $CV = \sqrt{\exp(\sigma^2) - 1}$; for small values of σ, this results in CV $\sim \sigma$, but this approximation does not hold for the majority of the species included in the trawl survey data. We assume that the CV rather than the variance is constant across tows, and to maintain consistency across likelihoods, we assign a weakly informative gamma prior to $1/CV^2$. For probability models, a log-link function is implemented, so that *log(E[x]) = f(covariates, strata effects, year effects, etc)*.

Inverse Gaussian likelihood

$\left(\frac{\lambda}{2\pi x^3}\right)^{1/2} \exp\left(\frac{-\lambda(x-\mu)^2}{2\mu^2 x}\right)$, with this distribution $E[x] = \mu$, and $CV = \sqrt{\mu}/\lambda$. We assume that the CV rather than the variance is constant across tows (so λ is held constant), and to maintain consistency across likelihoods, we assign a weakly informative gamma prior to $1/CV^2$. For probability models, a log-link function is implemented, so that *log(E[x]) = f(covariates, strata effects, year effects, etc)*.

Gamma ECE and Lognormal ECE likelihoods

Using these likelihoods, each tow is probabilistically assigned to be a member of the regular or extreme catch event (this parameterization is less compatible with DIC, however, and a second parameterization is below). Following Thorson et al. (2011), and the equations for the gamma and lognormal models described above, we assume that the mean catch size of ECE tows is equal to the non-ECE mean times a multiplication factor,

$u_{ECE} = \lambda * u_{non-ECE}$. We assign a uniform (0, 5) prior to ln(λ), constraining ECE tows to be larger than non-ECE tows. The only other parameter estimated in the ECE model is the mixture probability of an ECE tow occurring, p. We assign an uninformative Dirichlet (1,1) prior to p and $(1 - p)$, and we the membership of each tow (being an ECE or not) is treated as a latent categorical state.

Gamma ECE2 and Lognormal ECE2 likelihoods

Unlike the above, we implement the parameterization based on the marginalized values. Each tow is not assigned to a type of tow, but modeled as a mixture of two distributions. For identifiability, we again constrain the location parameter of the lognormal (log of the median) for extreme catch events to be greater than the location parameter of regular catch events. For the Lognormal ECE2 distribution, this is implemented by assigning a uniform (0, 5) prior to ln(λ), and modeling the location of ECE catches as $u_{ECE} = u_{reg} + \lambda$. For the Gamma ECE2 distribution, λ is used to shift the mean, $u_{ECE} = u_{reg} \cdot \exp(\lambda)$. We don't place a similar constraint on the variance, because it's possible that only several tows affect the ECE distribution. Given these two likelihood components, we assign an uninformative Dirichlet (1,1) prior to p and $(1 - p)$, and calculate the total likelihood as $p \cdot L_1 + (1 - p) \cdot L_2$.

Poisson likelihood

We implement the standard log-link function, relating covariates to expected values, where *log(E[x]) = f(covariates, strata effects, year effects, etc)*

Negative binomial likelihood

We implement the standard log-link function, relating covariates to expected values, where *log(E[x]) = f(covariates, strata effects, year effects, etc).* Following the lognormal and gamma parameterizations above that assigned priors to the CV, we assign a weakly informative gamma prior to $1/CV^2$.

`model.name :` character string (`default = "deltaGLM.txt"`). To preserve the model file for each run, this should be set to a unique string (otherwise the model files will be written over in the working directory).

`fit.model :` boolean (`default = "T"`). If `"F"`, the model file is written, but MCMC estimation is not done. One advantage in not fitting a model would be to manually edit the priors (for instance, they could be made more informative).

`mcmc.control` – this is a list of 2 elements,

`chains : numeric (default = 5),` number of MCMC chains. When run in parallel, each chain is assigned to a unique processor (when available).

`thin : numeric (default = 1),` MCMC thinning rate. When species occur with very low or very high frequency (probabilities near 0 or 1), the autocorrelation of the parameters may be high (because in logit space, the difference between -10 and -20 is much greater than in normal space). In these situations, it's more important to look at the acf of the yearly density estimates.

`burn : numeric (default = 5000),` length of MCMC burn-in phase. This should be at least 5000.

`iterToSave : numeric (default = 2000),` number of iterations (after burn-in) to be saved and returned to R. The total number of posterior samples returned to R can be calculated as (# MCMC chains * iterToSave) / thinning rate

`Parallel :` boolean (default = `"T"`). This is currently only an option for PCs (if estimation is done on a Mac, this must be = `"F"`).

`Species :` character string (default = `"NULL"`). This is required, but may be anything specified by the user ("darkblotched", "Dark blotched", "Db", etc)

`logBounds :` Numeric 2-element vector that describes bounds on the parameter space in link (log) space (default = `-20, 20`).

`logitBounds :` Numeric 2-element vector that describes bounds on the parameter space in link (log) space (default = `-20, 20`).

`prior.scale :` Numeric 4-element vector corresponding to the prior precision (default = c(`25,25,25,25,25,25`)), where the elements correspond to: (1) strata-year positive model, (2) strata-year binomial model, (3) vessel-year positive model, (4) vessel-year binomial model, (5) vessel positive model, (6) vessel binomial model.

This approach is different from usual treatment of random effects. For example, if a linear model contains random deviations $\boldsymbol{B_j} \sim \boldsymbol{Normal}(\mathbf{0}, \boldsymbol{\sigma_B})$, the variance expansion model treats them as $\boldsymbol{a} \cdot \boldsymbol{\eta_j} \sim \boldsymbol{Normal}(\mathbf{0}, \boldsymbol{\sigma_\eta})$, where $\boldsymbol{a}$ is a multiplicative nuisance parameter, and $\boldsymbol{a} \cdot \boldsymbol{\eta_j} = \boldsymbol{B_j}$. We assign $\mathbf{1}/(\boldsymbol{\sigma_\eta})^{\mathbf{2}}$ a ~ Chi-squared(1) prior distribution, and the prior.scale[i] argument determines the variance for $\boldsymbol{a}$, which is assigned a normally distributed prior centered on 0; the default prior is $\boldsymbol{a} \sim \boldsymbol{Normal}(\mathbf{0}, \boldsymbol{\sigma^2} = \mathbf{25})$. The standard deviation or variance of the random effects is then treated as a derived parameter, $\boldsymbol{\sigma_B} = |\boldsymbol{\alpha}|\boldsymbol{\sigma_\eta}$. For more details, see Gelman (2006) or Gelman et al. (2007).

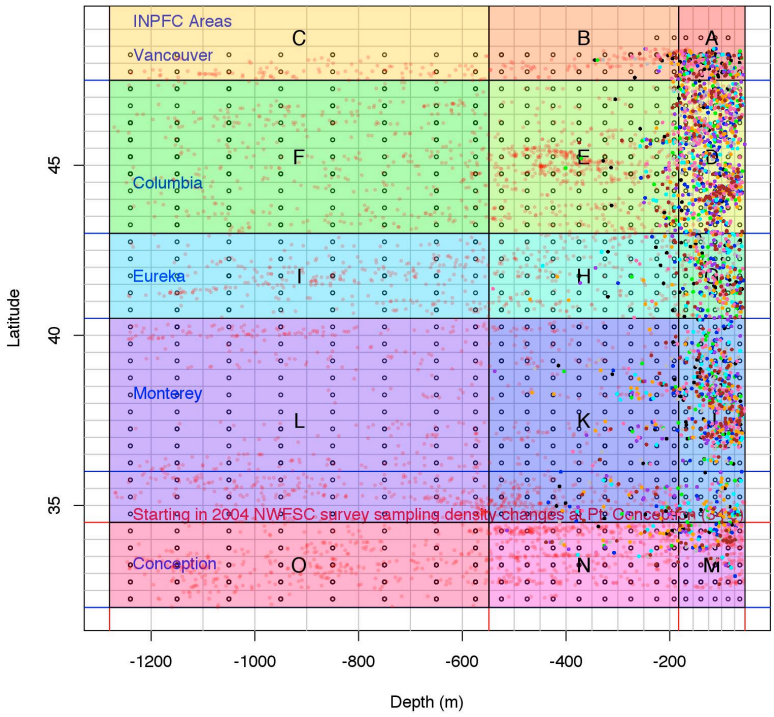
Plots and Model Diagnostics

Model output and processing is meant to be done after several models have been run, and stored in a list (see example below). We provide 2 types of diagnostics: (1) data diagnostics, common to all models for a given species, and (2) model diagnostics, specific to a particular parameterization.

1. Data diagnostics

Diagnostic figures summarizing raw data include:

- Positive catch rates by year
- Positive catch rates as a function of depth
- Positive catch rates as a function of depth (stratified by year)
- Positive catch rates as a function of latitude
- Positive catch rates as a function of latitude (stratified by year)
- Species' presence (% positive) by year
- Species' presence (% positive) by depth and year
- Species' presence (% positive) by latitude and year
- Tow map, by strata (example for petrale sole below)



2. Model diagnostics

If several models are stored in a list object (example below), the output can be processed simultaneously, with plots for each model being written to separate folders.

For diagnostic plots of MCMC convergence, we include the following:
- trace plots of all estimated parameters, for all MCMC chains
- autocorrelation plots and estimates for all MCMC chains
- density plots of variance parameters
- density plots of correlation parameters (if estimated)

For diagnostic plots of model fit, we include the following:
- realized offset
- posterior predictive plots for the positive model
- comparison of maximum likelihood and Bayesian indices of abundance (using both area-weighted and –unweighted) estimates by strata
- biomass and CV estimates, by year and strata

3. Model Selection

After simulation testing the results from multiple models across a variety of model selection criterion, we found that both popular calculations of the Deviance Information Criterion (DIC) to be unreliable, and not in agreement with one another (*Thorson & Ward, in review*). As an alternative, we have provided the log density, which can be calculated separately for each model. For a given model object, `mods[[1]]`, the log density can be calculated by calling the function `logDensity()`, called as:

```
> logDensity(mods[[1]])
```

After partitioning the total data set into 2 components (0s and 1s for presence/absence in the binomial model, and only values greater than 0 for the positive model), we separately calculate the mean density for each data point j as

$$E\left[L\left(y_j\middle|\theta\right)\right] = \sum_{i=1}^{N_{MCMC}} \frac{L(y_j|\theta_i)}{N_{MCMC}}$$

where θ_i represents a vector of model estimated parameters corresponding to the i[th] MCMC sample, and N_{MCMC} represents the total number of samples. The log-density of all data points is then calculated as log $(E[L(\boldsymbol{y}|\theta)])$, integrating over all MCMC parameters. The total quantity returned by the `logDensity()` function is the sum of the 2 components,

$$\log(E[L(\boldsymbol{y_{Total}}|\theta)]) = \log(E[L(\boldsymbol{y_{binomial}}|\theta)]) + \log\left(E\left[L\left(\boldsymbol{y_{positive}}\middle|\theta\right)\right]\right)$$

Literature Cited

Gelman, A. 2006. Prior distributions for variance parameters in hierarchical models. Bayesian analysis, 1(3):515-533.

Gelman, A., D.A. Van Dyk, Z. Huang, and W.J. Boscardin. 2007. Using redundant parameterizations to fit hierarchical models. Journal of Computational and Graphical Statistics, 17(1), 95–122.

Thorson, J.T., I. Stewart, and A.E. Punt. 2011. Accounting for fish shoals in single- and multi-species survey data using mixture distribution models. *Can. J. Fish. Aquat. Sci.*, 68(9): 1681-1693.

Thorson, J.T. and E.J. Ward. 2013. Accounting for space–time interactions in index standardization models. *Fisheries Research*, 147: 426-433.

Example Code

```
rm(list=ls())
library(stats)
library(runjags)
library(R2jags)
library(coda)
library(superdiag)
library(R2jags)
library(pscl)
load.module("glm") # this is loading a specific library in JAGS / BUGS that
 implements a conditional sampler
runif(1) # needed for PCs

setwd("/Users/xwarder/Dropbox/delta GLMM project")
my.wd = "bayesGLM_final/" # project wd
source(paste(my.wd,"bayesGLM v2.3.r",sep="")) # source the scripts
Letters = apply(MARGIN=1,FUN=paste,collapse="",expand.grid(letters,letters))

# read in the master data file
masterDat = read.csv(paste(my.wd,"Data.csv",sep=""))
masterDat = masterDat[which(masterDat$SURVEY==3),]
strata.limits = read.csv(paste(my.wd,"defaultLimits.csv",sep=""))

names(masterDat) # check species available to be run
species = "petrale"

# call the function to process the data frame:
processData()
# Set MCMC parameters
# Note: the total iterations saved will be chains*iterToSave/thin
mcmc.control = list(chains = 5, thin = 1, burn = 200, iterToSave = 500)
# Set Parallel argument - for PCs only
Parallel = TRUE

# Model 1 contains correlated positive components for strata-year and vessel-
 year interactions,
mods = list()
mods[[1]] = fitCPUEModel(modelStructure=list("StrataYear.positiveTows" =
 "correlated","VesselYear.positiveTows" = "correlated","StrataYear.zeroTows" =
 "correlated","VesselYear.zeroTows" = "correlated", "Catchability.positiveTows"
 = "linear", "Catchability.zeroTows" = "linear", "year.deviations" = "
 fixed","strata.deviations" = "fixed"),
mcmc.control=mcmc.control, Parallel=Parallel, Species=species)
# Model 2 is a simple model, with only strata and year effects estimated
mods[[2]] = fitCPUEModel(modelStructure=list("StrataYear.positiveTows" =
 "zero","VesselYear.positiveTows" = "zero","StrataYear.zeroTows" =
 "zero","VesselYear.zeroTows" = "zero", "Catchability.positiveTows" = "one",
 "Catchability.zeroTows" = "zero", "year.deviations" =
 fixed","strata.deviations" = "fixed"),
mcmc.control=mcmc.control, Parallel=Parallel, Species=species)

# process MCMC output
doMCMCDiags(my.wd, mods)
```

Frequently asked questions

1. **Q: Is it possible to fit a model with just a binomial part, or just a positive part?**

A: yes, but this requires a couple tricks in the code, where any 'data' assigned NA will not contribute to the likelihood. For example, to fit a positive model only, all of the binomial data needs to be set to 0,

```
isNonZeroTrawl[1:length(isNonZeroTrawl)]=NA
```

and then estimation can be done. Alternatively, if only a binomial GLMM is to be run, then the data associated with just the positive data can be set to NA,

```
y[1:length(y)]=NA
```

To recover either data objects for the full model, the processData() function needs to be run again.