**Why mixing model priors matter**

The default prior for most mixing model has a mean of 1/n, where n is  
the number of sources. So if we had five potential food sources this  
means our starting assumption is that on average the consumer eats and  
assimilates 20% of each prey item.

Uncareful use of mixing models has resulted in findings from some  
peer-reviewed being contested. For instance, if the user just puts  
sources in the model ‘just to see’ if they matter, the starting  
assumption is that, on average, each contributes an equal fraction to  
the diet. This starting assumption is in many cases ridicilous.

When the data are not particularly informative, the model will return  
the result that every item contributed an equal fraction to the animals  
diet. The authors may then write this up as a new ‘result’ when in fact  
it was just the default assumption of the software package being  
reflected in their outputs.

In general I am quite suspicious of all the isotope studies reporting  
generalist consumers that eat equal fractions of prey. These patterns  
may well just reflect the default priors.

Adding to the confusion is that some call the default priors  
‘uninformative’.

Priors for mixing models are all informative, eventhe so called  
‘uninformative’ priors. The prior for source contributions bounded  
between 0-1, and source contributions must sum to 1, so it can never be  
truly flat in that range.

**Solutions**

You should always plot your priors and posteriors to check what is going  
on. You can very quickly identify this issue of uninformative data. Then  
in your write up, you could put less emphasis on results that look like  
the priors.

You can also calculate statistics that measure how different prior and  
posterior are. In our paper, we described several statistics taken from  
information theory. You can then easily report these statistics to  
summarize where prior and posterior are different (or not).

For instance, in a [recent study of coastal fish  
specices](https://www.sciencedirect.com/science/article/pii/S0272771419308492?casa_token=Qm8VrU9QEukAAAAA:dMdCtjarurMhWUyoUYdQte1Xy07D00YkdDUQgSUNSRBdSietcm3PPJaIw4yVWkgFTR4bfNl6-iw), we fitted many models across many species and regions, so we reported  
the differences as a table in the supplemental material.

**How to use the R package**

This blog demonstrates how information criteria can be calculated for  
mixing models fit with  
[MixSIAR](https://cran.r-project.org/web/packages/MixSIAR/index.html).

We will apply the simple marginal information criteria from that paper  
to the Killer Whale example, see vignette("killerwhale\_ex") in  
MixSIAR.

The killer whale example is a nice simple one with no covariates or  
random effects. If you have covariates or random effects, you’ll need to  
be careful to compare priors to posteriors at the same locations on the  
fixed/random effects.

It will be helpful to have some understanding of MixSIAR’s data  
structures, because we need to find the posterior samples in the model  
output.

**Killer whale example**

First load the packages we need:

library(BayeSens)

library(MixSIAR)

Now load the data (this is verbatim from the Killer whale example).

mix.filename <- system.file("extdata", "killerwhale\_consumer.csv", package = "MixSIAR")

mix <- load\_mix\_data(filename=mix.filename,

iso\_names=c("d13C","d15N"),

factors=NULL,

fac\_random=NULL,

fac\_nested=NULL,

cont\_effects=NULL)

source.filename <- system.file("extdata", "killerwhale\_sources.csv", package = "MixSIAR")

source <- load\_source\_data(filename=source.filename,

source\_factors=NULL,

conc\_dep=FALSE,

data\_type="means",

mix)

discr.filename <- system.file("extdata", "killerwhale\_discrimination.csv", package = "MixSIAR")

discr <- load\_discr\_data(filename=discr.filename, mix)

**Draw samples from the prior**

Let’s draw samples from the prior. You can also plot this with MixSIAR’s  
plot\_prior function, but we need a matrix of the samples for  
calculating info criteria later.

alpha <- rep(1, source$n.sources) #default prior values

p\_prior <- MCMCpack::rdirichlet(10000, alpha) #draw prior samples

Let’s plot just the prior for the first source (since they are all the  
same in this case)

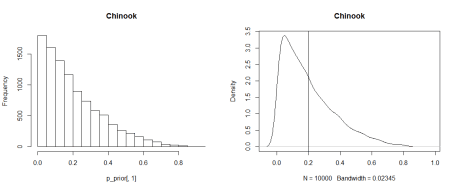
#Plot histogram and density (same data, different ways to view it )

par(mfrow = c(1,2))

hist(p\_prior[,1], 20, main = source$source\_names[1])

plot(density(p\_prior[,1]), main = source$source\_names[1])

abline(v = 1/source$n.sources)



As you can see the default prior clearly isn’t ‘uninformative’ because  
it is centred around 1/number of sources (in fact it has mean 1 over the  
number of sources). It might be better called the ‘uninformed’ (by the  
user) prior. This means the prior will have a lower mean the more  
sources you include in the model.

**Run the model**

This is verbatim from the Killer Whales example.

model\_filename <- "MixSIAR\_model\_kw\_uninf.txt" # Name of the JAGS model file

resid\_err <- TRUE

process\_err <- TRUE

write\_JAGS\_model(model\_filename, resid\_err, process\_err, mix, source)

jags.uninf <- run\_model(run="test",mix,source,discr,model\_filename,alpha.prior = alpha, resid\_err, process\_err)

## module glm loaded

## Compiling model graph

## Resolving undeclared variables

## Allocating nodes

## Graph information:

## Observed stochastic nodes: 12

## Unobserved stochastic nodes: 23

## Total graph size: 766

##

## Initializing model

I’ve used the test run mode here just to speed things up for the  
example.

You should absolutely use long chains (e.g. run = "long") when  
calculating info criteria. They are quite sensitive to the number of  
MCMC samples if there are few samples. We need enough samples to get a  
good idea of the posteriors full shape.

**Extract samples**

Here’s where it helps to have some idea of how MixSIAR structures  
outputs. We need to find the posterior samples. You can dig around using  
str(jags.uninf). I did that and found the samples under  
jags.uninf$BUGSoutput as below:

p\_post <- jags.uninf$BUGSoutput$sims.list$p.global

Now we have a matrix of prior samples and a matrix of posterior samples  
we can just compare them with the hellinger or kldiv  
(Kullback-Leibler divergence) functions from BayeSens. I’ll compare  
just the first source (Chinook salmon).

hellinger(p\_prior[,1], p\_post[,1])

## Hellinger distance - continuous

## [1] 0.61

##

## Hellinger distance - discrete

## [1] 0.65

kldiv(p\_prior[,1], p\_post[,1])

## Kullback-Leibler divergence

## [1] 5.7

We’d like to know what the info criteria are for all sources, so we  
could manually select columns to compare, or just use some sort of  
iterating function to do them all at once. Here I use lapply and put  
them into a dataframe:

hell\_out <- lapply(1:source$n.sources, function(i) hellinger(p\_prior[,i], p\_post[,i])$hdist\_disc)

kl\_out <- lapply(1:source$n.sources, function(i) kldiv(p\_prior[,i], p\_post[,i])$kd)

info\_df <- data.frame(source\_names = source$source\_names,

hellinger = unlist(hell\_out),

KLD = unlist(kl\_out))

info\_df

## source\_names hellinger KLD

## 1 Chinook 0.6536184 5.6601980

## 2 Chum 0.3659136 1.6746118

## 3 Coho 0.3053054 0.9676074

## 4 Sockeye 0.5260229 3.6985169

## 5 Steelhead 0.5430961 4.2273867

Hellinger values near 0 are very similar to the priors, Hellinger values  
near 1 are very different to the priors. The KLD ranges from >0 to  
infinity, so greater values indicate greater differences from the prior.  
So these results indicate to us that the model and data are not very  
informative about Coho, but much more informative about Chinook. To  
interpret why this is you should plot the priors and posteriors.

You can use output\_JAGS to do this. We will do it ourselves, just to  
practice data wrangling. For Chinook and Coho:

par(mfrow = c(1,2))

plot(density(p\_post[,1]), main = source$source\_names[1])

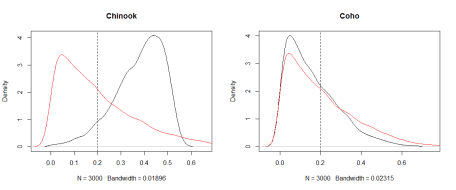
lines(density(p\_prior[,1]), col = "red")

abline(v = 1/source$n.sources, lty = 2)

plot(density(p\_post[,3]), main = source$source\_names[3])

lines(density(p\_prior[,3]), col = "red")

abline(v = 1/source$n.sources, lty = 2)



It is pretty clear that contributions for Chinook have shifted higher,  
whereas the data doesn’t give us much reason to believe Coho are any  
more important than the prior suggested.

Note that you can also get high information criteria stats if the  
posterior mean stays the same as the prior’s mean, but the distribution  
changes shape (e.g. gets thinner). For instance, if the data were  
strongly informative that Coho were not an important food source, then  
we could have the same posterior mean of 0.2, but the uncertainty  
intervals would be much narrower around 0.2 than in the prior.

**Informative priors**

The killer whale example also gives a model fit with informed priors.  
Here’s the code verbatim from MixSIAR:

kw.alpha <- c(10,1,0,0,3)

kw.alpha <- kw.alpha\*length(kw.alpha)/sum(kw.alpha)

kw.alpha[which(kw.alpha==0)] <- 0.01

model\_filename <- "MixSIAR\_model\_kw\_inf.txt"

resid\_err <- TRUE

process\_err <- TRUE

write\_JAGS\_model(model\_filename, resid\_err, process\_err, mix, source)

jags.inf <- run\_model(run="test",mix,source,discr,model\_filename,alpha.prior=kw.alpha, resid\_err, process\_err)

## Compiling model graph

## Resolving undeclared variables

## Allocating nodes

## Graph information:

## Observed stochastic nodes: 12

## Unobserved stochastic nodes: 23

## Total graph size: 766

##

## Initializing model

The only extra step we need to do now is draw samples from the prior and  
posteriors:

p\_prior\_inf <- MCMCpack::rdirichlet(10000, kw.alpha) #draw prior samples

p\_post\_inf <- jags.inf$BUGSoutput$sims.list$p.global

hell\_out\_inf <- lapply(1:source$n.sources, function(i) hellinger(p\_prior\_inf[,i], p\_post\_inf[,i])$hdist\_disc)

## Warning in sqrt(1 - integrate(fx1, minx, maxx)$value): NaNs produced

## Warning in sqrt(1 - integrate(fx1, minx, maxx)$value): NaNs produced

kl\_out\_inf <- lapply(1:source$n.sources, function(i) kldiv(p\_prior\_inf[,i], p\_post\_inf[,i])$kd)

info\_df <- cbind(info\_df,

data.frame(

hellinger\_inf = unlist(hell\_out\_inf),

KLD\_inf = unlist(kl\_out\_inf)))

info\_df

## source\_names hellinger KLD hellinger\_inf KLD\_inf

## 1 Chinook 0.6536184 5.6601980 0.7829688 8.7014174

## 2 Chum 0.3659136 1.6746118 0.3353133 1.0081589

## 3 Coho 0.3053054 0.9676074 0.1208581 0.1794651

## 4 Sockeye 0.5260229 3.6985169 0.1262016 0.1923406

## 5 Steelhead 0.5430961 4.2273867 0.6469540 5.7095752

The warning about NA’s comes from the prior for some groups being near  
zero, so the continuous version of the Hellinger stat isn’t able to be  
calculated. We are using the discrete version though, so its no problem  
to us.

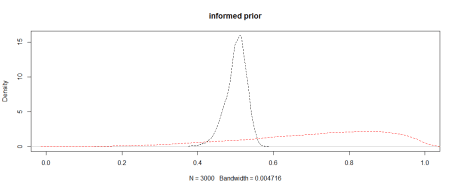
So with the informed priors the Hellinger has increased for Chinook and  
Steelhead and decreased for the others.

Remember that the information criteria just measure the distance from  
the prior. So if our data just confirm the informed priors, or there  
isn’t enough data to overcome the informed priors, then the information  
criteria will be near zero. In this case we have only two tracers and  
samples from 12 killer whales. The prior we used on Sockeye was very  
strong to zero consumption, so our result stays the same.

The below plot shows the priors in red and posteriors in black for the  
model with informed priors.

plot(density(p\_post\_inf[,1]), lty = 2, main = "informed prior", xlim = c(0,1))

lines(density(p\_prior\_inf[,1]), lty = 2, col = "red")



I haven’t plotted the informed Coho model because both the prior and  
posterior are a spikes near zero.

You can clearly see the model has shifted the consumption of Coho  
downwards relative to the informed prior.