**DDwSE3\_jags**

**Purpose**  This is the model for both Brown Bank North and Georges Bank “a”. This isn’t an R script but is a model written in BUGS/JAGS language

**Version Control**  This builds on DDwSE3 which is the winBUGS version of this JAGS code. This basically moved from winBUGS into JAGS, to do this only very minor changes to code were necessary, mostly around using truncated priors. I have added some output and commenting to the function…

**Required packages** R2jags (the locally derived functions have their own package needs as well)

**Preamble**

This help file has two function; a: it explains how the model is conceptually put together, this is done in an effort to help walk a new user through the model is as non-technical/jargony fashion as possible, it’s a reasonably complex model so it’s not light lifting no matter what, and b: this breaks down the BUGS/JAGS code used for the offshore model and walks through the script as best as possible relating it all back to the theory and trying to link everything together. One deviation that was undertaken was to walk through the model itself before discussing the priors (in the script the priors come first then the model), simply because if you don’t understand the model you really won’t understand the priors. I don’t’ get into any deep discussion of Bayesian modelling, you’ll need to look elsewhere to understand that.

1. **Theory**
   1. **Delay-difference model (aka the process model)**

A slight aside to start; the model doesn’t run on data from a calendar year, instead it is based on what we call a “survey year”. A survey year runs from the end of the survey to the following years survey. In a typical year the survey happens in August on GBa and May on BBn. This means that a “survey year” on GBa is from September to August, and on BBn it is usually from June-May. This causes complications since the TAC for the fishery is set based on a calendar year thus when you run this model you need to be wary that you have this complication sorted out in your head. The help scripts for the associated functions should help you here and everything is set up in these scripts to account for this quirk.

The heart of this model is a Delay-difference model (see Deriso 1980, Schnute 1985,1987, Fournier and Doonan 1987, or Hiborn and Walters 1992 for more details). The underlying model accounts for natural mortality, physical growth of the scallops over the course of a year, the catch of the scallops by the fishery, and the addition to the population from smaller scallops not yet “incorporated” into the fishery (recruit scallops). So this all results in the frightening looking equation…

(1)

While it looks scary it is nothing but a simple balance equation of removals and additions to the population, let’s define some terms

* Biomass (usually in tonnes or kT), this is the biomass at the time of the

survey. This is for fully recruited scallop.

* Natural mortality of fully recruited scallop. This is considered the

mortality for the current survey year (*t*), it is applied to the Biomass remaining after accounting for catch in the current “survey year” (*t*) and growth of that biomass over the course of the year.

* Annual growth rate of scallop found in survey from previous year (*t-1*)

The model enables growth to slow as average weight of the population increases (large individuals grow more slowly). Growth model details are outlined in detail below.

* Biomass (usually in tonnes or kT), estimate from previous survey (*t-1*)
* Fisheries catch (usually in tonnes or kT) during the “survey year”.
* Natural mortality of recruit scallops. This is considered the

mortality for the current survey year (*t*), it is applied to the Biomass of recruits from last “survey year” (*t*) and growth of that biomass over the course of the year. Note that catch for the recruits is assumed to be 0.

* Annual growth rate of recruit scallop found in survey from previous year

(*t-1*). Growth model details are outlined in detail below.

* Recruit Biomass (usually in tonnes or kT), this is the biomass at the time

of the previous survey (*t-1*)

There are some general concerns with delay difference models (see Hilborn and Walters 1992), but often these are related to a lack of quality information about the recruitment component of the model. Given our excellent recruitment data, which is extremely rare in almost any fishery, many of these issues will not be as problematic for scallop as for a typical fishery (for example we don’t need a stock recruit relationship).

It should be noted that at this point we haven’t discussed the influence of either observation error or process error, this summary so far provides a “deterministic” world view of our model, as we will see in the process model section we need to incorporate both forms of uncertainty into our model to start to understand the uncertainty in the results that our model provides.

* + 1. *Removals*

In the model there are two avenues that result in a decline in the number of scallop. The easiest to deal with is the catch (, which is simply the catch in tonnes from the end of the previous survey (*t-1*) to the end of the current survey (*t*). For GBa this is typically removals from September – August, for BBn this is typically removals from June-May.

The second types of removals is through natural mortality (fully recruited and a recruit each get their own natural mortality term). The natural mortality is calculated differently for both fully-recruit and recruit scallops. There are several pieces of information that the mortality terms are based on, the survey biomass index (), the CPUE index ( though this is down-weighted and I believe should be removed from the model in the next framework), and a fully recruited clapper index (). These 3 factors combine to give the mortality for fully recruited scallop. For the recruits the survey biomass index ) and recruit clapper index () are the primary drivers of natural mortality, but of course it is a big complex model so the fully recruited terms will interact with the recruit mortality estimates among many other parameters.

* + 1. *Growth*

Growth in terms of biomass also happens via two avenues. I’ll start with the second and it leads more naturally to the first. We assume that all scallops in the recruit size class last year (*t-1*) grow and become part of the fully recruited size class this year. The recruit part of the equation is everything after the addition sign in Equation (1). Of course last year’s recruits will grow over the course of a year (and some die, see natural mortality above in Removals sub-section) and the growth term () is where we account for this growth in size of the average scallop. The same idea holds for the fully recruited scallop, we take last year’s scallops and assume they grow at some average rate ().

These growth models are all calculated before running the model itself and are simply multiplicative terms used in the model, they are considered “known” parameters and any uncertainty in the growth terms are not included in the model itself. Explaining the theory of the growth terms is, therefore a slight aside so you don’t really need to read the below section to understand how the model works.

***Growth Model***

As mentioned the growth model doesn’t come directly into our model, but I think it will be helpful to discuss here. The growth model essentially is used to predict average meat size next year and divide it by the current meat size.and it contains two streams of information. The first is the growth of scallop shell over the course of a year, this is found from fitting ageing data to a von Bertalanffy growth model. The second component uses condition (mass of a 100 mm scallop) in conjunction with the shell height information to get both current average weight of a scallop meat and the projected average weight of a scallop meat next year

Von Bertalanffy model

The first piece of the growth model puzzle is the von Bertalanffy model. We fit a von Bertalanffy model to various sources of age-height data that have been cobbled together for the offshore . The von Bertalanffy model is:

where:

: The length at age *t*.

*t*: Scallop age (in years).

*K*: The growth rate. This is the first model parameter is estimated from our data.

: The length at which growth stops (obviously this assumes growth stops at some length/ag)e. This is the second model parameter that is estimated from our data.

Using the parameters from this model we can predict how much we expect a shell of a certain height to grow over the course of 1 year. In the model we take the average shell height () of the scallops in the current year and estimate the expected size of these scallops in the following year.

This is done for both the recruit and fully recruited scallops separately. This step is fairly straightforward and this is really all our ageing program is doing at the moment; getting us predictions of average shell height next year given the average shell height on the bank this year.

***Growth Model (cont)***

Growth

I won’t get into our condition model here (see help files and comments embedded in shwt.lme.r and condfac.r for details of the condition factor calculations). Using the average condition factor () for the bank in the current year and the average size of shell () in the current year we can get an estimate of the average meat weight of a scallop in the current year ().

From the von Bertalanffy model we have the predicted shell heights for next year, and for all but the current year we know what the condition factor was the following year so we can get the predicted meat weight next year as

Note that to get the projected meat weight in the final year we simply use the current years condition as the condition next year.

Finally the ratio of the predicted to current meat weight gets us an estimate of growth of the scallop this year (in terms of meat weight)

This same procedure is used to estimate the growth of the recruit meat weight over the course of a year.

* 1. **Observation Models**

Within this model there are several observation models. In state space lingo these observation models link the observed states (i.e. clapper numbers, survey biomass indices, and CPUE in this model) to the unobserved (hidden) state(s) (in our case the hidden states include fully recruited biomass, recruit biomass, and the natural mortalities (). Observation models can simply be thought of as models that inform the process model by having measured their “response” variable; these observation models get linked to the process model via an observation equation which includes the observation as the response and the hidden state as model parameter. The response variable from the observation model does not directly enter into the process model. The ability of a measure to inform the overall model can depend on many things, including a: on the uncertainty associated with our measure, b: the ability of the observed data to inform the process model, c: the relationship between the different observation models, d: surely many other things I’m not thinking about.

1. *Natural mortality model (aka Clapper popcorn model)*

This model largely builds on the Dickie (1955) and Merrill and Posgay (1964) models which found (among many other things) that clappers once formed will last less than one year (somewhere between 50 and 231 days). Their solution to estimate natural mortality was a steady-state solution.

where *m* is instantaneous rate of natural mortality (), *Z* is the number of clappers, *L* is number of live individuals (Note this isn’t biomass it is abundance), and *S* is the time it takes for a clapper shell to separate (proportion of a year, 1 would mean it takes 1 year to separate the equation would simplify to meaning clappers were a direct estimate of mortality). Given the research above it appears clappers tend separate in less than one year, so when we survey annually the clappers we see are some subset of the clappers that occur during the year. While more complex than if we could assume S = 1, we can at least think of S as a proportionality constant and don’t have to worry about accounting for clappers from previous years. This makes our calculations somewhat easier when we move from this steady-state solution to something more dynamic. The previous equation is actually the steady-state solution to this equation (I’m guessing if we sat down we could come up with other formulations that would give us this same result, but this is a reasonable model IMHO).

If we assume steady state thenand don’t vary over the course of a single year they can be considered constants in this equation, if you integrate it you’ll see you end up with the Dickie equation . Now one thing that messed with me for a bit, within this model time is multi-scaled, I think of *δ* is the time within a year (intra-annual) and is really looking at rates within a year, whereas *t* is interested in annual change (inter-annual). For example the term is the number of clappers in year t, whereas the terms inside the integral are saying these vary over the course of a year. So our integration is attempting to look at changes over the course of a single year. This is nice as we can assume is a constant value for any one year and we can remove this from the integration, resulting in.

Now what to do with, this is the number of live indivduals during the year, we can assume this changes linearly over the course of the year. If we do this we can simply interpolate the number alive between last estimate and the current estimate at time *δ* (remember *δ* is on a within year scale, so *δ* is between 0 and 1, *δ* = 0 would represent 1 year ago, and *δ* = 1 would represent the most recent year). Doing so results in a simple weighted average/interpolation formula

Remember that the only reason we can do this within a year is that empirical evidence suggests *S < 1* (i.e. all clappers we see formed within the current year. Now we can pop this back into the main integral

Integration of this (good step by step walk through of this in the Appendix of Res Doc (2002/018 - Smith and Lundy 2002) with some re-arrangement to solve for results in (there are a couple of different ways this equation gets written, this is a nice way to see how the weighted average/interpolation across the separation time (S) remains in the model results).

(2)

Note that all of this ignores very real issued with clapper separation due to gear interactions and likely age effects in separation time, something that could perhaps be address by a closer inspection of our survey results from seedboxes.

1. *Biomass models*
   * + 1. *Survey Biomass Index*

The fully-recruited and recruit biomasses obtained from the survey are the critical observations used to get the model estimated biomasses (for fully recruited and recruit scallop) in the main process equation. The survey results provide an index of biomass but when coupled with information about some agent of mortality (generally fisheries removals is used though I wonder how different our estimate would look like if we just used a clapper index, I think that would be sufficient to get a biomass estimate) an actual biomass for an area can be estimated. For this model the survey biomass index is assumed to be proportional to the “true” biomass. For the fully recruited biomass this is (note that see below for explanation of and):

While for recruits it is (note that):

* + - 1. *CPUE Index*

In addition the CPUE from the fishery is also used to estimate the fully recruited biomass. It too is assumed to be proportional to the “true” fully recruited biomass:

Notice our proportionality constant in these equations is for the survey data, and for the CPUE data. These numbers can be very different given that the units for are entirely different than for . To balance the survey biomass equations must be a unitless number since is already in the same units as and as. For example

So hopefully you can see that to balance the “units” of this equation must be a unitless number, this makes its interpretation as a catchability coefficient fairly straightforward (i.e. it’s a simple proportion).

For the CPUE the catchability coefficient is a rather different beast as the units for while the units for

:

Rearranging the equation on the right yields the units of to be…

Which clearly isn’t a coefficient that we can easily compare with. This does suggest that it would make sense to convert from kg to tonnes though I that would make a factor of 1000 smaller (and you’ll see it’s already small!). Perhaps for modeling purposes when dealing with CPUE changing the unit effort to be something much larger would make sense (set CPUE to be catch in tonnes if sweeping for an “area” in terms of km-days (e.g. CPUE in would be 24 times larger than the CPUE), we’d be looking at an “area” swept that was 1 day long and 1 km wide rather than an “area” swept that was 1 hour long and 1 meter wide. Hope that all makes sense!

1. **Dealing with uncertainty**

Before discussing uncertainty in the models there is one very annoying thing that we need to clarify with WinBUGS, that is that the “variance” position for normal and log-normal distributions are actually inverse variances and are known as precisions. This can get confusing, especially where we start talking about the variance parameters themselves. In this section we talk in terms of variances in the traditional statsy way, but where necessary I may diverge into a side-bar to attempt to clarify what is going on (especially weird to wrap ones head around in the Gamma-inverse Gamma discussion later on).

So far we haven’t discussed how we incorporation variability/uncertainty/error into our model. The model parameters all have some level of “variability” associated with them (except for our growth model). All the non-growth terms in the model are modeled as random variables with distributions specified in the model. You’ll notice the specified error distributions for the parameters used directly in the process and observation models are all log-normal, this is a direct result of the processes being assumed to be multiplicative on the original scale of the data. This results in an additive error structure when the process equations are log-transformed and the error is assumed to be normally distributed in “log space”.

There are two types of distributions that are specified in a Bayesian model, the first I’ll mention specify the uncertainty associated with our process and observation models themselves. The error terms within the multiplicative models are given a mean of 1 (i.e. that is mean 0 in log-space where we have an additive model) and as discussed above all the process/observation model error terms are assumed to follow a log-normal distribution for this model. The error distributions used in the model include:

* Process error
* Fully-recruited clapper index
* Recruit clapper index (3)
* Fully recruited biomass survey index
* Recruit biomass survey index
* CPUE index

The model parameters also need to have an initial distribution assigned to them; these are known as prior distributions. For some parameters we have a good expectation of what the value of the parameter is. Somewhat informative priors are used for as there is evidence that survey catchability is between 0.2 and 0.5 (based on dredge studies in the USA), a beta distribution is used for this prior (a beta is bounded by 0 and 1 and can take all sorts of wacky shapes. The dissolution rate of clappers is also likely in the range of 0.2-0.7 years so the prior there is also reasonably informative

* survey catchability
* dissolution rate

For a subset of parameters in the model we very have limited information on the range or expected value of the parameter and we attempt to set the prior so it is “uninformative” and allow the parameter value to take on a wide range of values. For example we have little knowledge (or interest in ) the scaling parameter K, but we still need to give it some range of values it can take and this is done by setting up a prior distribution with an extremely large range of potential values (I have no idea why this isn’t just a log-normal distribution). Also we do not have good information regarding the fishery catchability () so the prior for this is intended to be vague as well (uniform between 0 and 1). Because is not a dimensionless number the CPUE we use will impact the prior boundaries, given the current CPUE is much smaller than the biomass used in the model our prior is o.k., but we need to be careful if we ever change the CPUE used here!

* Scaling constant K
* fishery catchability

Other model parameters we are interested in can vary from year to year (much like the above “error” priors) and these get a relatively uninformative prior for each year, again as with the error terms these are log-normal priors. These include:

* Fully-recruited natural mortality
* Recruit natural mortality
* Scaled recruit biomass

1. *Variance parameters*

There is a second subset of priors that is worth its own little section. So far all of the parameters we’ve discussed are used to model the underlying process. But if you notice there are also parameters within the error structure distributions to account for variance which are estimated from the data as well, I’ve labeled this group of distributions Equation (3) on the previous page. The variance parameters found in Equation (3) also have prior distributions also have their own parameters (the parameters of a prior are known as hyper-parameters). For the variance of the process error the prior is set on the standard deviation:

* process error variance prior

For the clapper priors an inverse gamma distribution is used to set the prior on the variance itself. Now I should mention that in the BUGS script you will see these as Gamma but also note that these are gamma priors on the precision, this is equivalent to an inverse Gamma prior on the variance (which is what we want!). In the below specification of a gamma function the “rate” (rate =) parameter is the second parameter value (the first is known as the shape). It appears that for the gamma distribution ( BUGS uses the “scale” parameter whereas in R it is the “rate” parameter, these are simply inverses of each other, but it is important to note and the literature appears either mistaken or ambiguous about this difference. For example in BUGS using dgamma(0,2) is the same as dgamma(x,0,0.5) in R. This appears to lead to no end of confusion in these models.

The upshot is that using these priors should correspond to a coefficient of variation (CV) of approximately 0.5 for the clappers.

* Fully-recruited clapper variance prior
* Recruit clapper variance prior

Excitingly, for the variance parameters of the survey indices (,) and the CPUE index () the situation is a more complex with respect to the priors. First, and rather important to note, the observed CV for the fully-recruited and recruit biomass and CPUE indices are used to account for the uncertainty in these indices, so these variance parameters are estimated from the data unlike any other variance parameter so far. There is a funky little relationship between CV of a log-normal random variable and the variance of its logarithm that is needed to get a good prior of the variance, the general relationship is…

So these CV’s are used to get an estimate of the variance.

* Fully-recruited biomass index variance
* Recruit biomass index variance
* CPUE index variance

Now these variance parameters are also modeled as inverse gamma so this variance needs to be used to set reasonable values for the gamma prior. The resulting priors used are (see Smith and Hubley 2014 ICES article).

* Fully-recruited biomass index variance prior
* Recruit biomass index variance prior
* CPUE index variance prior

1. *Posteriors*

Each Bayesian iteration (once the model is converged) retained gives us one realization of the potential value of a given parameter. When we repeat this for numerous iterations it results in a large number of potential values for each parameter in the model. The range of these values gives us a measure of the uncertainty in the parameter value and the mean/median of all these iterations for a parameter gives us a measure of the ‘most plausible’ value our simulation suggests this parameter will take. For a given parameter each of the iterations taken together provides a summary of the realized distribution of the parameter, this “posterior distribution” is often quite different from the above specified prior distribution. In general when a parameter is “important” in a model (here I mean the model + the data feeding the model) then this means the model has a lot of information about what the value of the parameter will be (put another way, to maximize your likelihood there are some parameters that will necessarily be limited in the range of values they can take). The more ‘important’ a parameter is the more the model will inform the posterior distribution and this posterior distribution will differ notably from the prior distribution. For parameters that are less “important” or that we know very little about the posterior distribution can look essentially identical to the prior distribution. Generally these parameters are non-informative; when your prior looks like your posterior it basically means you have no information in your model to tell you what the correct value of that parameter is so the posterior is simply your prior. Another way to think of this is that there is more information in the prior about the correct value of the parameter than there is in your data. In other cases you can see a posterior distribution is all stacked up at one end of your prior, this often suggests that the model is suggesting the parameter falls outside the range allowed for in the prior and you have constrained the model parameter space such that you are not really maximizing your likelihood, this is a sign you should either change your prior so enable the model to find the correct parameter value or if you are sure the prior is bounded as you suggested then there is something wrong with your model.

1. **BUGS/JAGS Implementation(Model)**

In this section I try to relate the “theory” from above to the actual code in the model, hopefully this helps clarify how the model is put together, but it might not...

* 1. ***Process Model***

The process component of our model consists of two pieces, the main model which comes from Equation (1) and the uncertainty of our Biomass estimate. Notice that we do not have a direct measure of biomass; it is our “hidden state” (process equation) that we are trying to figure out by using all sorts of data the we believe are proxies that can inform us of what that biomass is. Essentially in Equation (1) you could think of our terms as fixed numbers that we know with certainty, now we start to show how each term is actually a random variable that has some level of uncertainty associated with it, starting in this case with . A critical advantage of this framework is that we can start to quantify how confident we are in our overall biomass predictions and in the estimation of our parameters (e.g. natural mortality). Starting with the main model we see the actual model used is slightly different from equation (1)

In terms of code this breaks down into these lines in the model script. Note that for the first year we have to give the model a starting value

# Give Pmed an initial value, note that Pmed is on log scale. Remember # P/Pmed is simply used as a scaling term to help convergence

# and is B/K, where B is biomass and K is the scaling constant

Pmed[1] ~ dnorm(0, 0.1)

# Based on the initial value for Pmed, this gets an initial value for P

P[1] ~ dlnorm(Pmed[1], isigma2)

# Here is our process/state equation from year 2 to the final year (likelihood) for the biomass

Pmed[t] <- log(max(exp(-m[t]) \* g[t-1] \* (P[t-1] - C[t] / K) + exp(-mR[t])\*gR[t-1] \* r[t-1], 0.001))

# Now incorporate the process noise for our estimates of P

P[t] ~ dlnorm(Pmed[t], isigma2)

The differences between this equation and Equation (1) are largely made for computational efficiency except for the term which essentially turns this into a statistical model:

* Rescaled biomass, the biomass needs to be rescaled to help improve

the convergence of the Gibbs sampler (this sampler can be thought of the engine behind the JAGS/BUGS algorithms). The only reason we rescale if so the model converges more easily. is simply multiplied by K to get back to fully recruited biomass.

* Rescaling constant, the biomass needs to be rescaled to help improve

the convergence of the Gibbs sampler (this is kinda the engine behind the JAGS/BUGS algorithms. The only reason we rescale if so the model converges more easily. The exact number is essentially meaningless.

* Rescaled recruit biomass, this needs to be rescaled to help improve

the convergence of the Gibbs sampler (this is kinda the engine behind the JAGS/BUGS algorithms. The only reason we rescale if so the model converges more easily. is simply multiplied by K to get back to actual recruit biomass (.

* The “process error” term, process error is considered the true variation

in the population dynamics, think of this as “real biologically driven” variation due to natural processes. In this case it is the variation in the “true biomass” that we can’t explain with our model. This is opposed to “observation error” which we will discuss below, but is basically the component that is attributable to “errors” that result from imperfect counting (no sampling regime is ever perfect!). The process error is added in multiplicatively (this becomes additive on the log scale and is why we take the log of the equation in the model itself). On the log scale we assumed the model is normally distributed (i.e. we are using a log-normal distribution in the model itself) with a mean of 0, and a variance we can estimate from the data ( . This is a pretty standard formulation for this type of model.

* 1. ***Observation Models***

See above theory section for a brief summary of how the observation models link to a process model.

* + 1. *Clapper Popcorn Model*

The equations used for the clappers are simply equation 2 re-arranged to solve for *Z* and to account for uncertainty in our estimation of Z*,* for the fully recruited clappers this looks like:

For the recruit clappers this looks like:

In terms of code this breaks down into these lines in the model script. Notice that for the first year we need to use a different equation because we don’t have the clappers from the previous year.

# For the first year get the clapper index as Natural mortality \* dissolution rate \* Abundance

Cmed[1]<-log(m[1]\*S\*N[1])

# And same for the recruits

CRmed[1]<-log(mR[1]\*S\*NR[1])

# The clapper process/state equation from year 2 to the final year (likelihood)

for(t in 2:NY)

{

Cmed[t]<-log(m[t]\*S\*(S\*N[t-1]+(2-S)\*N[t])/2)

CRmed[t]<-log(mR[t]\*S\*(S\*NR[t-1]+(2-S)\*NR[t])/2)

}

# And now include the observation error for the clappers.

for(t in 1:NY)

{

clappers[t] ~ dlnorm(Cmed[t],ikappa.tau2)

clappersR[t] ~ dlnorm(CRmed[t],ikappa.rho2)

}

Note that S is assumed to be the same for fully-recruited and recruit clappers in this model, though it is likely that S is lower for smaller sized scallop. The new terms introduced by these equations are simply to deal with the uncertainty of the clapper abundances

* The fully recruited clapper “observation error” term, On the log scale (i.e.

we are using a log-normal distribution in the model itself) we assumed the model is normally distributed with a mean of 0, and a variance we can estimate from the data ( .

* The recruit clapper “observation error” term, On the log scale (i.e.

we are using a log-normal distribution in the model itself) we assumed the model is normally distributed with a mean of 0, and a variance we can estimate from the data ( .

* + 1. *Survey Biomass and CPUE Indices model*

The fully-recruited and recruit biomasses obtained from the survey are the critical observations used to get the model estimated biomasses (for Fully recruited and recruit scallop) in the main process equation. The survey results provide an index of biomass but when coupled with information about some agent of mortality (generally fisheries removals is used though I wonder how different our estimate would look like if we just used a clapper index, I think that would be sufficient to get a biomass estimate) an actual biomass for an area can be estimated. For this model the survey biomass index is assumed to be proportional to the “true” biomass. For the fully recruited biomass this is (note that ) :

while for recruits it is (note that):

In addition the CPUE from the fishery is also used to estimate the fully recruited biomass. It too is assumed to be proportional to the “true” fully recruited biomass:

Here is the JAGS/BUGS code for the observation equations.

# observation models for fully-recruited and recruit biomass indices and the catch rate index.

for(t in 1:NY)

{

# Assumed relationship between observed survey biomass index and "true" biomass

Imed[t] <- log(q \* K \* P[t])

# the observation error for the survey biomass index

I[t] ~ dlnorm(Imed[t], I.precision[t])

# Assumed relationship between observed survey recruit biomass index and "true" recruit biomass

IRmed[t] <- log(q \* K \* r[t])

# the observation error for the survey recruit biomass index

IR[t] ~ dlnorm(IRmed[t], IR.precision[t])

# Assumed relationship between Commercial catch rate index and "true" fully recruited biomass

Umed[t] <- log(qU \* K \* P[t])

# The observation error for the commercial catch rate index.

U[t] ~ dlnorm(Umed[t], U.precision[t])

}

* 1. ***Priors***

The full list of priors in the order they are entered in the model are

* Scaling constant K
* Scaled recruit biomass
* Fully-recruited natural mortality
* Recruit natural mortality
* dissolution rate
* survey catchability
* fishery catchability
* process error variance prior
* Fully-recruited clapper variance prior
* Recruit clapper variance prior
* Fully-recruited biomass index variance prior
* Recruit biomass index variance prior
* CPUE index variance prior

The code to generate these priors are found below. The values of the parameters used for the prior are found in the commented sections (lines that start with a # and in smaller font) for most of the priors. For the variance priors used with the Survey biomasses and CPUE indices the transformations to convert the CV into a vague variance prior are done in the r-script Update\_function\_JAGS.r.

# K is the model "rescaling" constant, this is basically used to help with model convergence.

# Normal on log scale, why aren't we just using a log-normal distribution for K??

# Priors of logK.a = 7 and logK.b =7 results in a median of around 1100, and a narrow range of # potential values

# remember logK.b is the precision on the variance, this results in a rather narrow prior for K

# Ranging from around 500 - 2100.

logK ~ dnorm(logK.a, logK.b)

# Exponentiate to get K back to correct scale.

K <- exp(logK)

# priors for survey recruitment index, note we get get this estimate annually.

# priors of r.a=0 and r.b=1 give a median of 1, mean of 1.6, 90% of distribution density is < 4 and the distribution #does allow for extreme values

for(t in 1:NY)

{

r[t] ~ dlnorm(r.a, r.b)

}

# priors for natural mortality of both recruits and fully recruited, again we estimate these #parameters annually.

# Priors of m.a =-2 and m.b = 2 results in a sampling distribution with a median of around 0.14, and m is generally < 1 though extreme values can happen.

for(t in 1:NY)

{

m[t] ~ dlnorm(m.a, m.b)

mR[t] ~ dlnorm(mR.a, mR.b)

}

# Dissolution rate in days/365, the rate at which clappers disappear from the population

# (clapper connective tissue becomes too weak to hold together)

# A prior of a=8 and b = 11 gives this prior a normal-ish shape with mean around 0.45 and most of the density #between 0.15-0.7

S ~ dbeta(S.a,S.b)

# Survey catchability prior, q priors should be between 0.2 and 0.5 (based on US dredge studies), q.a = 20, q.b = 40 #gets us this result.

q ~ dbeta(q.a, q.b)

# Fishery catchability prior, Unknown so given a uniform between 0 and 1. This proportionality

# coefficent is not unitless (see help file discussion) so there is no reason why this term would be bounded #between 0 and 1, if we change our CPUE

# calculations we need to be aware/careful here that we don't accidentally give this a highly informative prior.

qU ~ dunif(qU.a, qU.b)

# All the variance related Priors start here.

# prior for process noise, uniform between 0 and 5.

sigma ~ dunif(sigma.a, sigma.b)

isigma2 <- pow(sigma, -2)

# Remember for these prior this is being done on the precision, a gamma on the precision is the #same as an inverse gamma on the variance

# There is some debate if the second term in the gammma is a rate or a scale term (rate = #1/scale), it appears that it is scale (in R it is rate)

# so using a value of 2.24 in BUGS gives the same result as using 1/2.24 in R

# ikappa.tau2.a = 3, ikaap.tau2.b = 2.24 gives slightly right skewed prior with a median of 6(ish)

# and maximum values in the 40's.

ikappa.tau2 ~ dgamma(ikappa.tau2.a, ikappa.tau2.b)

kappa.tau <- pow(ikappa.tau2, -0.5) # Standard deviation

# recruitment clappers prior, ikappa.tau2.a = 3, ikaap.tau2.b = 2.24 gives slightly right skewed prior with a median # of 6(ish) and maximum values in the 40's.

ikappa.rho2 ~ dgamma(ikappa.rho2.a, ikappa.rho2.b)

kappa.rho <- pow(ikappa.rho2, -0.5)

# Prior for the precision of the survey biomass index variance (that's a mouthful eh!)

for(t in 1:NY)

{

I.precision[t] ~ dgamma(I.precision.a[t],I.precision.b[t])

# Variance of the survey biomass index variance!

I.vsam[t]<-1/I.precision[t]

# Prior for the precision of the survey recruit biomass index variance...

IR.precision[t] ~ dgamma(IR.precision.a[t],IR.precision.b[t])

# And it's variance

IR.vsam[t]<-1/IR.precision[t]

# Prior for the precision of the Commercial Catch rate (CPUE)...

U.precision[t] ~ dgamma(U.precision.a[t],U.precision.b[t])

# And it's variance

U.vsam[t]<-1/U.precision[t]

}

* 1. ***Diagnostics and Derived Parameters***

The final section of the model focuses on taking the model results and turning those values into some derived parameters of interest. By derived parameters I mean parameters that are some “interesting” combination of model parameters. These include both parameters we have direct interest in (e.g. exploitation rate, model biomass, modeled biomass indices) and parameters that can help perform model diagnostics (e.g. residuals).

The primary derived parameter we obtain is the true biomass estimate () since the model itself is rescaled. Additionally the observation equations produce model fitted predictions for the two biomass indices and the CPUE index (), these give us an indication of how the model is fitting the observed data (the names of these in the below are Ipred, IRpred, and Upred).

The final derived parameters provide us with estimates of the fishing mortality, we get both the exploitation rate () and the instantaneous fishing mortality () which are based on the catch and biomass. The offshore fishery is year round and this makes this calculation a little funky to wrap your head around; more so than you’d first think. For the offshore model we take the total biomass as the catch from the current survey year + the current biomass. Figuring out what the catch for the current survey year is can hurt ones brain as it depends on the survey timing:

* BBn: The survey happens at the end of May, so the current survey year catch is from the previous June to May of the current year.
  + For example, when we run the model to get the 2016 biomass, the catch data used is from June of 2015 to May of 2016. Our biomass estimate is for June of 2016.
* GBa: The survey happens in August, so the current survey year catch is from the previous September to August of the current year.
  + For example, when we run the model to get the 2017 biomass, the catch data used is from September of 2016 to August of 2017. Our biomass estimate is for June of 2017

Model code for dérive paramètres

# Various derived quantities we're interested in.

for(t in 1:NY)

{

# Fully recruited biomass

B[t] <- P[t] \* K

# Recuit biomass

R[t] <- r[t] \* K

# Modelled fully recruited survey biomass estimate

Ipred[t] <- q \* B[t]

# Modelled recruit survey biomass estimate

IRpred[t] <- q \* R[t]

# Modelled fully recruited survey biomass estimate

Upred[t] <- qU \* B[t]

# Exploitation rate

mu[t] <- C[t]/(B[t]+C[t])

# Instantaneous fishing mortality

Fmort[t] <- -log(max(1 - mu[t], 0.0001))

}

***Inshore exploitation rate aside***

The exploitation rate calculations for the inshore and offshore model are actually the same, but there is some confusion about this which I hope to clarify here. In the inshore model the model catch used comes from the fishery that runs from October of the previous year until March of the current year. If you look above at the “survey year catch” explanation you’ll see these two things are very similar, the fishing year starts in the previous year and runs into the year in which you are calculating the biomass. The difference is in what we call these, the easiest way to explain is using example (with fake numbers).

* On Georges Bank “a” from September 2015 to August 2016 we catch 3000 tonnes
  + This is assigned to as catch in 2016 in the offshore model
* In SPA4 from October 2015 to March 2016 we catch 200 tonnes
  + This is assigned as catch in 2015 in the inshore model, no doubt because the fishery views this as the 2015/16 fishery

Hopefully you see that we have assigned catch from essentially the same time period different years in the different models. For this reason the code for the inshore and offshore have to differ even though they are trying to do the exact same calculation

* Offshore: mu[t] <- C[t]/(B[t]+C[t])
* Inshore : mu[t] <- C[t]/(B[t + 1] + C[t])

Note the subscript difference between the models, this is done simply because of the above difference in how we assign the “catch year” as mentioned above.

There is one more issue because of how we sub-script that causes confusion. The exploitation rate mu[t] in the model is given the same sub-script as the catch, so the exploitation rate from the 2015/2016 fishery is called 2015 exploitation rate in the inshore model but it is called 2016 exploitation rate in the offshore model. Critically for the inshore there is a post model step where the exploitation rate is re-labeled from t to t+1 (e.g. 2015 mu from the model output is changed to be 2016 mu). I am strongly of the opinion we should change this directly in the model by simply changing mu[t] to mu[t+1]) this would reduce the confusion and would make this consistent with the offshore model.

There are also a suite of diagnostic derived parameters using the model results. These are largely residual plots of the data. These include residuals of the observation error for fully recruited survey biomass (Iresid), recruit survey biomass (IRresid), fully recruited clapper residual (Cresid), recruit clapper residual (CRresid), and the process error residual (Presid). Standardized residuals are also calculated (dividing raw residual by standard error of the appropriate measure), these are given the same name as above but start with a small ‘s’. Finally the biomass and clapper posteriors are given a unique name; I believe this is done to avoid confusion between input variables (I, IR, clappers, and clappersR) and output parameters for these.

# Diagnostics

for(t in 1:NY)

{

# Standard residuals for fully recruited and recruit biomass along with the Process term

Iresid[t] <- log(I[t]) - Imed[t]

IRresid[t] <- log(IR[t]) - IRmed[t]

Presid[t] <- log(P[t]) - Pmed[t]

# We should also look at the clapper residuals to see how that model does.

Cresid[t] <- log(clappers[t]) - Cmed[t]

CRresid[t] <- log(clappersR[t]) - CRmed[t]

# Same residuals but standardized (remember these are precisions mostly so multiply by sqrt(precision) = divide by standard deviation)

sPresid[t] <- Presid[t] / sigma

sIresid[t] <- Iresid[t] \* pow(I.precision[t],0.5)

sIRresid[t] <- IRresid[t] \* pow(IR.precision[t],0.5)

sCresid[t] <- Cresid[t] \* pow(ikappa.tau2,0.5)

sCRresid[t] <- CRresid[t] \* pow(ikappa.rho2,0.5)

# These give the posteriors for the observation error around survey biomass for fully recruited and the recruits.

# We already have these they are just given a different name for plotting the posteriors in a post-processing stage

# (this is I[t] and IR[t])

Irep[t] ~ dlnorm(Imed[t], I.precision[t])

IRrep[t] ~ dlnorm(IRmed[t], IR.precision[t])

# These give the posteriors for the clapper estimates

# We already have these they are given a different name for plotting the posteriors in a post-processing stage

# (this is clappers[t] and clappersR[t])

Crep[t] ~ dlnorm(Cmed[t],ikappa.tau2)

CRrep[t] ~ dlnorm(CRmed[t],ikappa.rho2)

}

***Required Inputs***

***Priors***

1. logK.a
2. logK.b
3. r.a
4. r.b
5. m.a
6. m.b
7. mR.a
8. mR.b
9. S.a
10. S.b
11. q.a
12. q.b
13. qU.a
14. qU.b
15. sigma.a
16. sigma.b
17. ikappa.tau2.a
18. ikappa.tau2.b
19. ikappa.rho2.a
20. ikappa.rho2.b
21. I.precision.a
22. I.precision.b
23. IR.precision.a
24. IR.precision.b
25. U.precision.a
26. U.precision.b

***Data***

1. NY
2. C
3. g
4. gR
5. N
6. NR
7. clappers
8. clappersR
9. I
10. IR
11. U