MixSIAR GUI User Manual v3.1

Brian Stock and Brice Semmens

last update: April 2018

Scripps Institution of Oceanography, UC San Diego

b1stock@ucsd.edu and semmens@ucsd.edu

Please cite as:

Stock, B. C. and B. X. Semmens (2016). MixSIAR GUI User Manual. Version 3.1. https://github.com/brianstock/MixSIAR/. doi:10.5281/zenodo.47719.

Contents

1 Introduction

1.1 What is the MixSIAR GUI?

The MixSIAR GUI is a graphical user interface (GUI) that helps you create and run Bayesian mixing models to analyze biotracer data, following the MixSIAR model framework [23]. Both the GUI and script versions are written in the open source languages R [22] and JAGS [21], are freely available online, and can be installed on machines running Mac OS X, Microsoft Windows, and Linux.

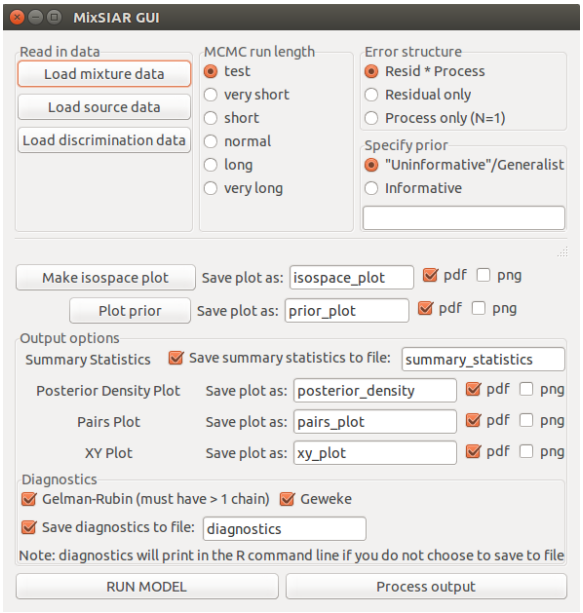


Figure 1: MixSIAR GUI screenshot

1.2 What is the MixSIAR model framework? How does it relate to SIAR and MixSIR?

Analysis of biotracers (i.e. stable isotopes, fatty acids)[1] has become an important tool to ecologists, since it can be used to determine diet composition, population structure, and animal movement. Mixing models use biotracer data to estimate the proportions of source (prey) contributions to a mixture (consumer)[2]. Bayesian mixing models improve upon simpler linear mixing models by explicitly taking into account uncertainty

[1] Mixing models were originally developed using stable isotopes, but they are general statistical models that apply to any mixing process. We make a conscious effort to keep terminology general by using *biotracers* throughout—can be stable isotopes, fatty acids, compound-specific stable isotopes, element concentrations, or color coefficients from diffuse reflectance spectrometry (used in sediment mixing models).

[2] Likewise maintaining general terminology, we use *mixture* and *source* throughout. In diet studies the mixture data are consumers/predators and source data are prey. In sediment studies the mixture data is a downstream soil sample and the source data are upstream sediments (e.g. pasture, forest, cropland).

in source values [15, 27], categorical and continuous covariates [24, 5, 18], and prior information [15]. While diet analysis using stable isotope data is the most common application of mixing models, MixSIAR can also be applied to solve many other environmental questions. To name a few: pollutant sourcing, determining carbon sources for soils, determining carbon sources for ecosystem respiration, and calculating plant water use from soil horizons.

MixSIAR incorporates several years of advances in Bayesian mixing model theory since MixSIR and SIAR:

- MixSIR (original Bayesian mixing model, MATLAB GUI) [15]
- SIAR (residual error, R package, support group) [19]
- Population structure (categorical covariate) [24]
- Uncertainty in discrimination[3] values, concentration dependence[4] [19]
- Continuous covariate [5]
- Source tracer covariance [9]
- Sources and consumers changing through time (via P-splines) [18]
- Multiplicative error structure [25]

MixSIAR is not a single mixing model—it is a general framework that can create many different models based on user data and choices (see Model options). MixSIAR represents a collaborative coding project between the investigators behind MixSIR and SIAR: Brice Semmens, Brian Stock, Eric Ward, Andrew Parnell, Donald Phillips, Andrew Jackson, Stuart Bearhop, and Richard Inger. [23]

1.3 Model options

The current implementation of MixSIAR includes the following model options:

- Any number of biotracers (examples with 1 and 2 isotopes, 8 and 22 fatty acids)
- Source data fit hierarchically within the model
- Source data by categorical covariate (e.g. sources by Region)
- Categorical covariates (up to 2, choice of modeling as random or fixed effects, either nested or independent)

[3] We use *discrimination* to refer to the differences between biotracer values found in the mixture and sources. In diet analysis, these differences are also refered to as 'fractionation', 'trophic enrichment factors' ('TEF'), and 'trophic discrimination factors' ('TDF').

[4] Concentration dependence occurs when sources contain variable amounts of the biotracer element. See Phillips and Koch [20]

- Continuous covariate (up to 1)
- Error structure options with covariance (Residual * Process, Residual only)
- Concentration dependence
- Ability to plot and include "uninformative"/generalist or informative priors

Any combination of these options **creates a different model**, which is then fit to the data. Each of the above options are demonstrated with various examples (see Working Examples).

1.4 Great! How do I get started using MixSIAR?

****Warning** Not so fast!**

We have created the MixSIAR GUI because we feel it can be a useful tool for ecologists wishing to conduct stable isotope analysis on their data. However, it can also be easily abused, since there are a number of ways to receive completely fallacious results:

- MCMC chains not converged
- Mixing model assumptions not met (i.e. missing a source, incorrect TDF)
- Incorrect model options selected (fitting a nonsensical model)

Before using MixSIAR ouput, you should be able to comfortably explain to someone how you've considered and ruled out these concerns. Take some time to understand the assumptions and limitations of mixing models and Bayesian methods (e.g. how MCMC works, what is a prior vs. a posterior) before beginning. See the last page for several links we think are helpful to someone new to these topics. Once you know the basics of what the model does, you're ready to run the working examples. After you are confident MixSIAR is installed and working correctly, you can then begin using it to analyze your own data.

2 Installation

The MixSIAR package can be run as a GUI with `mixsiar_gui()`, or as a sequence of R commands (`.R` script). The GUI depends on the `gWidgetsRGtk2` package in R. MixSIAR GUI has been tested and runs on Windows, Mac OS X, and Linux. Previously Mac users commonly reported install problems, but those seem to be resolved now. If all else fails, the script version should always work! If you have install issues, feel free to contact the authors at https://github.com/brianstock/MixSIAR/issues.

While the GUI may be convenient for users less familiar with R, we advise using the script version of MixSIAR for several reasons:

1. *Repeatability*: You can run different models and have a record of the commands that created each one. There are many reasons you'd want to do this. For example, you may want to compare model results with an uninformative prior vs. an informative prior, one error term option vs. another, grouping sources a priori vs. a posteriori, etc.
2. *Speed*: Clicking through the GUI buttons can get onerous after a while. You can modify the included example .R files and run your entire analysis with `source("your_file.R")`.
3. *Installation ease*: Some users aren't able to install the GTK+ software that the GUI depends on (more issues on Mac). It may be worth figuring out the script version (R skills!) instead of figuring out how to get GTK+ installed.

2.1 Windows

1. *Download and install/update R*. You can check which version of R (and any packages) you are running using the command `sessionInfo()`. If you have an older version of R installed, the easiest thing to do is use the installr package.
2. *Download and install* JAGS.
3. (Optional) *If you want to build the vignettes, install pandoc or R Studio*.
4. *Open R (or R Studio)*.
5. *Install GTK+ dependent packages*.

```
install.packages(c("gWidgets", "RGtk2", "gWidgetsRGtk2", "devtools"))
```

6. *Load* `RGtk2`. **You will be prompted to install GTK+. **Follow the automatic prompts and do not interrupt the GTK+ installation!**:**

```
library(RGtk2)
```

7. *Restart R.*
8. *Install MixSIAR package:*

```
library(devtools)
devtools::install_github("brianstock/MixSIAR",
                         dependencies = TRUE,
                         build_vignettes = TRUE) # FALSE if no pandoc/R Studio
```

8. *Load MixSIAR and run GUI:*

```
library(MixSIAR)
mixsiar_gui()
```

The MixSIAR GUI should then appear as a separate window! If you have problems installing GTK+, search your computer for all listings of "GTK" and delete/uninstall them (including the `RGtk2` and `gWidgetsRGtk2` package libraries in your R folder), then try again. You can also download GTK+ directly from `http://www.gtk.org/download/index.php`, but it's very tricky to manually get GTK+ and R to interface properly (you may have to adjust your PATH variable).

2.2 Mac OS X

We have had reports in the past of installation issues on Mac, but the following consistently works now:

1. *Download and install/update R*. You can check which version of R (and any packages) you are running using the command `sessionInfo()`.
2. *Download and install* JAGS.
3. (Optional) *If you want to build the vignettes, install pandoc or R Studio*.
4. *Open R (or R Studio)*.
5. *Install GTK+ dependent packages*.

```
install.packages(c("gWidgets", "RGtk2", "gWidgetsRGtk2", "devtools"))
```

6. *Close R.*
7. *Download and install the newest GTK+ framework*.
8. *Install the latest X11 application, xQuartz*.
9. *Open R and run:*

```
library(devtools)
devtools::install_github("brianstock/MixSIAR",
                         dependencies = TRUE,
                         build_vignettes = TRUE) # FALSE if no pandoc/R Studio
```

10. *Load MixSIAR and run GUI:*

```
library(MixSIAR)
mixsiar_gui()
```

2.3 Linux

1. *Download and install/update R*. You can check which version of R (and any packages) you are running using the command `sessionInfo()`.
2. *Download and install* JAGS. Or, from the terminal: `sudo apt-get install jags r-cran-rjags`.
3. *Download and install* GTK+. Or, from the terminal: `sudo apt-get install libgtk2.0-dev`.
4. (Optional) *If you want to build the vignettes, install pandoc or R Studio*.
5. *Open R (or R Studio)*.
6. *Check if GTK+ is installed correctly*. Open R, install and load the RGtk2 package with:

```
install.packages("RGtk2")
library(RGtk2)
```

7. *Install and load devtools, then install MixSIAR*:

```
install.packages("devtools")
library(devtools)
devtools::install_github("brianstock/MixSIAR",
                         dependencies = TRUE,
                         build_vignettes = TRUE) # FALSE if no pandoc and pandoc-citepr
```

8. *Load MixSIAR and run GUI:*

```
library(MixSIAR)
mixsiar_gui()
```

3 Running the working examples

The working examples will familiarize yourself with MixSIAR, and it's a good idea to walk through them before playing around with your own data. This manual steps through the examples with the GUI (below), as well as the script version. The script examples are also written as vignettes, which you can access via:

```
library(MixSIAR)
browseVignettes("MixSIAR")
```

The basic MixSIAR workflow is:

1. Load data
    (a) Mixture
    (b) Sources
    (c) Discrimination (TDF)
2. Plot data and prior
3. Choose model structure options
4. Choose output options
5. Run model
6. Use diagnostics to decide if the model has converged
7. Analyze output

A summary of the working examples and MixSIAR options they demonstrate is below:

Dataset	Tracers	Random effects	Fixed effects	Continuous effect	Source data	Error structure	Informative prior	Additional features
Wolves	2 isotopes	Region, Pack (nested)	—	—	Means/SD/n (by Region)	Resid * Process	—	—
Geese	2 isotopes	—	Group	—	Means/SD/n	Resid	—	Concentration dependence
Lake	2 isotopes	—	—	Secchi:Mixed	Raw	Resid	—	—
Palmyra	2 isotopes	—	Taxa	—	Raw	Resid * Process	—	—
Killer whale	2 isotopes	—	—	—	Means/SD/n	Resid * Process	Fecal contents	—
Storm-petrel	2 isotopes	—	Region	—	Raw	Resid	—	—
Snail	1 isotope	—	—	—	Raw	Resid * Process	—	—
Isopod	8 fatty acids	Site	—	—	Means/SD/n	Resid * Process	—	—
Cladocera	22 fatty acids	—	ID	—	Means/SD/n	Process	—	—
Mantis	2 isotopes	—	Habitat	—	Means/SD/n	Resid * Process	Expert opinion, Prey abundance	Combine sources *a posteriori*
Alligator	2 isotopes	—	Sex, Size class	Length	Means/SD/n	Resid * Process	—	Compare models with LOO/WAIC

Table 1: Working examples included with MixSIAR.

3.1 Wolves Example

The "Wolves Example" uses data reconstructed from (not identical to) Semmens et al [12]. Here, we investigate the diet of 66 wolves in British Columbia with:

- 2 biotracers (δ^{13}C, δ^{15}N)
- 2 random effects (Region and Pack), where Pack is nested within Region
- Source data as means and SDs (by Region)
- Resid * Process error

3.1.1 Loading mixture data

1. Click "Load mixture data", then "Load mixture data file" in the new window.
2. Choose "wolves_consumer.csv". This file has the δ^{13}C and δ^{15}N isotope values, and two categorical covariates (Region and Pack). The covariates don't have to be numerical, these just happen to be.
3. You should see the column headings of the mixture data file in the window. Select which columns are isotopes (d13C and d15N) and which are random effects (Region, Pack). Click "I'm finished".
4. Another box will appear asking if the data is nested/hierarchical. Choose "Pack within Region" and then click "I'm finished".

3.1.2 Loading source data

1. Click "Load source data".
2. If you look at the source data file ("wolves_sources.csv"), you will see that each Region has different isotope values—Click "Yes", source data vary by Region, and "No", source data do not vary by Pack.
3. There is no concentration dependence data in "wolves_sources.csv", so "Do you have Concentration Dependence data?" should be "No".
4. In this example we only have source summary statistics (Mean, SD, and sample size), not the original "raw" data—Click "Load source means and SDs".
5. Choose "wolves_sources.csv" when prompted for the data file. *Note that the source data file has a column titled "n" with the sample size of each source estimate. This must be in your source data file when you run your data!*
6. Click "I'm finished".

3.1.3 Loading discrimination data

1. Click "Load discrimination data"

2. Choose "wolves_discrimination.csv" when prompted for the discrimination data file. We use 'discrimination' to refer to the differences between isotope values found in the mixture and sources. In diet analysis, these differences are also refered to as 'fractionation', 'trophic enrichment factors' ('TEF'), and 'trophic discrimination factors' ('TDF').

3.1.4 Making an isospace plot

Once the mixture, source, and discrimination data are loaded, you can click "Make isospace plot". Your plot should match that of Figure 2. If you want to save the isospace plot as a .pdf or .png, make sure either/both of the appropriate boxes are checked. You can also change the name of the file here (default is to save the plot as "isospace_plot.pdf").

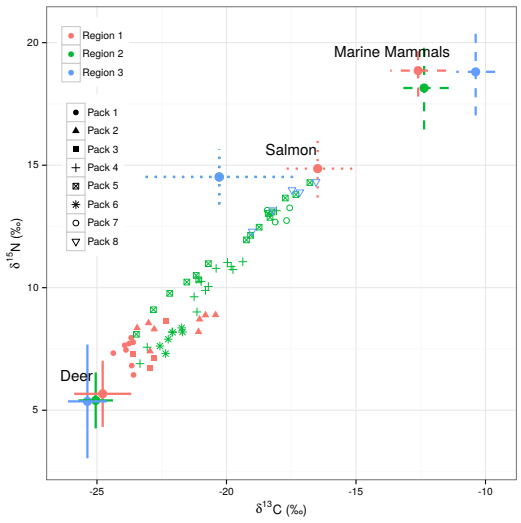


Figure 2: **Stable isotope input for the Wolves Example.** Mixture data (wolves) are by Region (color) and Pack (shape). Source data are by Region and have been adjusted by discrimination means **and SDs**. Error bars indicate ± 1 SD, the combined source+discrimination SD calculated as $\sqrt{\sigma^2_{source} + \sigma^2_{discr}}$, under the assumption of independence.

You should ALWAYS look at the isospace plot—this is a good check that the data is loaded correctly, and that the isospace geometry makes sense. If the mixture data are well outside the source polygon, you have a serious violation of mixing model assumptions, and it must be true that either 1) You're missing a source, or 2) You're using an incorrect discrimination factor. MixSIAR, like SIAR, fits a residual error term, and thus will always find a solution *even if it is nonsensical*.

Also note that the MixSIAR isospace plot adds the discrimination means AND SDs to the raw source values, since the model uses the source+discrimination values to fit the mixture data. Error bars indicate $\pm$ 1 SD[5]. The "Wolves Example" uses 2 isotope values (δ^{13}C and δ^{15}N), as is most often the case. If you have only 1 biotracer, MixSIAR will create a 1-D plot (see Snail Example). If you have more than 2 biotracers, MixSIAR will create all possible pairwise 2-D plots (see Isopod Example for a case with 8 biotracers).

3.1.5 Plot your prior

Click the "Plot prior" button, and you should get a plot that matches Figure 3.

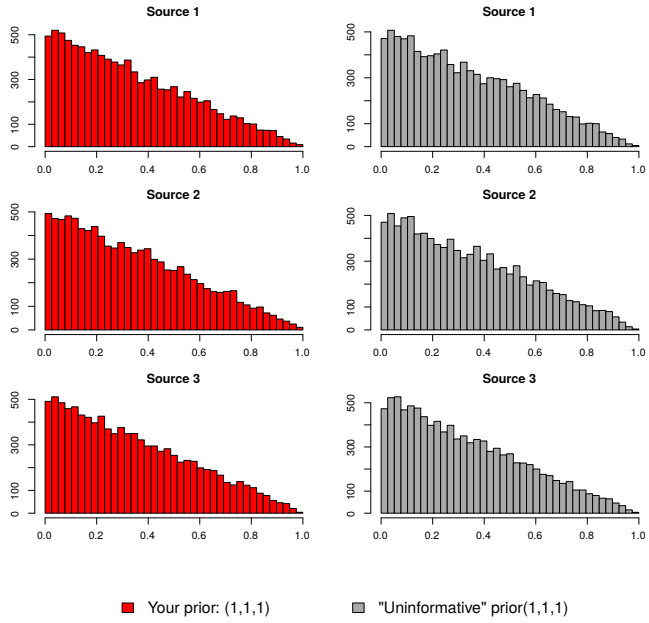


Figure 3: **Uninformative prior used in Wolves Example**. A few things to note: 1) the proportion distributions are the same for all sources, 2) the mean is $\frac{1}{n.sources} = \frac{1}{3}$, and 3) lower proportions are more likely.

Bayesian analyses require priors, and MixSIAR includes a `plot_prior` function to plot the prior on the mixture (diet) proportions (at the highest hierarchical level, `p.global`). The prior represents our knowledge about the proportions *before we consider the biotracer data*. A natural tendency is to want a flat/"uninformative" prior, where all values between 0 and 1 are equally likely. However, because proportions are not independent, there is no truly uninformative prior (e.g. the histograms in Figure 3 are not flat). The best we can do with the Dirichlet distribution is set α = `c(1,1,1)`, which is *uninformative on the simplex*. In other words, all *combinations* of the proportions are equally likely. See the section titled "Constructing informative Bayesian priors" in Semmens et al. [23].

[5]The combined source+discrimination SD is calculated as $\sqrt{\sigma^2_{source} + \sigma^2_{discr}}$, under the assumption of independence.

Because the mean of the "uninformative" prior, α = `c(1,1,1)`, is $\frac{1}{n.sources}$, we also call it the *generalist* prior. This reflects two facts: 1) it is not really uninformative, and 2) for weakly informative data it shifts the posterior towards a generalist diet, $p_1 = p_2 = p_3 = \frac{1}{3}$. The amount of shift depends on the informativeness (quality and quantity) of the data. Brett [2] showed significant shifts occur when the source biotracer values are very uncertain and you only have 1 mixture datapoint.

3.1.6 Specify MCMC parameters

It is beyond the scope of this manual to explain Markov Chain Monte Carlo (MCMC) methods—see 5 for more information. What you need to know to effectively use the MixSIAR GUI can be summarized in a paragraph or two though:

Running MCMC in JAGS can take a long time for a model with many parameters. It's a good idea to always run the "test" length first. This lets you confirm the model is specified correctly, and will give you an idea how long a longer MCMC run will take. **For now, leave "test" selected. If you want, skip ahead to 3.1.7.**

Briefly, MCMC is a method of estimating the probability density functions of variables of interest (e.g. proportion of Region 1 Wolves diet that is Deer). Instead of only a mean and variance, MCMC estimates the entire distribution for each variable. From this estimated "posterior" distribution we can then calculate familiar statistics like mean/median, standard deviation, and Bayesian credible intervals (NOT 95% confidence intervals, see 5). As you increase the number and length of the MCMC "chains", they will converge on the true posterior distribution for each variable. But more and longer chains take...longer, and we don't have all day. If your chains are 'long enough,' according to some diagnostics (see 3.1.9), then you will get accurate estimates of the posterior distributions and we say the chains have "converged." Setting the MCMC parameters is a balancing act between not waiting forever for an answer and achieving convergence.

The MCMC settings available in the GUI are in Table 2. You can specify exact MCMC parameters using the script version (see lines 127-137 of `mixsiar_script.r`). If you're unfamiliar with MCMC, you may want to briefly skim the references in Introductions to Bayesian Analysis.

	Chain Length	Burn-in	Thin	# Chains	Est. runtime
test	1,000	500	1	3	5 sec
very short	10,000	5,000	5	3	3 min
short	50,000	25,000	25	3	15 min
normal	100,000	50,000	50	3	30 min
long	300,000	200,000	100	3	90 min
very long	1,000,000	500,000	500	3	5 hours
extreme	3,000,000	1,500,000	500	3	15 hours

Table 2: **MCMC settings in MixSIAR GUI**. You can specify exact MCMC parameters using the script version (see lines 127-137 of `mixsiar_script.r`). Estimated runtime listed is for the Wolves Example on my laptop (Ubuntu 14.04, Intel Core i5 1.7GHz, 8GB RAM).

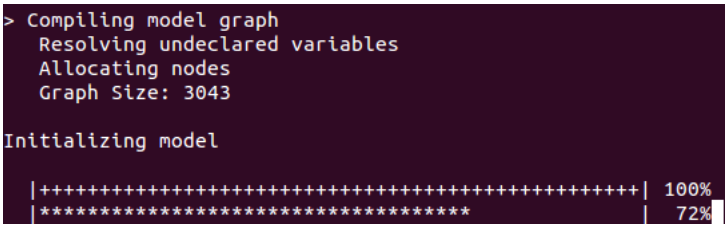
3.1.7 Error structure options

In the Wolves Example we want the "Residual * Process" error option, and it is already selected. The differences between "Residual * Process", "Residual only", and "Process only" are explained in Stock and Semmens [25].

3.1.8 Run MixSIAR (test)

In the future you may want to change the output saving options, but not now. Click the "RUN MODEL" button at the bottom!

You should see something like this in the R console:

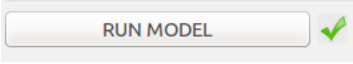


This is good! It means MixSIAR has created a JAGS model file, passed your data to JAGS, and JAGS is fitting your model. The "test" MCMC run should only take a few seconds. If you're familiar with JAGS and want to check out the model file, "MixSIAR model.txt" will be in your working directory folder.

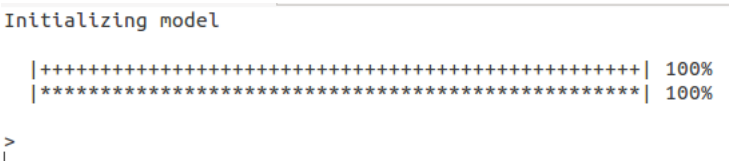
Before we look at the output, it is useful to have at least a hand-wavy understanding of how the MixSIAR model is set up. In the introduction, we stated that MixSIAR is a Bayesian mixing model. This is true, but we can also add another descriptive term and say that MixSIAR is a hierarchical Bayesian mixing model. The hierarchical structure allows MixSIAR to analyze diet by covariates, as illustrated in Figure 4.

3.1.9 Using MCMC diagnostics

When the model is finished, a green check will appear next to the "RUN MODEL" button.



The R console will have two progress bars at 100% as well:



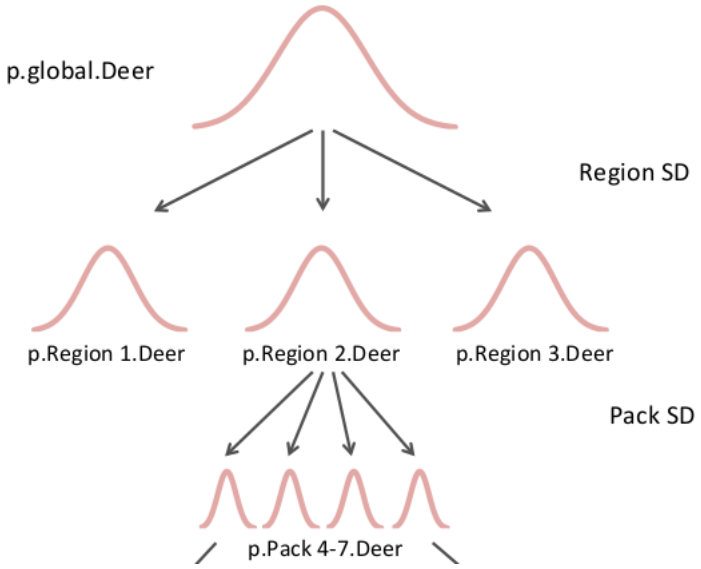


Figure 4: **Model structure and variables in the Wolf Example.** For the diet proportion, *p*, of each source (Deer shown here), we fit a distribution for all wolves (p.global.Deer). Means for each Region (p.Region 1-3.Deer) are drawn from this p.global.Deer distribution, with deviation Region.SD. The process repeats for the Pack level: means for each Pack are drawn from their Region distributions (p.Pack 4-7.Deer from p.Region 2.Deer), with deviation Pack.SD. From Semmens et al. [24].

Now click the "Process output" button. A bunch of plots will be spit out, but first recall that the aim of MCMC is to converge on the posterior distributions for all variables in the model. *Before accepting any of the MixSIAR results, it is imperative that you determine the model has converged.* You should use the diagnostic tests and plots in this effort. MixSIAR prints the diagnostic tests in the R console (above Summary Statistics) and saves them as "diagnostics.txt". Check them out!

MixSIAR displays 2 diagnostic tests by default: Gelman-Rubin and Geweke. Briefly, the Gelman-Rubin test needs > 1 chain to be calculated, and will be near 1 at convergence. Gelman states that "values below 1.1 are acceptable for most examples" [8]. The Geweke test is a two-sided z-test comparing the mean of the first part of the chain with the mean of the second part. At convergence these means should be the same, and large absolute z-scores indicate rejection. The SAS website has a good brief explanation of how to visually analyze trace plots, and Patrick Lam's notes may also be useful in understanding the diagnostic tests.

Looking at the "diagnostics.txt" file, our model is not even close to convergence with the "test" settings:

```
################################################
# Gelman-Rubin Diagnostic
################################################

Generally the Gelman diagnostic should be < 1.05

Out of 41 variables: 41 > 1.01
                     39 > 1.05
                     33 > 1.1

The worst variables are:

            Point est. Upper C.I.
p.global[2]   8.495770  20.363330
p.fac2[5,2]   7.645885  16.154276
p.fac2[5,3]   7.011953  15.151394
p.fac2[4,3]   6.957779  16.236915
```

MixSIAR also produces diagnostic plots useful for determining convergence, using the `ggmcmc` package [12]. Open "diagnostics.pdf" and you will find:

- Gelman-Rubin plot (same as in diagnostics.txt)
- Geweke plot (same as in diagnostics.txt)
- Posterior density plots *by chain*
- Traceplots
- Running mean plots
- Autocorrelation plots
- Crosscorrelation plot

3.1.10 Run MixSIAR (for real)

Let's choose a longer MCMC run and try again:

1. Select "Normal" under "MCMC run length".
2. Click the "RUN MODEL" button again. This will take much longer, about 30 minutes instead of 5 seconds.

Note: MixSIAR normalizes the mixture and source tracer data before running the JAGS model. This most likely should not concern or interest you. If it does, see section Run JAGS model.

3.1.11 Check diagnostics (again)

When the model is finished, again click the "Process output" button. This time it looks like our model is pretty well converged according to the Gelman-Rubin:

```
################################################
# Gelman-Rubin Diagnostic
################################################

Generally the Gelman diagnostic should be < 1.05

Out of 41 variables: 8 > 1.01
                     1 > 1.05
                     0 > 1.1

The worst variables are:

            Point est. Upper C.I.
p.global[3]   1.063742   1.171759
p.fac1[1,3]   1.033035   1.089155
p.fac1[1,1]   1.024669   1.078329
fac2.sig      1.019601   1.034644
p.global[2]   1.019084   1.064416
p.fac1[3,3]   1.013102   1.040326
```

Looking at the Geweke, we can see that chains 1 and 2 are good, but chain 3 was not quite converged:

```
################################################################################
# Geweke Diagnostic
################################################################################

The Geweke diagnostic is a standard z-score, so we'd expect 5% to be outside +/-1.96
Number of variables outside +/-1.96 in each chain (out of 41):

       Chain 1 Chain 2 Chain 3
Geweke       2       6      14
```

I would run this longer (run="long" or "very long") if it were for a final analysis, but this is good for now.

3.1.12 Interpreting MixSIAR output

Several plots are created:

- Posterior plot comparing the variation in diet by each factor (Figure 5, "posterior_density_SD.pdf
- Posterior plots of diet by Region (Figure 6, "posterior_density_diet_p_Region 1.pdf" etc.)
- Posterior plots of diet by Pack (Figure 7, "posterior_density_diet_p_Pack 4.pdf" etc.)
- Posterior plot of overall population diet (Figure 8, "posterior_density_diet_p_global.pdf")
- Pairs plot of overall population diet (Figure 9, "pairs_plot.pdf")

In addition to the plots, summary statistics are output to the R console and saved if you check the box. First, let's look at the medians (50% quantiles) of **Region.SD** and **Pack.SD** terms—these are the variation in diet for Factor 1 (Region) and Factor 2 (Pack):

```
##################################################################
# Summary Statistics
##################################################################

DIC = 175.6317

                  Mean    SD  2.5%    5%   25%   50%   75%   95% 97.5%
Region.SD        1.176 1.054 0.053 0.097 0.463 0.920 1.566 3.198 3.947
Pack.SD          1.304 0.583 0.595 0.655 0.918 1.182 1.556 2.364 2.740
```

You should have values close to $\sigma_{Region} = 0.92$ and $\sigma_{Pack} = 1.18$. These are about the same (well within 95% CI), so we can say that Region and Pack contribute about equally to the variation in wolves' diets. Check that the mean, SD, and quantiles for **Region.SD** and **Pack.SD** match their posterior density distributions (posterior_density_SD.pdf, Figure 5).

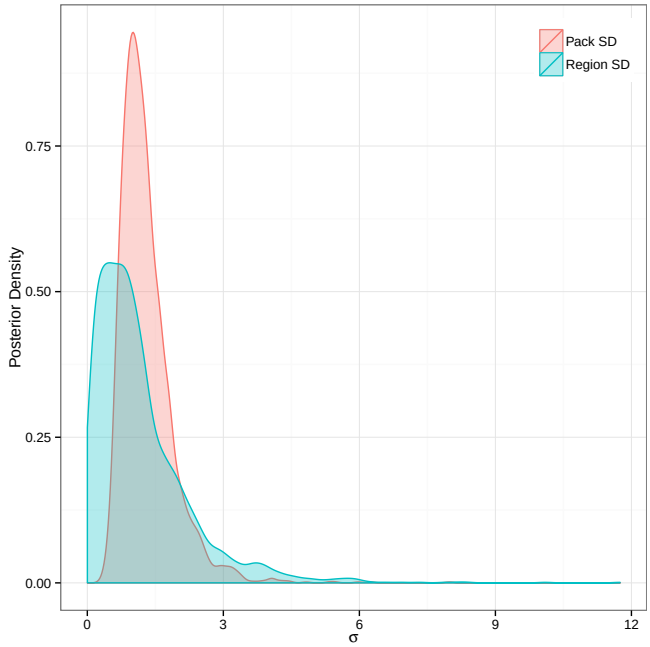


Figure 5: **Posterior plot comparing the variation in diet by each factor**. Including covariates as Random Effects in MixSIAR allows you to examine which explains more of the variation in consumer diet. In the Wolves Example here, σ_{Region} and σ_{Pack} are about the same, so we can say that Region and Pack contribute about equally to the variation in wolves' diets. Median and quantile values are printed in Summary Statistics—see Region.SD and Pack.SD.

You can also look at the mean, SD, and quantiles for **p.Region** (diet proportion by Region), **p.Pack** (diet proportion by Pack), and **p.global** (diet proportion of the overall wolf population). As an example, to find the median diet for Region 1 wolves (71.4% Deer, 8.5% Marine Mammals, and 16.5% Salmon), look in Summary Statistics for the 50% quantiles of p.Region 1.Deer, p.Region 1.Marine Mammals, and p.Region 1.Salmon (red):

	Mean	SD	2.5%	5%	25%	50%	75%	95%	97.5%
Region.SD	1.176	1.054	0.053	0.097	0.463	0.920	1.566	3.198	3.947
Pack.SD	1.304	0.583	0.595	0.655	0.918	1.182	1.556	2.364	2.740
p.global.Deer	0.467	0.185	0.128	0.157	0.316	0.485	0.620	0.738	0.763
p.global.Marine Mammals	0.211	0.140	0.024	0.036	0.111	0.175	0.288	0.499	0.563
p.global.Salmon	0.322	0.182	0.079	0.095	0.179	0.279	0.431	0.682	0.753
p.Region 1.Deer	0.691	0.164	0.297	0.371	0.592	0.714	0.817	0.914	0.936
p.Region 2.Deer	0.546	0.158	0.211	0.265	0.444	0.553	0.658	0.795	0.825
p.Region 3.Deer	0.411	0.225	0.016	0.042	0.228	0.416	0.586	0.770	0.827
p.Region 1.Marine Mammals	0.111	0.101	0.001	0.005	0.037	0.085	0.153	0.306	0.368
p.Region 2.Marine Mammals	0.164	0.106	0.023	0.033	0.090	0.141	0.216	0.374	0.429
p.Region 3.Marine Mammals	0.244	0.188	0.007	0.023	0.106	0.195	0.339	0.637	0.718
p.Region 1.Salmon	0.198	0.154	0.009	0.020	0.082	0.165	0.271	0.510	0.626
p.Region 2.Salmon	0.290	0.153	0.062	0.083	0.179	0.266	0.377	0.589	0.666
p.Region 3.Salmon	0.345	0.242	0.030	0.055	0.163	0.284	0.473	0.870	0.945

95% credible intervals[6] for Deer, Marine Mammals, and Salmon contribution to Region 1 wolves' diet would be found in Summary Statistics 2.5% and 97.5% of p.Region 1.Deer, p.Region 1.Marine Mammals, and p.Region 1.Salmon (blue). Check that these values agree with the posterior plot of Region 1 wolves' diet (posterior_density_diet_p_Region 1.pdf, Figure 6).

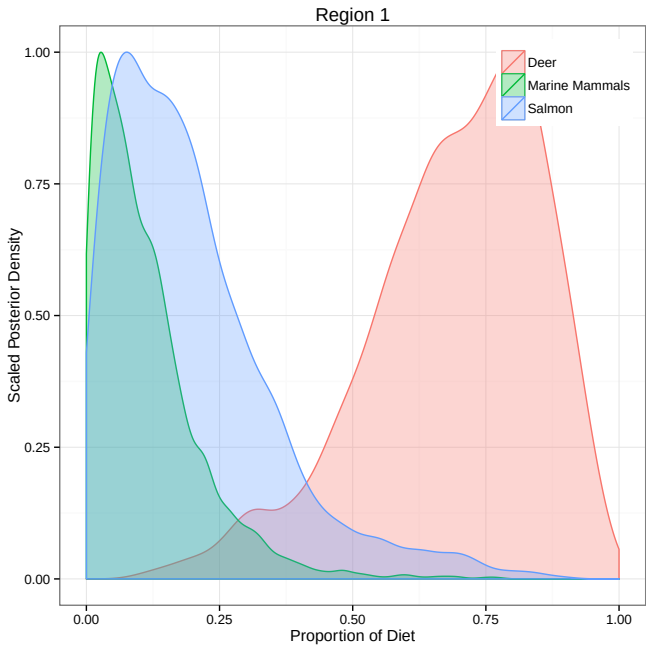


Figure 6: **Posterior plot of Region 1 wolves' diet**. MixSIAR plots the diet of each level of each fixed/random effect. The diet proportions of Region 1 are in Summary Statistics as p.Region 1.Deer, p.Region 1.Marine Mammals, and p.Region 1.Salmon (medians = 71.4% Deer, 8.5% Marine Mammals, and 16.5% Salmon).

[6]These are Bayesian credible intervals, NOT frequentist confidence intervals.

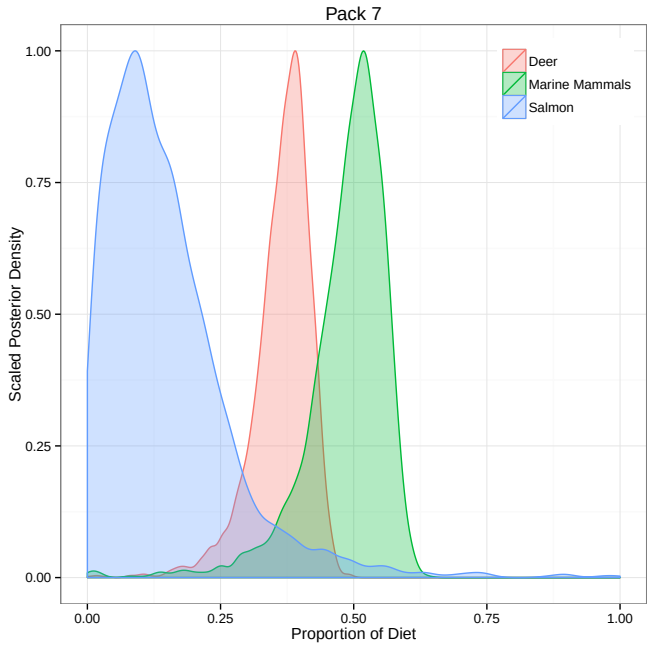


Figure 7: **Posterior plot of Pack 7 wolves' diet**. MixSIAR plots the diet of each level of each categorical covariate. The diet proportions of Pack 7 are in Summary Statistics as p.Pack 7.Deer, p.Pack 7.Marine Mammals, and p.Pack 7.Salmon.

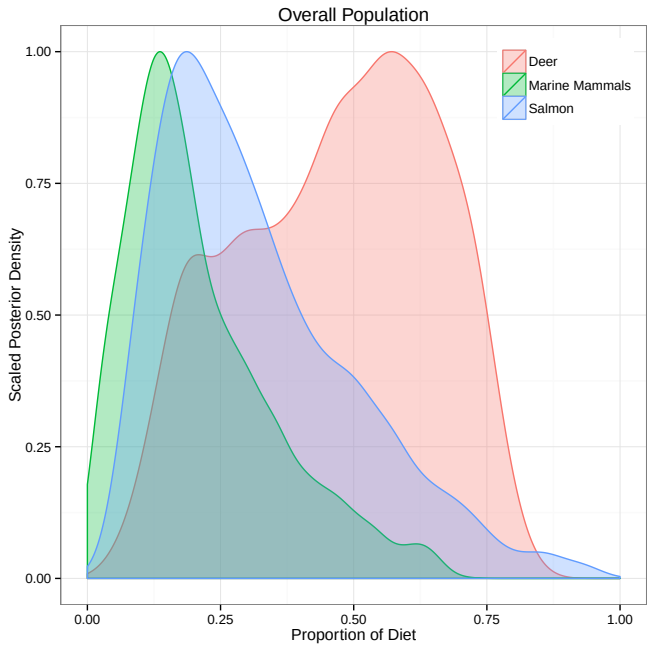


Figure 8: **Posterior plot of overall wolves' diet**. MixSIAR also plots the diet of the overall consumer population, p.global—this is what we'd expect for a new wolf if we didn't know which region or pack it belonged to.

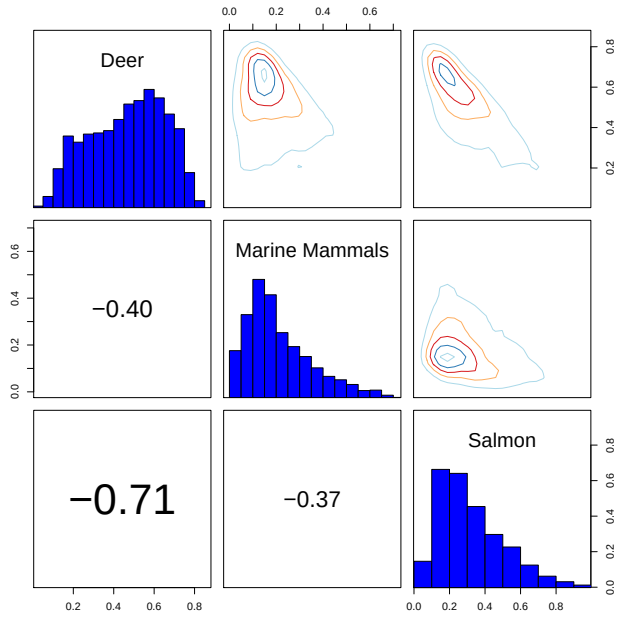


Figure 9: **Pairs plot of the posterior diet proportions of the overall wolf population**. The upper-diagonal shows contour plots, the diagonal shows histograms, and the lower-diagonal shows the correlations between the different sources.

3.1.13 Saving MixSIAR output

By default MixSIAR saves .pdf files of all plots and .txt files of the diagnostics and summary statistics. Plot files can also be saved as .png files if you prefer. If you don't want the diagnostics or summary statistics files, uncheck the relevant boxes. If you don't want to be bothered with the plots, you can check "Suppress plot output".

To save everything in your R workspace, run:

```
save.image(file="wolves_normal.RData")
```

Then if you close and reopen R in the future, you can load the completed Wolves Example with:

```
load(file="wolves_normal.RData")
```

You can also directly access the JAGS model objects if you want to make your own plots, do a post-hoc aggregation of sources, or run other diagnostics. MixSIAR GUI creates a `mixsiar` environment, with `jags.1` the `rjags` object holding the MCMC chains, accessed using `mixsiar$jags.1`. Objects holding MCMC chains you may be interested in are:

- `p.global`: Overall consumer diet proportions

- `p.fac1`: Region consumer diet proportions
- `p.fac2`: Pack consumer diet proportions
- `fac1.sig`: Variation in diet among Regions
- `fac2.sig`: Variation in diet among Packs

Note that while the variables in Summary Statistics are renamed to be more easily interpreted, these objects are necessarily named more generally. For instance, say we are interested in the diet proportions of Region 1 wolves. In the Summary Statistics, we find **p.Region 1.Deer**, **p.Region 1.Marine Mammals**, and **p.Region 1.Salmon**. However, in the JAGS model, `p.fac1` is the diet proportion by Region, indexed as [MCMC chain, Region, Source]:

```
dim(p.fac1)

## [1] 3000    3    3
```

`p.fac1[,1,1]` is the chain for the proportion of Region 1 wolves' diet that was Deer:

```
median(p.fac1[,1,1])

## [1] 0.7142127
```

`p.fac1[,1,2]` is the chain for the proportion of Region 1 wolves' diet that was Marine Mammals:

```
median(p.fac1[,1,2])

## [1] 0.08482204
```

Likewise for Pack, coded as **`p.fac2`**, indexed as [Pack, Source]). *Sources are sorted alphabetically.* If you are unsure of the indexing, compare the Summary Statistics to the posterior density plots and you should be able to figure out which is which.

Notice that while you have access to these objects, they are not actually in your workspace. If you save your workspace, then close and re-open R, you will need to enter `attach.jags(mixsiar$ja` to access `p.fac1` again. Alternatively, you can add `p.fac1` to your workspace by entering `save(p.fac1,file="p.saved")` followed by `load("p.saved")`.

3.2 Geese Example

For this walk-through I assume you've already done the Wolves Example, so if something is unclear please refer to the relevant section there. The "Geese Example" uses data from Inger et al. [10] of 251 wintering geese feeding on terrestrial grasses, Zostera spp., Enteromorpha spp., and Ulva lactuca. This is the same data included as a demo in SIAR and in Parnell et al. [18]:

- 2 biotracers (δ^{13}C, δ^{15}N)
- 1 *fixed effect* (Group)
- Source data as means and SDs
- *Concentration dependence*
- Residual only error

3.2.1 Load mix data

1. Click "Load mixture data", then "Load mixture data file" in the new window.
2. Choose "geese_consumer.csv". This file has the C and N isotope values, and categorical covariate (Group).
3. Select which columns are isotopes (d13C and d15N) and which is a fixed effect (Group). *We choose to treat Group as a fixed effect instead of a random effect here because we are interested in the diet of each group separately and NOT in the overall diet.*
4. Click "I'm finished".

3.2.2 Load source data

1. Click "Load source data"
2. Our geese source data are not by Group, but we DO have concentration dependence data (see "geese_sources.csv" file). Next to "Do you have Concentration Dependence data?", click "Yes".
3. We only have source means, SD, and sample size—not the original "raw" data—click "Load source means and SDs".
4. Choose "geese_sources.csv" when prompted for the file, and click "Ok".
5. Click "I'm finished".

3.2.3 Load discrimination data

1. Click "Load discrimination data".
2. Choose "geese_discrimination.csv" when prompted for the file, and click "Ok".

3.2.4 Make isospace plot

Click "Make isospace plot". Your plot should match that of Figure 10.

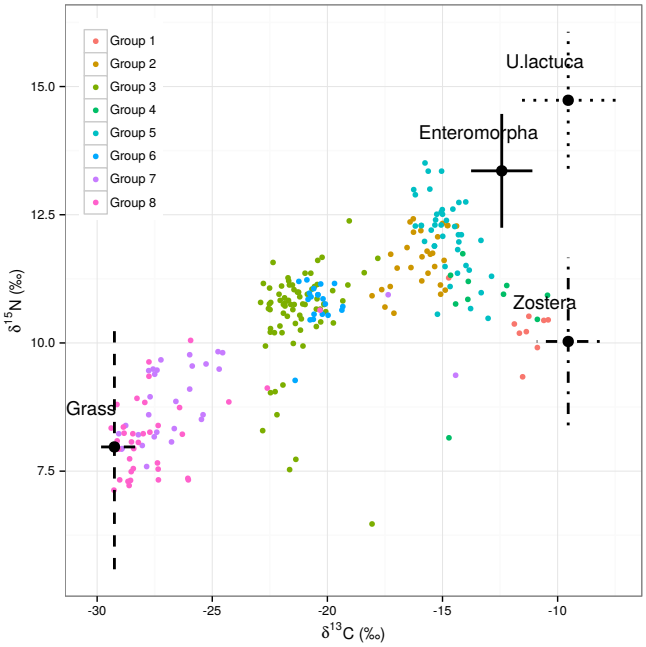


Figure 10: **Stable isotope input for the Geese Example.** Consumer data are by Group (color) and source data are labeled. Error bars indicate combined source and discrimination uncertainty $\pm$ 1 SD.

3.2.5 Specify MCMC parameters

First run using the "test" settings to make sure the model is setup correctly. For more description of these parameters, see Table 2.

3.2.6 Error term options

In the Geese Example we want "Residual error only".

3.2.7 Run MixSIAR (test)

Click the "RUN MODEL" button at the bottom. The Geese Example is larger than the others, so it will take significantly longer to run (~50 seconds for "test").

3.2.8 Using MCMC diagnostics

When the model is finished, click the "Process output" button to generate diagnostics, summary statistics, and posterior plots. Has our model converged? See Section 3.1.9

if you need a refresher on how to do this. Based on the Gelman-Rubin diagnostics, our "test" run is not even close to convergence:

```
################################################
# Gelman-Rubin Diagnostic
################################################

Generally the Gelman diagnostic should be < 1.05

Out of 37 variables: 37 > 1.01
                     31 > 1.05
                     25 > 1.1

The worst variables are:

            Point est. Upper C.I.
p.fac1[3,1]   3.505785  14.090565
p.fac1[3,3]   3.038753   7.326779
p.fac1[5,1]   2.694420   5.754130
p.fac1[5,3]   2.666437   5.508339
```

3.2.9 Run MixSIAR (for real)

1. Under "MCMC run length", click "short".
2. Click "RUN MODEL" button.

This will take roughly 40 minutes to complete, so be patient!

3.2.10 Check diagnostics (again)

After the model is finished, click the "Process output" button again. Based on the Gelman-Rubin diagnostic, our "short" run is much closer to convergence:

```
################################################
# Gelman-Rubin Diagnostic
################################################

Generally the Gelman diagnostic should be < 1.05

Out of 37 variables: 28 > 1.01
                     19 > 1.05
                     12 > 1.1

The worst variables are:

            Point est. Upper C.I.
p.global[1]   1.170947   1.546743
p.fac1[1,1]   1.170947   1.546743
p.fac1[1,3]   1.129189   1.405320
p.global[3]   1.129189   1.405320
p.fac1[2,3]   1.127919   1.406975
```

We will call these good enough for the purposes of this tutorial—would NOT accept these as final results.

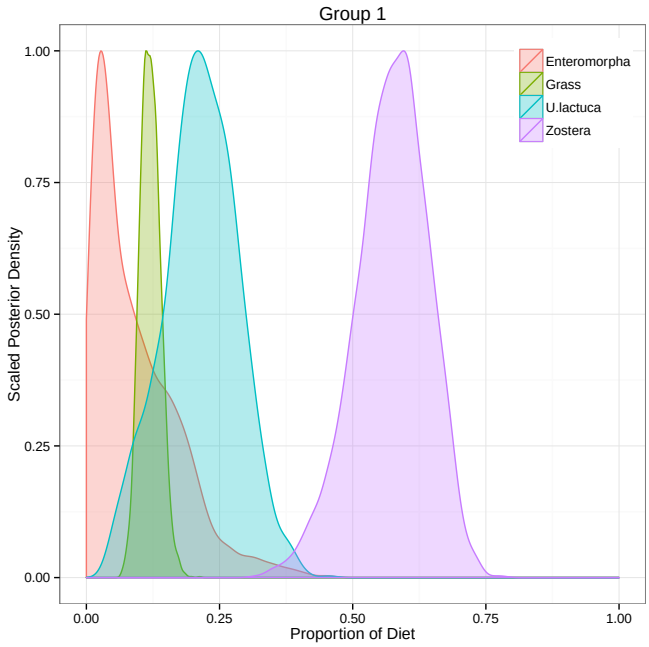


Figure 11: **Posterior plot of Group 1 geese diet.** For medians and credible interval limits, look in the Summary Statistics at p.Group 1.Grass, p.Group 1.Enteromorpha, p.Group 1.U.lactuca, and 1.Group 6.Zostera.

3.2.11 Interpreting output

You should see posterior plots of diet by Group (Figure 11, "posterior_density_diet_p_Group 1.pdf", etc.). *Note that there is no global/overall estimated diet*—this is because we fit Group as a fixed effect instead of a random effect.

As in the other working examples, MixSIAR also prints and saves summary statistics. As an example, to find the median diet proportions for Group 1 geese (7.2%, 11.9%, 21.6%, 57.7%), look in Summary Statistics for the 50% values of **p.Group 1.Enteromorpha**, **p.Group 1.Grass**, **p.Group 1.U.lactuca**, and **p.Group 1.Zostera**. The 95% credible interval for Grass contribution to Group 1 geese diet would be found in Summary Statistics, 2.5% and 97.5% of **p.Group 1.Grass**. See that these values agree with the posterior plot of Group 1 diet in Figure 11.

3.3 Lake Example

The "Lake Example" data is simulated based on Francis et al. [5] and we include it as a example of how MixSIAR can include a *continuous effect*. Here we examine the diet of zooplankton in 21 lakes using:

- 2 biotracers (δ^{13}C, δ^{15}N)
- 1 continuous effect (Secchi Depth : Mixed Layer Depth)

- Raw source data

Fitting a model with a continuous effect is more complex than the categorical fixed/random effects and can be a bit finicky.

3.3.1 Load mix data

1. Click "Load mixture data", then "Load mixture data file" in the new window.
2. Choose "lake_consumer.csv". This file has the C and N isotope values, and continuous covariate (Secchi.Mixed).
3. Select which columns are isotopes (d13C and d15N) and which is the continuous effect (Secchi.Mixed).
4. Click "I'm finished".

3.3.2 Load source data

1. Click "Load source data"
2. We do not have concentration dependence data (see "lake_sources.csv" file), so click "No".
3. We have the original "raw" source data—click "Load raw source data".
4. Choose "lake_sources.csv" when prompted for the file, and click "Ok".
5. Click "I'm finished".

3.3.3 Load discrimination data

1. Click "Load discrimination data".
2. Choose "lake_discrimination.csv" when prompted for the file, and click "Ok".

3.3.4 Make isospace plot

Click "Make isospace plot". Your plot should match that of Figure 12.

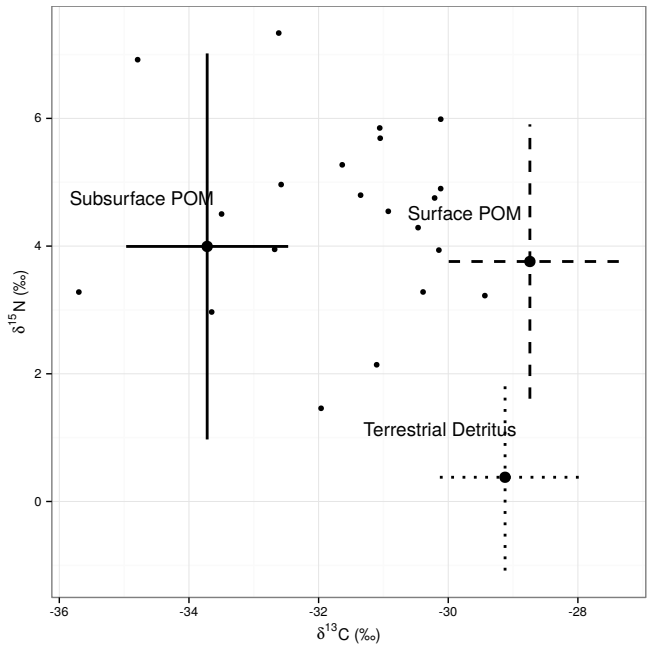


Figure 12: **Stable isotope input for the Lake Example.** Consumer data are highly variable, but much of this variance is explained by the Secchi:Mixed continuous covariate. Error bars indicate combined source and discrimination uncertainty $\pm$ 1 SD.

3.3.5 Specify MCMC parameters

First run using the "test" settings to make sure the model is setup correctly. When you're satisfied eveything is ok, then choose the "normal" MCMC run length. For more description of these parameters, see Table 2.

3.3.6 Error term options

In the Lake Example we want "Residual error only".

3.3.7 Run MixSIAR

Click the "RUN MODEL" button at the bottom. The Lake Example took my laptop ~13 minutes for "normal".

3.3.8 Check diagnostics

After the model is finished, click the "Process output" button. Based on the Gelman-Rubin and Geweke diagnostics, our "normal" run has converged well enough to accept the results:

```
################################################
# Gelman-Rubin Diagnostic
################################################

Generally the Gelman diagnostic should be < 1.05

Out of 71 variables: 0 > 1.01
                     0 > 1.05
                     0 > 1.1

The worst variables are:

             Point est. Upper C.I.
p.ind[18,2]    1.009192   1.016706
p.ind[8,3]     1.008704   1.011874
ilr.cont1[1]   1.007628   1.023459


################################################################################
# Geweke Diagnostic
################################################################################

The Geweke diagnostic is a standard z-score, so we'd expect 5% to be outside +/-1.96
Number of variables outside +/-1.96 in each chain (out of 71):

       Chain 1 Chain 2 Chain 3
Geweke       9       3       0
```

3.3.9 Interpreting output

MixSIAR fits a continuous covariate as a linear regression in ILR/transform-space. Two terms are fit for the proportion of each source: an intercept and a slope. The plot uses the posterior median estimates of the intercept and slope, and the lines are curved because of the ILR-transform back into p-space (Figure 13). For details, or if you would like to make modifications, see `plot_continous_var.r`.

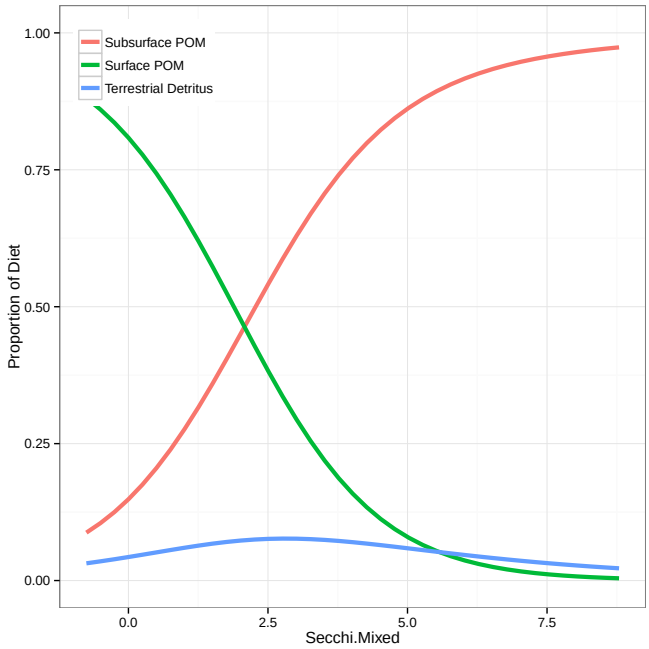


Figure 13: **Plot of lake zooplankton diet as a function of Secchi depth:Mixed layer depth ratio.** The plot uses the posterior median estimates of the intercept and slope, and the lines are curved because of the ILR-transform back into p-space.

The other posterior plots MixSIAR produces for a continuous effect show the estimated diet for the minimum, median, and maximum individuals (Figure 14).

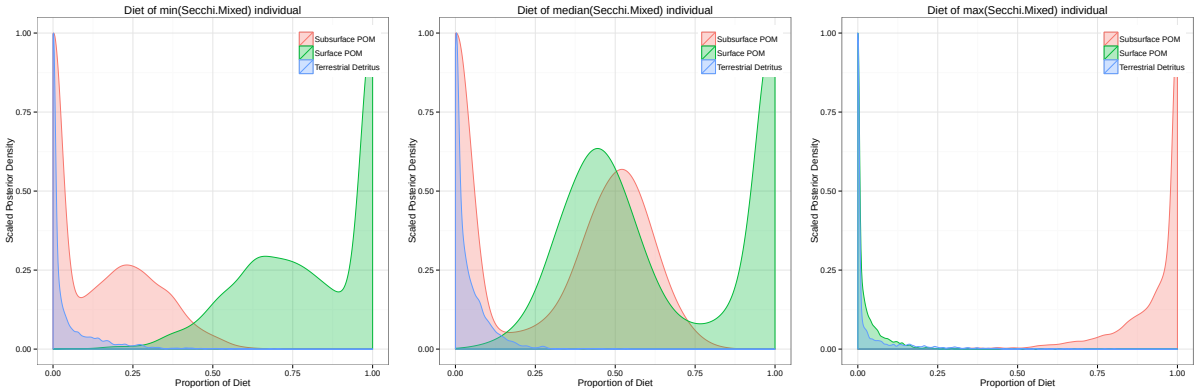


Figure 14: **Lake zooplankton diet for the min, median, and max individuals.**

3.4 Palmyra Example

The "Palmyra Example" data is actual data from McCauley et al. [13] and analyzes the diet of 3 large reef predators (Taxa = *fixed effect*) around the Palmyra Atoll with:

- 2 biotracers (δ^{13}C, δ^{15}N)

- 1 fixed effect (Taxa)
- Raw source data

3.4.1 Load mix data

1. Click “Load mixture data”, then “Load mixture data file” in the new window.
2. Choose “palmyra_consumer.csv”. This file has the C and N isotope values, and fixed effect (Taxa).
3. Select which columns are isotopes (d13C and d15N) and which is the fixed effect (Taxa).
4. Click “I’m finished”.

3.4.2 Load source data

1. Click “Load source data”
2. We do not have concentration dependence data (see “palmyra_sources.csv” file), so click “No”.
3. We do have the original “raw” source data—click “Load raw source data”.
4. Choose “palmyra_sources.csv” when prompted for the file, and click “Ok”.
5. Click “I’m finished”.

3.4.3 Load discrimination data

1. Click “Load discrimination data”.
2. Choose “palmyra_discrimination.csv” when prompted for the file, and click “Ok”.

3.4.4 Make isospace plot

Click “Make isospace plot”. Your plot should match that of Figure 15.

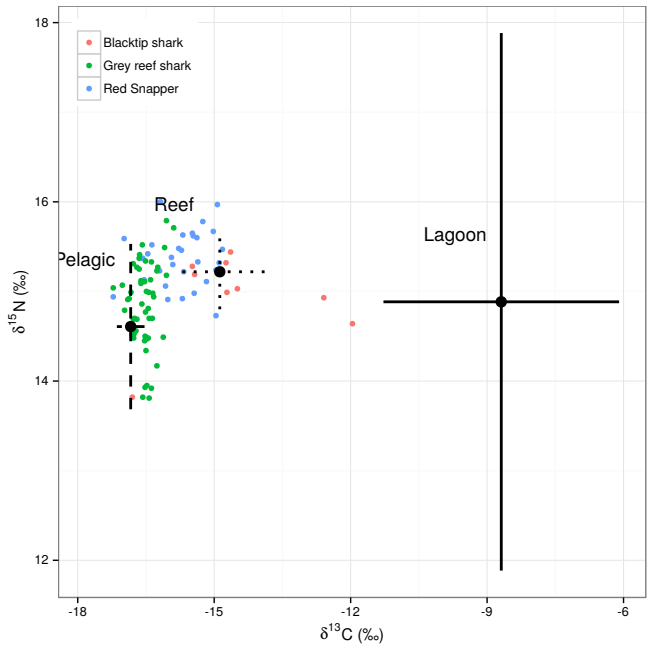


Figure 15: **Stable isotope input for the Palmyra Example.** Error bars indicate combined source and discrimination uncertainty $\pm$ 1 SD. The lagoon source error bars are wider than the other sources because it is assumed to be one trophic step below them (and thus has additional discrimination uncertainty). See McCauley et al. [13], Appendix A, for explanation.

3.4.5 Specify MCMC parameters

First run using the "test" settings to make sure the model is setup correctly. When you're satisfied eveything is ok, then choose the "short" MCMC run length. For more description of these parameters, see Table 2.

3.4.6 Error term options

In the Palmyra Example we want "Residual * Process".

3.4.7 Run MixSIAR

Click the "RUN MODEL" button at the bottom. The Palmyra Example took my laptop ~75 minutes for "short".

3.4.8 Check diagnostics

After the model is finished, click the "Process output" button. Based on the Gelman-Rubin and Geweke diagnostics, our "short" run has converged well enough to accept the results:

```
##################################################
# Gelman-Rubin Diagnostic
##################################################

Generally the Gelman diagnostic should be < 1.05

Out of 15 variables: 2 > 1.01
                     0 > 1.05
                     0 > 1.1

The worst variables are:

              Point est. Upper C.I.
p.fac1[1,3]     1.010946   1.032718
p.global[3]     1.010946   1.032718
p.fac1[1,2]     1.009324   1.030703
p.global[2]     1.009324   1.030703
p.fac1[3,3]     1.007461   1.008460
```

```
################################################################################
# Geweke Diagnostic
################################################################################

The Geweke diagnostic is a standard z-score, so we'd expect 5% to be outside +/-1.96
Number of variables outside +/-1.96 in each chain (out of 15):

       Chain 1 Chain 2 Chain 3
Geweke       0       0       0
```

3.4.9 Interpreting output

Your posterior density plots for blacktip and grey reef sharks should match Figure 16.

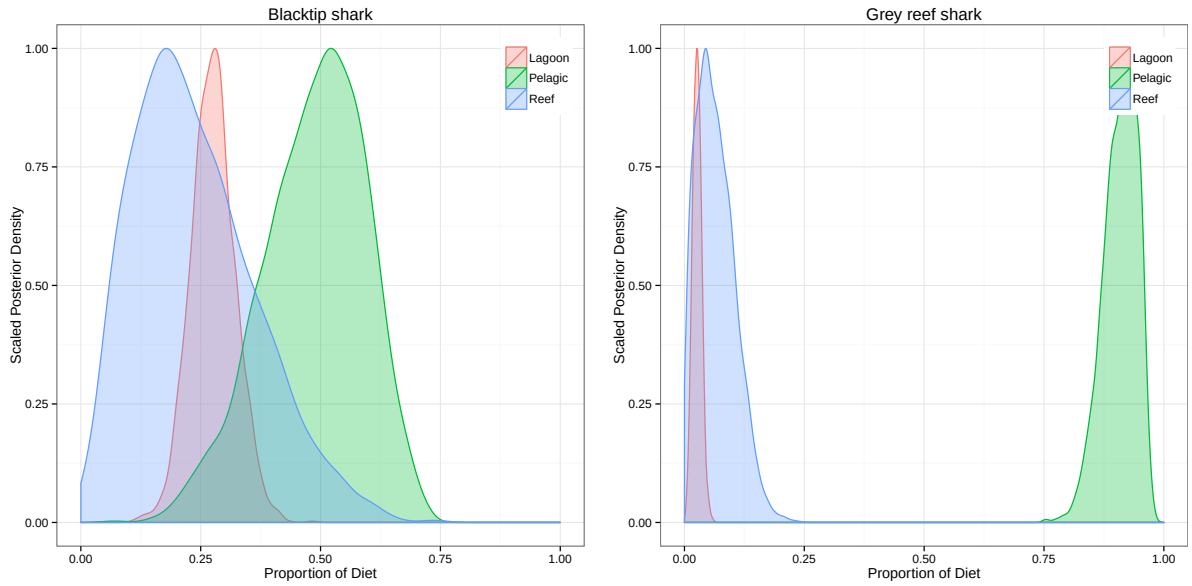


Figure 16: **Posterior diet estimates for blacktip and grey reef sharks in the Palmyra Example.** Median diet proportions are (27.6% Lagoon, 49.8% Pelagic, 21.9% Reef) for blacktip sharks , and (2.6% Lagoon, 91.3% Pelagic, 6% Reef) for grey reef sharks.

Check that the medians and 95% CIs in the summary statistics match the posterior plots in Figure 16:

	Mean	SD	2.5%	5%	25%	50%	75%	95%	97.5%
p.Blacktip shark.Lagoon	0.276	0.047	0.188	0.200	0.245	0.276	0.305	0.354	0.370
p.Grey reef shark.Lagoon	0.026	0.009	0.009	0.012	0.020	0.026	0.033	0.042	0.045
p.Red Snapper.Lagoon	0.073	0.033	0.017	0.023	0.051	0.071	0.094	0.130	0.147
p.Blacktip shark.Pelagic	0.487	0.106	0.259	0.298	0.417	0.498	0.565	0.643	0.667
p.Grey reef shark.Pelagic	0.909	0.036	0.830	0.844	0.886	0.913	0.936	0.958	0.963
p.Red Snapper.Pelagic	0.461	0.113	0.259	0.299	0.396	0.452	0.511	0.669	0.788
p.Blacktip shark.Reef	0.237	0.127	0.045	0.061	0.141	0.219	0.317	0.472	0.525
p.Grey reef shark.Reef	0.065	0.040	0.006	0.009	0.034	0.060	0.091	0.137	0.153
p.Red Snapper.Reef	0.466	0.133	0.058	0.219	0.412	0.478	0.541	0.651	0.698

3.5 Killer Whale Example

The Killer Whale Example demonstrates the difference *informative priors* make, and illustrates how to construct them.

- 2 biotracers (δ^{13}C, δ^{15}N)
- Source data as means + SDs
- *Informative vs. uninformative priors*

3.5.1 Load mix data

1. Click "Load mixture data", then "Load mixture data file" in the new window.
2. Choose "killerwhale_consumer.csv". This file has the C and N isotope values.
3. Select which columns are isotopes (d13C and d15N).
4. Click "I'm finished".

3.5.2 Load source data

1. Click "Load source data"
2. We do not have concentration dependence data (see "killerwhale_sources.csv" file), so click "No".
3. We only have source summary statistics (mean, SD, and sample size), not the original "raw" data—Click "Load source means and SDs".
4. Choose "killerwhale_sources.csv" when prompted for the file, and click "Ok".
5. Click "I'm finished".

3.5.3 Load discrimination data

1. Click "Load discrimination data".
2. Choose "killerwhale_discrimination.csv" when prompted for the file, and click "Ok".

3.5.4 Make isospace plot

Click "Make isospace plot". Your plot should match that of Figure 17. Notice that the high degree of correlation places the sources on a line, which makes it difficult for the model to determine if killer whales are eating Chinook vs. Coho, or Sockeye vs. Steelhead vs. Chum. We can give the model additional information to resolve this problem by using an *informative prior*.

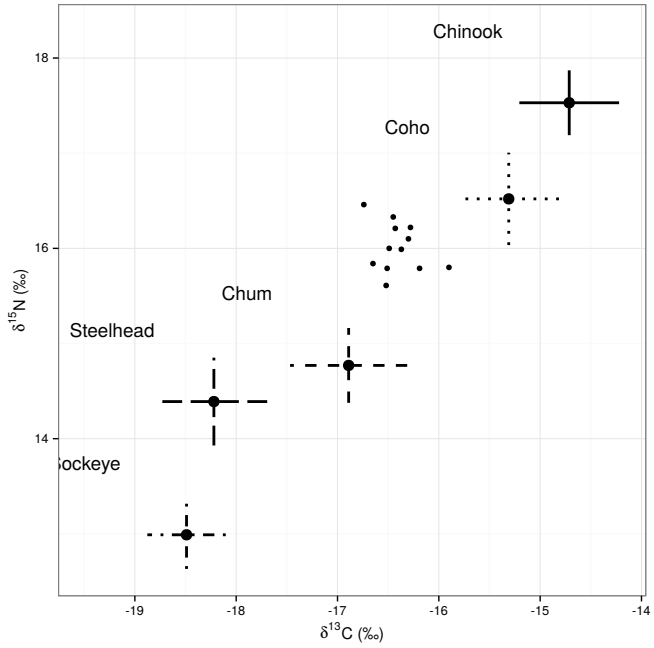


Figure 17: **Stable isotope input for the Killer Whale Example.** Error bars indicate combined source and discrimination uncertainty $\pm$ 1 SD. Notice that the high degree of correlation places the sources on a line, which makes it difficult for the model to determine if killer whales are eating Chinook vs. Coho, Sockeye vs. Steelhead vs. Chum.

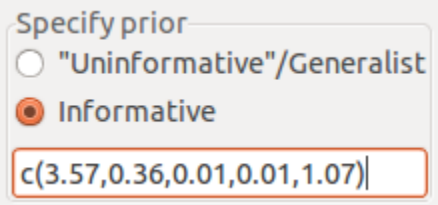
3.5.5 Constructing the informative prior

From Semmens et al. [23]:

One of the benefits to conducting mixture models in a Bayesian framework is that information from other data sources can be included via *informative prior distributions* Moore and Semmens [15]. Suppose we have collected 14 fecal samples from the killer whale

population we are studying, and we find 10 chinook, 1 chum, 0 coho, 0 sockeye, and 3 steelhead. The sum of the Dirichlet hyperparameters roughly correspond to prior sample size, so one approach would be to construct a prior with $\alpha = (10, 1, 0, 0, 3)$. A downside of this prior is that a sample size of 14 represents a very informative prior, with much of the parameter space given very little weight. Keeping the relative contributions the same, the hyperparameters can be rescaled to have the same mean, but different variance. An alternative is to scale the prior to have a weight of 5, equal to the same weight as the "uninformative" prior $\alpha = (1, 1, 1, 1, 1)$. This prior could be represented as $\alpha = \left(\frac{10*5}{14}, \frac{1*5}{14}, \frac{0*5}{14}, \frac{0*5}{14}, \frac{3*5}{14}\right) = (3.57, 0.36, 0.01, 0.01, 1.07)$.

In the MixSIAR GUI, under "Specify prior", click "Informative." In the box, enter your α vector in R format, e.g. `c(...,  ...,  ...)`:



You cannot have α parameters = 0, so while we had no coho and sockeye in our fecal samples, we set their α parameters = 0.01. *Remember that the sources are sorted alphabetically*. Click the "Plot prior" button to compare your informative prior with the uninformative prior $\alpha = (1, 1, 1, 1, 1)$, as in Figure 18.

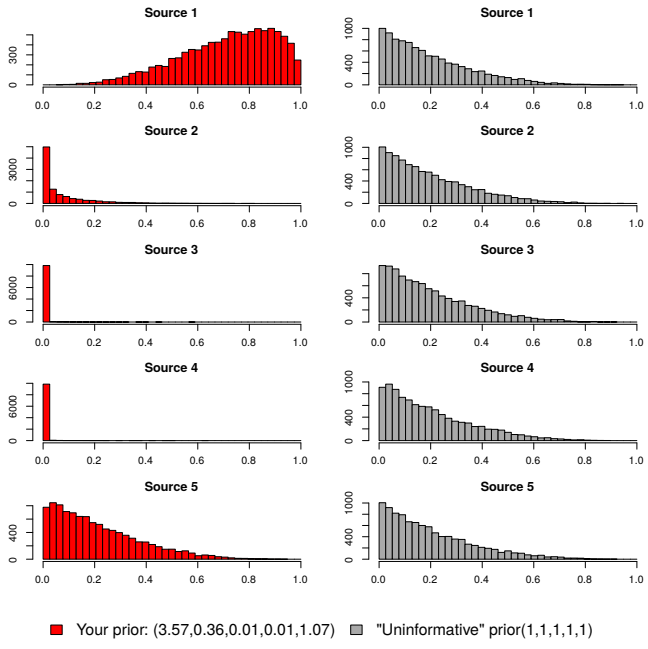


Figure 18: **Informative prior for Killer Whale Example.** The fecal data was $(10, 1, 0, 0, 3)$, and then the α vector is scaled to sum to the number of sources, keeping the same mean.

3.5.6 Specify MCMC parameters

First run using the "test" settings to make sure the model is setup correctly. When you're satisfied eveything is ok, then choose the "very long" MCMC run length. For more description of these parameters, see Table 2.

3.5.7 Error term options

In the Killer Whale Example, choose "Residual only".

3.5.8 Run MixSIAR

Click the "RUN MODEL" button at the bottom. The Killer Whale Example took my laptop ~5 minutes for "very long".

3.5.9 Check diagnostics

After the model is finished, click the "Process output" button. Check the Gelman-Rubin and Geweke diagnostics as usual.

3.5.10 Interpreting output

Your posterior density plot with the informative prior should match the right panel of Figure 19. If you run the model again using the uninformative prior, you should get the left panel of Figure 19.

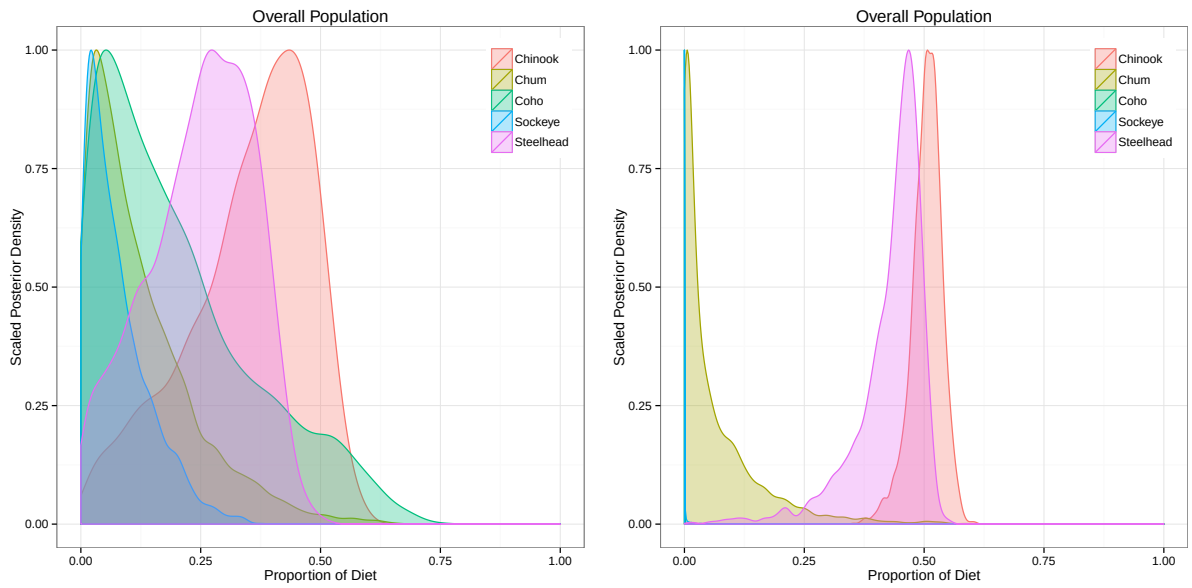


Figure 19: **Posterior diet estimates for the Killer Whale Example.** On the left are the results using the uninformative prior $\alpha = (1,1,1,1,1)$, and on the right are the results using the informative prior $\alpha = (3.57, 0.36, 0.01, 0.01, 1.07)$.

Check that the medians and 95% CIs in the summary statistics match the right plot (informative prior) in Figure 19:

```
                   Mean    SD  2.5%    5%   25%   50%   75%   95% 97.5%
p.global.Chinook   0.505 0.032 0.429 0.447 0.487 0.508 0.526 0.553 0.561
p.global.Chum      0.063 0.088 0.000 0.000 0.004 0.025 0.090 0.244 0.324
p.global.Coho      0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
p.global.Sockeye   0.003 0.017 0.000 0.000 0.000 0.000 0.000 0.008 0.032
p.global.Steelhead 0.430 0.072 0.221 0.289 0.405 0.450 0.476 0.504 0.512
```

Our prior was quite informative on Coho and Sockeye, effectively setting them to 0. Notice, however, that there was still significant information in our data—evidenced by the posteriors for Chinook and Steelhead being very different from their priors (Chinook prior mean = 10/14 = 71.4%, Chinook posterior mean = 50.5%, Steelhead prior mean = 3/14 = 21.4%, Steelhead posterior mean = 43.0%).

3.6 Isopod Example

The Isopod Example is from Galloway et al. [6] and demonstrates MixSIAR applied to an *8-dimensional fatty acid* dataset. Here the mixture data are isopod polyunsaturated fatty acid (PUFA) profiles, with 5 replicates at each of 6 sites in Puget Sound, WA:

- 8 biotracers (carbon 16.4ω3, 18.2ω6, 18.3ω3, 18.4ω3, 20.4ω6, 20.5ω3, 22.5ω3, 22.6ω3)
- 1 random effect (Site)

- Source data as means and SDs

Fatty acid data greatly increase the number of biotracers beyond the typical 2 stable isotopes, δ^{13}C and δ^{15}N, which gives the mixing model power to resolve more sources. We caution, however, that it is not only a matter of # biotracers > # sources + 1. As the number of sources increases, the "uninformative" prior $\alpha = 1$ has greater influence, illustrated in the Cladocera Example, Figure .

3.6.1 Load mix data

1. Click "Load mixture data", then "Load mixture data file" in the new window.
2. Choose "isopod_consumer.csv". This file has the fatty acid and random effect (Site) values.
3. Select which columns are "isotopes" (c16.4w3, c18.2w6, c18.3w3, c18.4w3, c20.4w6 , c20.5w3, c22.5w3, c22.6w3).
4. Select Site as a random effect
5. Click "I'm finished".

Here we treat Site as a *random effect*. This makes sense if we are interested in the overall population and think of Site as a nuisance factor. Fitting Site as a *fixed effect* would make more sense if we were interested specifically in the diet at each Site, as opposed to the overall population diet and variability between Sites. This differs from the analysis in Galloway et al. [6].

3.6.2 Load source data

1. Click "Load source data"
2. We do not have concentration dependence data (see "isopod_sources.csv" file), so click "No".
3. We only have source summary statistics (mean, SD, and sample size), not the original "raw" data—Click "Load source means and SDs".
4. Choose "isopod_sources.csv" when prompted for the file, and click "Ok".
5. Click "I'm finished".

3.6.3 Load discrimination data

1. Click "Load discrimination data".
2. Choose "isopod_discrimination.csv" when prompted for the file, and click "Ok".

Note that Galloway et al. [6] conducted feeding trials to create a "resource library". In the mixing model, the sources are actually consumers fed exclusively each of the sources. This allowed them to set the discrimination = 0 (see "isopod_discrimination.csv").

3.6.4 Make isospace plot

Click "Make isospace plot". When there are more than 2 biotracers, MixSIAR currently plots every pairwise combination. Here, that means $\binom{8}{2} = 28$ plots are produced. In the future, MixSIAR will offer non-metric multidimensional scaling (NMDS) plots for these cases.

3.6.5 Specify MCMC parameters

First run using the "test" settings to make sure the model is setup correctly. When you're satisfied eveything is ok, then choose the "normal" MCMC run length. For more description of these parameters, see Table 2.

3.6.6 Error term options

In the Isopod Example, choose "Residual only".

3.6.7 Run MixSIAR

Click the "RUN MODEL" button at the bottom. The Isopod Example took my laptop ~60 minutes for "normal".

3.6.8 Check diagnostics

After the model is finished, click the "Process output" button. Check the Gelman-Rubin and Geweke diagnostics as usual:

```
##################################################
# Gelman-Rubin Diagnostic
##################################################

Generally the Gelman diagnostic should be < 1.05

Out of 23 variables: 2 > 1.01
                     0 > 1.05
                     0 > 1.1

The worst variables are:

            Point est. Upper C.I.
p.global[1]   1.013676   1.038449
p.global[2]   1.010528   1.023756
p.global[3]   1.009386   1.027817
p.fac1[4,1]   1.003440   1.011701
p.fac1[6,3]   1.002001   1.005790
```

```
The Geweke diagnostic is a standard z-score, so we'd expect 5% to be outside +/-1.96
Number of variables outside +/-1.96 in each chain (out of 23):

       Chain 1 Chain 2 Chain 3
Geweke       2       3       0
```

3.6.9 Interpreting output

Since we fit Site as a random effect, we can make inference on diet at the overall population level (`p.global`, posterior plot in Figure 20):

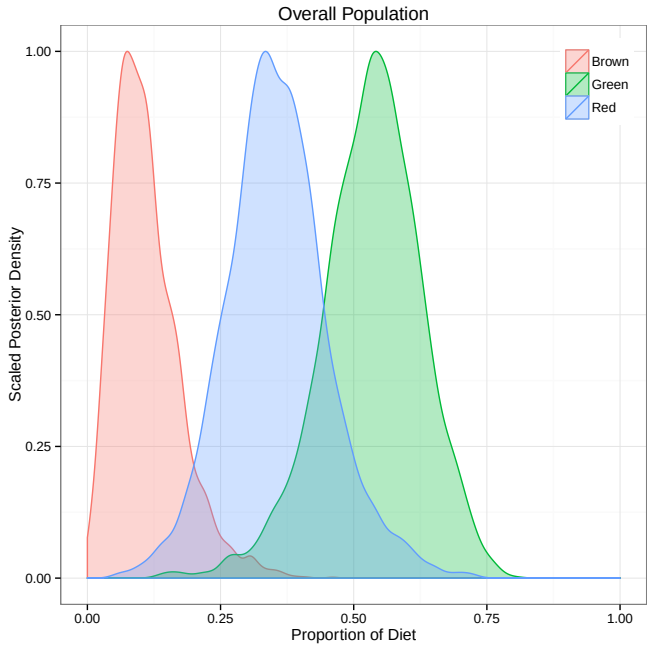


Figure 20: **Posterior diet estimates for the overall population in the Isopod Example.** Median diet proportions are 10% Brown algae, 53.7% Green algae, 35.2% Red algae.

Check that the medians and 95% CIs in the summary statistics for `p.global` match the posterior distributions in Figure 20:

	Mean	SD	2.5%	5%	25%	50%	75%	95%	97.5%
Site.SD	0.679	0.333	0.255	0.295	0.449	0.605	0.826	1.312	1.538
p.global.Brown	0.111	0.062	0.022	0.030	0.066	0.100	0.145	0.227	0.266
p.global.Green	0.533	0.093	0.334	0.373	0.477	0.537	0.595	0.681	0.703
p.global.Red	0.356	0.095	0.178	0.206	0.296	0.352	0.413	0.524	0.570

We also have posterior plots for each Site (Site shown in Figure 21):

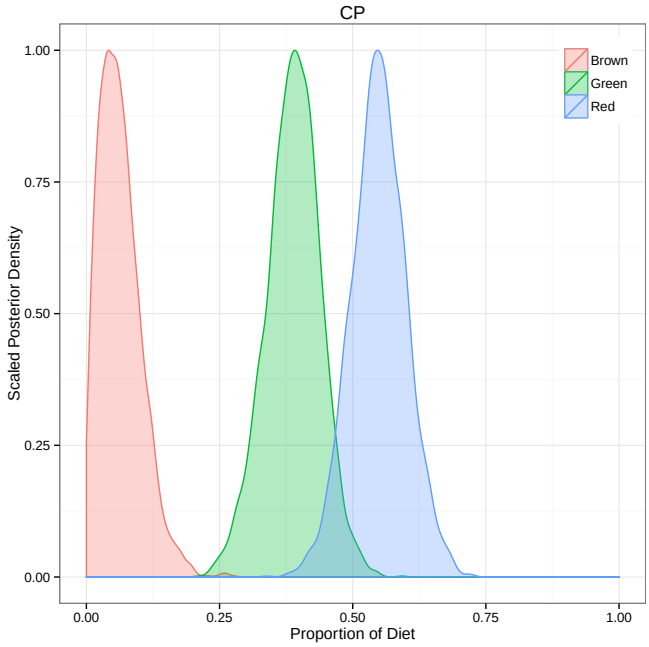


Figure 21: **Posterior estimates of isopod diet at Site CP.** Median diet proportions are 5.8% Brown algae, 39.1% Green algae, 54.9% Red algae.

3.7 Cladocera Example

The Cladocera Example is from Galloway et al. [7] and demonstrates MixSIAR applied to a *22-dimensional fatty acid* dataset. Here the 14 mixture datapoints are Cladocera (water fleas) fatty acid profiles from 6 lakes in Finland over 2 seasons. Besides the high dimensionality, the other difference with this analysis is that we *fit each mixture datapoint individually*, because there is no clear covariate structure (some sites have 2 seasons, some have 1, some sites in the same lake). We do this by creating an "id" column and treating "id" as a fixed effect.

- 22 biotracers (carbon 14.0, 16.0, 16.1ω9, 16.1ω7, 16.2ω4, 16.3ω3, 16.4ω3, 17.0, 18.0, 18.1ω9, 18.1ω7, 18.2ω6, 18.3ω6, 18.3ω3, 18.4ω3, 18.5ω3, 20.0, 22.0, 20.4ω6, 20.5ω3, 22.6ω3, BrFA)

- *Mix datapoints fit independently*
- Source data as means and SDs

Fatty acid data greatly increase the number of biotracers beyond the typical 2 stable isotopes, δ^{13}C and δ^{15}N, which gives the mixing model power to resolve more sources. We caution, however, that it is not only a matter of # biotracers > # sources + 1. As the number of sources increases, the "uninformative" prior $\alpha = 1$ has greater influence, illustrated in Figure .

3.7.1 Load mix data

1. Click "Load mixture data", then "Load mixture data file" in the new window.
2. Choose "cladocera_consumer.csv". This file has the fatty acid and fixed effect (id) values.
3. Select which columns are "isotopes" (c14.0, c16.0, c16.1w9, c16.1w7, c16.2w4, c16.3w3, c16.4w3, c17.0, c18.0, c18.1w9, c18.1w7 , c18.2w6, c18.3w6, c18.3w3, c18.4w3, c18.5w3, c20.0, c22.0, c20.4w6, c20.5w3, c22.6w3, BrFA).
4. Select "id" as a fixed effect
5. Click "I'm finished".

Here we treat "id" as a fixed effect—this will estimate the diet of each mixture datapoint separately (sample size of 1). This makes sense to do when you think there will be clear differences between sites/seasons/etc., but only have 1 or 2 points from each site/season (i.e, *you don't have enough data to estimate a site/season effect*). If you are interested in the site/season effect, you need replicates within each site/season, and then it is best to fit site/season as a fixed or random effect.

3.7.2 Load source data

1. Click "Load source data"
2. We do not have concentration dependence data (see "cladocera_sources.csv" file), so click "No".
3. We only have source summary statistics (mean, SD, and sample size), not the original "raw" data—Click "Load source means and SDs".
4. Choose "cladocera_sources.csv" when prompted for the file, and click "Ok".
5. Click "I'm finished".

3.7.3 Load discrimination data

1. Click "Load discrimination data".
2. Choose "cladocera_discrimination.csv" when prompted for the file, and click "Ok".

Note that Galloway et al. [7] conducted feeding trials to create a "resource library". In the mixing model, the sources are actually consumers fed exclusively each of the sources. This allowed them to set the discrimination = 0 (see "cladocera_discrimination.csv").

3.7.4 Make isospace plot: DON'T

Do NOT click "Make isospace plot"! When there are more than 2 biotracers, MixSIAR currently plots every pairwise combination. Here, that means $\binom{22}{2} = 231$ plots are produced. Yikes! In the future, MixSIAR will offer non-metric multidimensional scaling (NMDS) plots for these cases.

3.7.5 Plot prior

Click "Plot prior" to see our "uninformative" $\alpha = (1, 1, 1, 1, 1, 1, 1)$ prior, as in Figure 22. Fatty acid data greatly increase the number of biotracers beyond the typical 2 stable isotopes, δ^{13}C and δ^{15}N, which gives the mixing model power to resolve more sources. We caution, however, that it is not only a matter of # biotracers > # sources + 1. As the number of sources increases, the "uninformative" prior $\alpha = 1$ has greater influence, illustrated in the difference between the plots with 7 sources (Figure 22) vs. 3 sources (Figure 3).

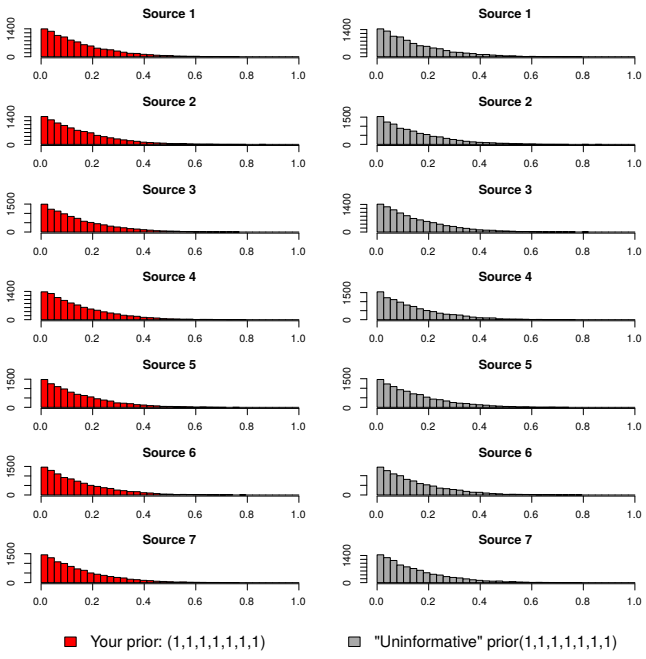


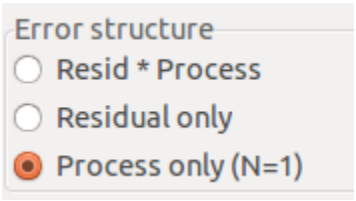
Figure 22: **Uninformative $\alpha = (1,1,1,1,1,1,1)$ prior for 7 sources.** Compare to the uninformative prior with 3 sources (Figure 3).

3.7.6 Specify MCMC parameters

First run using the "test" settings to make sure the model is setup correctly. When you're satisfied eveything is ok, then choose the "normal" MCMC run length. For more description of these parameters, see Table 2.

3.7.7 Error term options

In the Cladocera Example, choose "Process only (N = 1)":



This will fit the "process error" model of MixSIR, which we MUST do when we only have one mix datapoint (or here, one mix datapoint per fixed effect). With only one datapoint, there is no information to estimate an additional mixture variance term, so we have to assume a fixed variance based on the variance of the sources (see Moore and Semmens [15] and Stock and Semmens [25]).

3.7.8 Run MixSIAR

Click the "RUN MODEL" button at the bottom. The Cladocera Example took my laptop ~30 minutes for "normal".

3.7.9 Check diagnostics

After the model is finished, click the "Process output" button. Check the Gelman-Rubin and Geweke diagnostics as usual (these are not great, and should be run longer for a final analysis):

```
##################################################
# Gelman-Rubin Diagnostic
##################################################

Generally the Gelman diagnostic should be < 1.05

Out of 106 variables: 30 > 1.01
                      2 > 1.05
                      0 > 1.1

The worst variables are:

             Point est. Upper C.I.
p.fac1[1,7]    1.059743   1.183363
p.global[7]    1.059743   1.183363
p.fac1[1,4]    1.035258   1.076738
p.global[4]    1.035258   1.076738


##############################################################################
# Geweke Diagnostic
##############################################################################

The Geweke diagnostic is a standard z-score, so we'd expect 5% to be outside +/-1.96
Number of variables outside +/-1.96 in each chain (out of 106):

       Chain 1 Chain 2 Chain 3
Geweke      17      25       9
```

3.7.10 Interpreting output

Since we fit "id" as a fixed effect, there is no inference on diet at the overall population level (no `p.global`). You should see posterior plots for all 14 mixture samples, as in Figure 23.

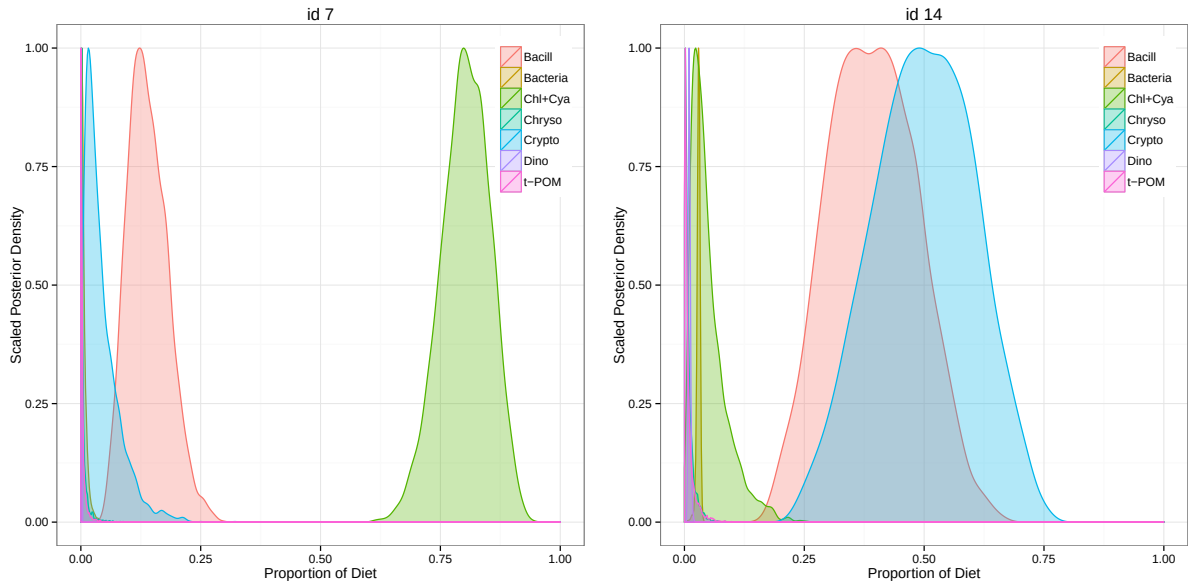


Figure 23: **Posterior estimates of Cladocera diet for samples 7 and 14.** Cladocera diet for sample 7 is dominated by Chl+Cya, while diet for sample 14 is comprised of Crypto and Bacil.

3.8 Storm-petrel Example

The "Storm-petrel Example" is based on Bicknell et al. [1] and shows MixSIAR in a *non-diet application: movement*. Here the mixture data are juvenile (non-breeding) birds and the source data are adult (breeding) birds from 3 breeding colonies. Discrimination is set to zero, making the reasonable assumptions that 1) juvenile and adult birds feeding in the same region will look isotopically identical, and 2) discrimination is the same for juvenile and adult birds. We have:

- 2 biotracers (δ^{13}C, δ^{15}N)
- 1 fixed effect (Region)
- Raw source data

See "stormpetrel_consumer.csv", "stormpetrel_sources.csv", and "stormpetrel_discrimination.csv".

3.9 Snail Example

The "Snail Example" data is from Yanes et al. [28] and demonstrates how MixSIAR handles *1-dimensional data*. The preference of land snails between C3 and CAM plants is analyzed with:

- 1 biotracer (δ^{13}C)

- Raw source data

See "snail_consumer.csv", "snail_sources.csv", and "snail_discrimination.csv".

3.10 Mantis Example

The "Mantis Example" data is from deVries et al. [4] and demonstrates two additional features in MixSIAR:

- Multiple informative priors
- Combining sources *a posteriori* using the `combine_sources` function

3.10.1 Load mix data

```
library(MixSIAR)

# find mantis consumer data file within MixSIAR package
mix.filename <- system.file("extdata", "mantis_consumer.csv", package = "MixSIAR")

# Load mixture data
mix <- load_mix_data(filename=mix.filename,
                     iso_names=c("d13C","d15N"),
                     factors="Habitat",
                     fac_random=FALSE,
                     fac_nested=FALSE,
                     cont_effects=NULL)
```

3.10.2 Load source data

```
# find mantis source data file
source.filename <- system.file("extdata", "mantis_source.csv", package = "MixSIAR")

# Load source data
source <- load_source_data(filename=source.filename,
                           source_factors=NULL,
                           conc_dep=FALSE,
                           data_type="means", mix)
```

3.10.3 Load discrimination data

```
# find discrimination data file
discr.filename <- system.file("extdata", "mantis_discrimination.csv", package = "MixSIAR
# Load discrimination data
discr <- load_discr_data(filename=discr.filename, mix)
```

3.10.4 Plot data

```
plot_data(filename="isospace_plot",
          plot_save_pdf=TRUE,
          plot_save_png=FALSE,
          mix,source,discr)
```

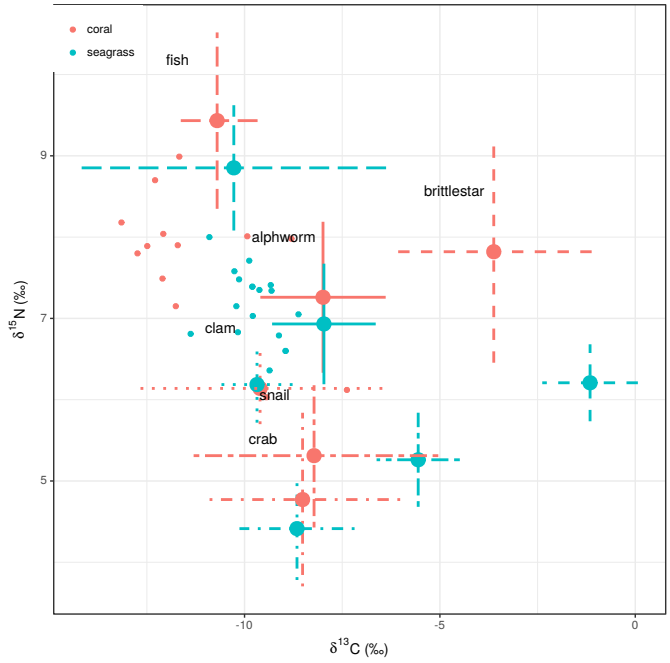


Figure 24: **Stable isotope input for the Mantis Example.** Both consumer and source data are by Habitat (color) and source data are labeled. Error bars indicate combined source and discrimination uncertainty ± 1 SD.

3.10.5 Define and plot priors

```
# default "UNINFORMATIVE" / GENERALIST prior (alpha = 1)
alpha.unif <- rep(1, source$n.sources)
plot_prior(alpha.prior=alpha.unif,
```

```
          source=source,
          filename="prior_uninf")
```

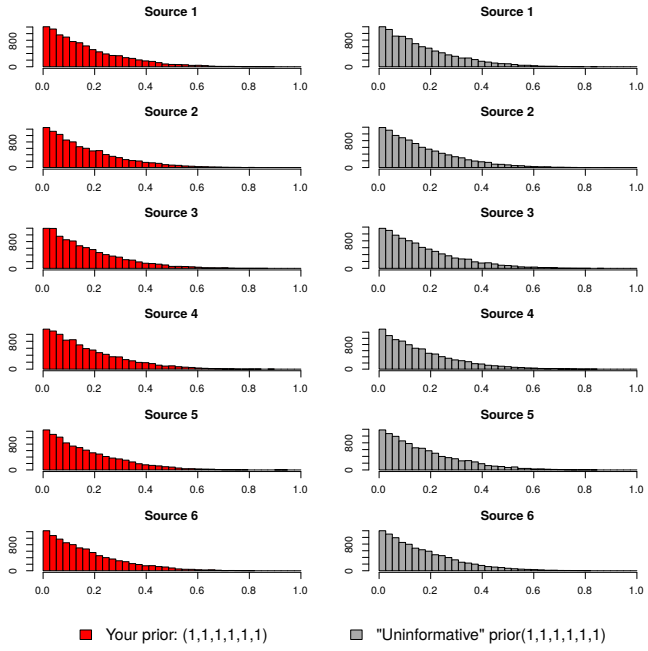


Figure 25: "Uninformative"/generalist prior for mantis shrimp example with 6 sources.

But wait... the GENERALIST prior does not reflect our a priori ecological knowledge! The current thought is that 'smasher' mantis shrimp SPECIALIZE on hard-shelled prey, because they have specialized hammer-like clubs allowing them to break open shells. Let's construct a "specialist" informative prior reflecting the expectation that *N. bredini* consumes primarily hard-shelled prey. This will be a more conservative test of the hypothesis that *N. bredini* specializes on hard-shelled prey. We rescale the α_k so they sum to K, the number of sources:

```
# 'specialist' informative prior
# hard-shelled prey get 4x weight of soft-bodied prey,
# based on dietary observations of N. bredini (Caldwell et al. 1989).
alpha.spec <- c(1,1,4,4,1,4)
# rescale so sum(alpha) = n.sources
alpha.spec <- alpha.spec*length(alpha.spec)/sum(alpha.spec)
plot_prior(alpha.prior=alpha.spec,
          source=source,
          filename="prior_specialist")
```

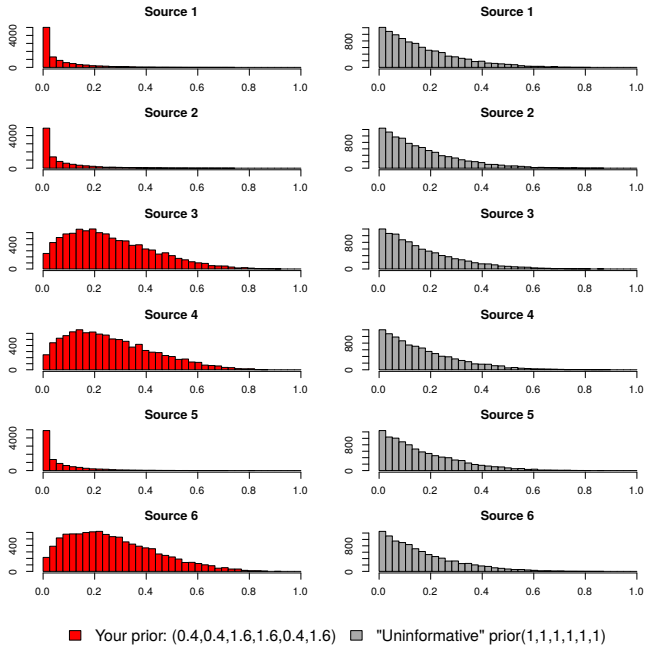


Figure 26: "Specialist" prior for mantis shrimp example, where hard-shelled prey receive 4x weight of soft-bodied prey.

We also have data on prey abundance in the two habitats (coral, seagrass). Let's construct informative priors using prey abundance so that the α_k sum to K, the number of sources. *Note: MixSIAR cannot set different informative priors for different levels of a factor (here, coral vs. seagrass). In order to run the abundance prior model, the only option is to run separate models.*

```
# Seagrass habitat prior, based on prey abundance
alpha.grass <- c(0.35,1.61,0.43,(51.65+0.26),5.18,40.5)*6/100
plot_prior(alpha.prior=alpha.grass, source=source, filename="prior_seagrass")

# Coral habitat prior, based on prey abundance
alpha.coral <- c((14.31+24.74),0.01,15.48,(13.81+4.71),8.44,18.51)*6/100
plot_prior(alpha.prior=alpha.coral, source=source, filename="prior_coral")
```

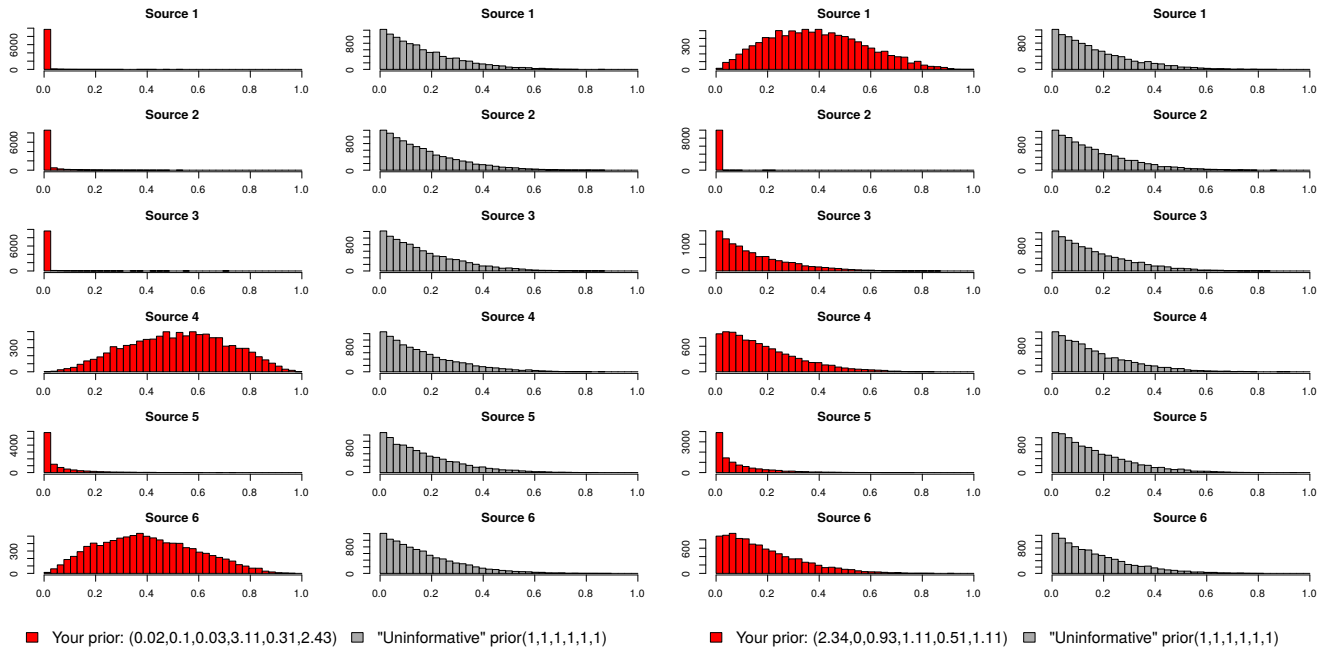


Figure 27: "Abundance" priors for mantis shrimp example, where the α_k reflect prey abundance in the two habitats: seagrass (left) and coral rubble (right).

3.10.6 Write JAGS model file

We'll use the default Resid * Process error structure:

```
model_filename <- "MixSIAR_model.txt"
resid_err <- TRUE
process_err <- TRUE
write_JAGS_model(model_filename, resid_err, process_err, mix, source)
```

3.10.7 Run model

Takes about 40 minutes to run.

```
# Good idea to use 'test' first to check if
#   1) the data are loaded correctly, and
#  2) the model is specified correctly
# jags.1 <- run_model(run="test", mix, source, discr, model_filename,
#                     alpha.prior = alpha.spec, resid_err, process_err)

# After a test run works, increase the MCMC run to a value that may converge
jags.spec <- run_model(run="normal", mix, source, discr, model_filename,
                       alpha.prior = alpha.spec, resid_err, process_err)
```

3.10.8 Confirm MCMC has converged using diagnostics

```
# Create diagnostics, summary statistics, and posterior plots
output_JAGS(jags.spec, mix, source)
```

3.10.9 Combine sources *a posteriori*

In this study, the primary scientific question was whether *N. bredini* specializes on hard-shelled prey. To directly answer this question, we can use the `combine_sources` function to group our original 6 sources into 2 groups: hard-shelled versus soft-bodied.

```
# combine hard-shelled and soft-bodied prey sources
combined <- combine_sources(jags.spec, mix, source, alpha.spec,
              groups=list(hard=c("clam","crab","snail"),
                          soft=c("alphworm","brittlestar","fish")))

# also pass original grouping to combine_sources so we can compare results
original <- combine_sources(jags.spec, mix, source, alpha.spec
              groups=list(alphworm="alphworm",
                          brittlestar="brittlestar",
                          clam="clam",
                          crab="crab",
                          fish="fish",
                          snail="snail"))
```

An important consideration when grouping sources *a posteriori* is that you are effectively changing the prior weighting on the sources. Aggregating uneven numbers of sources will turn an 'uninformative'/generalist prior into an informative one. In the mantis shrimp example, this has no effect because each new group has the same number of sources (3), and each group member had the same prior weight (0.4 for soft-bodied, 1.6 for hard-shelled). To highlight this effect, MixSIAR automatically generates a plot of the new, aggregated prior so you can compare to the original and what an "uninformative"/generalist prior would be with the new number of sources (as if you aggregated sources BEFORE running the model).

```
----------------------------------------------------------------------
*** WARNING ***
Aggregating sources after running the mixing model (a posteriori)
effectively changes the prior weighting on the sources. Aggregating uneven
numbers of sources will turn an 'uninformative'/generalist prior into an
informative one. Please check your new, aggregated prior (red), and note
the difference between it and the original prior  (blue). The right (grey)
column shows what the 'uninformative'/generalist  prior would be if
you aggregate sources before running the mixing model  (a priori).
----------------------------------------------------------------------
```

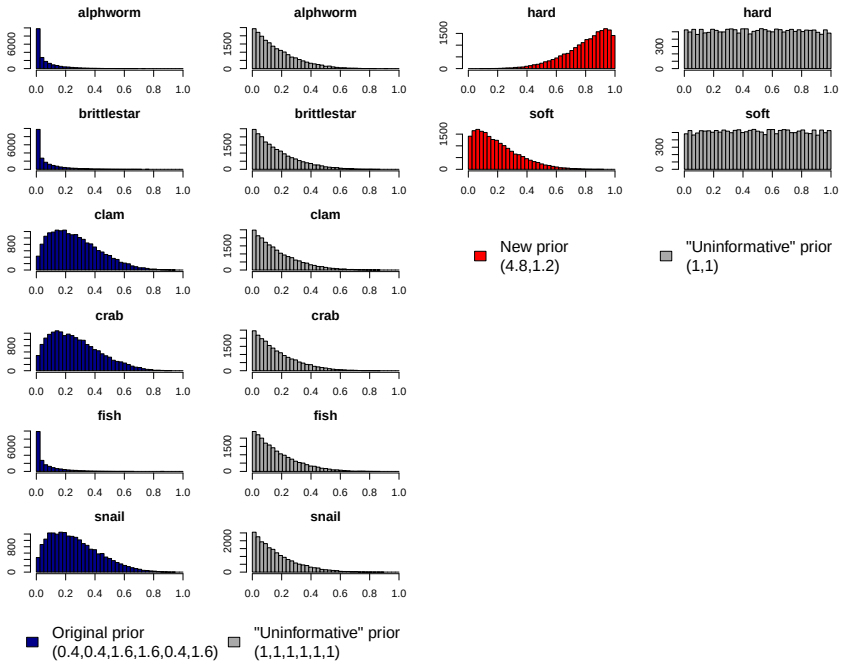


Figure 28: Original, 6-source (blue) and aggregated, 2-source (red) prior distributions, automatically output from `combine_sources` function.

3.10.10 Summary statistics (original 6-source model)

```
# Original 6-source model
summary_stat(original)
```

```
#####################################################################
# Summary Statistics
#####################################################################

                        Mean    SD  2.5%   25%   50%   75% 97.5%
Epsilon.1              0.841 0.435 0.299 0.543 0.743 1.026 1.979
Epsilon.2              2.646 1.091 1.149 1.883 2.452 3.165 5.383
p.alphworm.coral       0.027 0.037 0.000 0.002 0.012 0.038 0.132
p.brittlestar.coral    0.016 0.024 0.000 0.001 0.006 0.020 0.084
p.clam.coral           0.454 0.097 0.255 0.390 0.460 0.523 0.631
p.crab.coral           0.076 0.058 0.006 0.033 0.062 0.105 0.222
p.fish.coral           0.357 0.067 0.237 0.312 0.353 0.399 0.499
p.snail.coral          0.069 0.053 0.006 0.031 0.056 0.093 0.208
p.alphworm.seagrass    0.037 0.062 0.000 0.001 0.009 0.043 0.220
p.brittlestar.seagrass 0.015 0.025 0.000 0.001 0.004 0.018 0.088
p.clam.seagrass        0.570 0.130 0.276 0.491 0.588 0.665 0.774
p.crab.seagrass        0.065 0.066 0.002 0.018 0.042 0.089 0.248
p.fish.seagrass        0.262 0.055 0.155 0.224 0.261 0.298 0.373
p.snail.seagrass       0.052 0.051 0.002 0.015 0.035 0.072 0.189
```

```r
# Aggregated posteriors, 6 sources combined into 2 groups
summary_stat(combined)
```

```
#####################################################################
# Summary Statistics
#####################################################################

                 Mean    SD  2.5%   25%   50%   75% 97.5%
Epsilon.1       0.841 0.435 0.299 0.543 0.743 1.026 1.979
Epsilon.2       2.646 1.091 1.149 1.883 2.452 3.165 5.383
p.hard.coral    0.600 0.070 0.450 0.557 0.603 0.647 0.729
p.soft.coral    0.400 0.070 0.271 0.353 0.397 0.443 0.550
p.hard.seagrass 0.687 0.079 0.500 0.642 0.694 0.740 0.815
p.soft.seagrass 0.313 0.079 0.185 0.260 0.306 0.358 0.500
```

3.10.11 Plot credible intervals

```r
# Original 6-source model
plot_intervals(original,toplot="fac1")
```

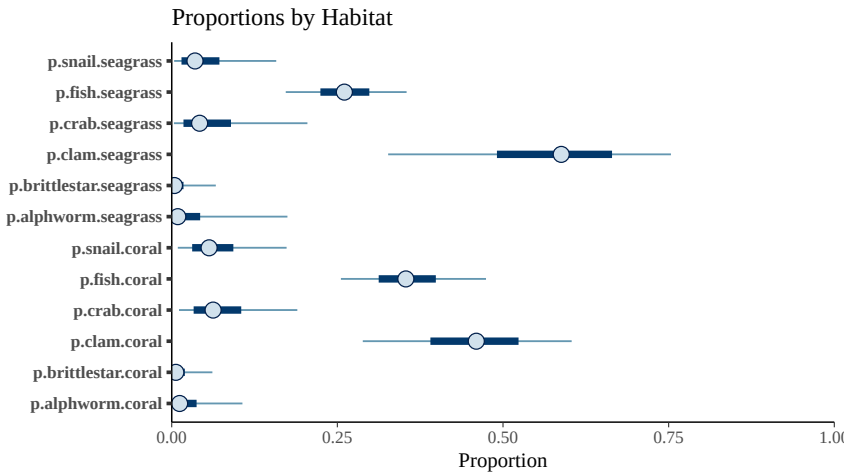


Figure 29: Proportion estimates from original 6-source mantis shrimp model: posterior medians (points), 50% credible intervals (thick lines), and 90% credible intervals (thin lines).

```r
# Aggregated posteriors, 6 sources combined into 2 groups
plot_intervals(combined,toplot="fac1")
```

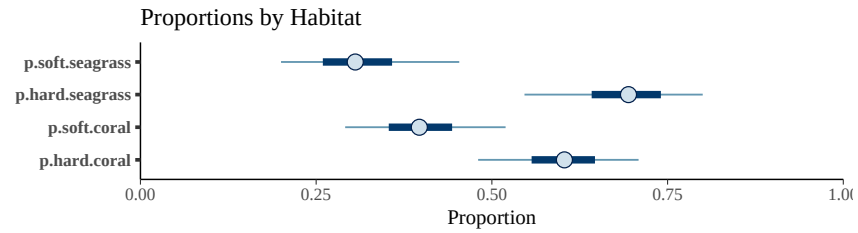


Figure 30: Proportion estimates after aggregating sources into 2 groups, hard-shelled and soft-bodied: posterior medians (points), 50% credible intervals (thick lines), and 90% credible intervals (thin lines).

```
# level 1 = Seagrass
plot_intervals(combined,toplot="fac1",levels=1)
```

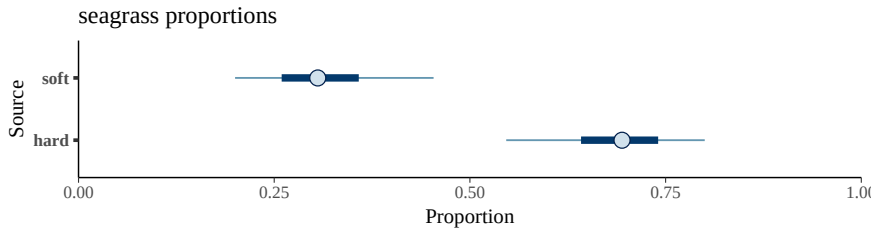


Figure 31: Proportion estimates after aggregating sources into 2 groups, only for Seagrass habitat: posterior medians (points), 50% credible intervals (thick lines), and 90% credible intervals (thin lines).

```
# level 2 = Coral
plot_intervals(combined,toplot="fac1",levels=2)
```

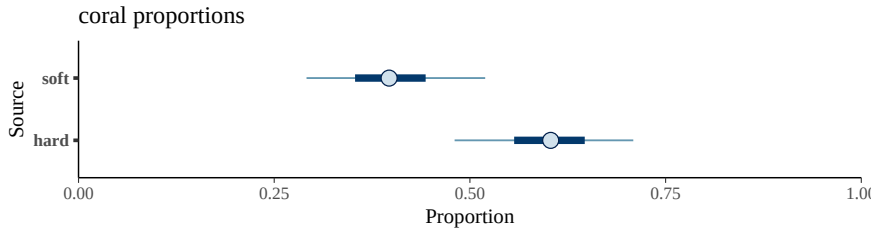


Figure 32: Proportion estimates after aggregating sources into 2 groups, only for Coral habitat: posterior medians (points), 50% credible intervals (thick lines), and 90% credible intervals (thin lines).

3.11 Alligator Example

The "Alligator Example" highlights the main advantage of MixSIAR over previous mixing model software–the ability to include fixed and random effects as covariates explaining variability in mixture proportions, and calculate relative support for multiple models via information criteria (LOO/WAIC) using the `compare_models` function.

Nifong et al. [17] were interested in freshwater vs. marine resource use by alligators. Their specific scientific questions were:

1. What is p_{marine} vs. $p_{freshwater}$?
2. How does p_{marine} vary with the covariates Length, Sex, and Individual?
3. How variable are individuals' diets relative to group-level variability?

Nifong et al. [17] modeling approach:

- group consumers into 8 subpopulations (all combos of Sex x Size Class)
  - 2 sexes {male, female}
  - 4 size classes {small juvenile, large juvenile, subadult, adult}
- run 8 separate mixing models for each using SIAR [19].
- to calculate p_{marine} estimates for the overall population, had to also run a mixing model with all consumers

Instead, using MixSIAR allows the following analysis:

- fit several models with fixed/random/continuous effects
- evaluate relative support for each model using the `compare_models` function

3.11.1 Run models

See "mixsiar_script_alligator.R" for code to run the following 8 models:

1. NULL
2. Habitat (3: fresh, intermediate, marine)
3. Sex (2: male, female)
4. Size class (4: small juv, large juv, sub-adult, adult)
5. Length (continuous effect)
6. Sex + Size class (both as fixed effects)
7. Sex + Length (intercept by sex, same slope)
8. Sex : Size class (create new factor = sex * sclass, closest to original analysis)

3.11.2 Compare models using LOO

The JAGS models should be stored in a list, `jags.mod`, where `jags.mod[[i]]` is the *i*th model object. Confirm that all models have converged, and then we can use `compare_models`.

`compare_models` uses the `loo` package to compute LOO (leave-one-out cross-validation) or WAIC (widely applicable information criterion) for 2 of more fit MixSIAR models. LOO and WAIC are "methods for estimating pointwise out-of-sample prediction accuracy from a fitted Bayesian model using the log-likelihood evaluated at the posterior simulations of the parameter values" [26]. In brief:

- LOO and WAIC are preferred over AIC or DIC
- LOO is more robust than WAIC
- `loo` estimates standard errors for the difference in LOO/WAIC between two models (dLOOic)
- We can calculate the relative support for each model using Akaike weights (p.75 in 3). Interpretation: "an estimate of the probability that the model will make the best predictions on new data, conditional on the set of models considered" [14].

```
# add model names to 'jags.mod'
names(jags.mod) <- c("Null", "Habitat", "Sex", "Size class", "Length",
                     "Sex + Size class", "Sex + Length", "Sex : Length")

# get model comparison table
(comparison.table <- compare_models(jags.mod))
```

	Model	LOOic	se_LOOic	dLOOic	se_dLOOic	weight
5	Length	1678.3	31.3	0.0	NA	0.799
7	Sex + Length	1681.2	31.4	2.9	2.2	0.187
4	Size class	1687.5	31.8	9.2	11.8	0.008
6	Sex + Size class	1689.2	31.5	10.9	12.2	0.003
8	Sex : Length	1690.4	29.8	12.1	13.6	0.002
2	Habitat	1747.9	28.8	69.6	43.4	0.000
3	Sex	1831.3	17.6	153.0	30.1	0.000
1	Null	1834.6	16.7	156.3	31.4	0.000

The model with a continuous effect of Length had the lowest LOOic and received 80% of the Akaike weight, indicating an 80% probability it is the best model (expected predictive performance on new data, out of the models considered, given the data observed). The model with Sex + Length cannot be ruled out (19% weight). dLOOic is the difference in LOOic between each model and the model with lowest LOOic.

We can also plot p_{marine} as a function of Length from the best model:

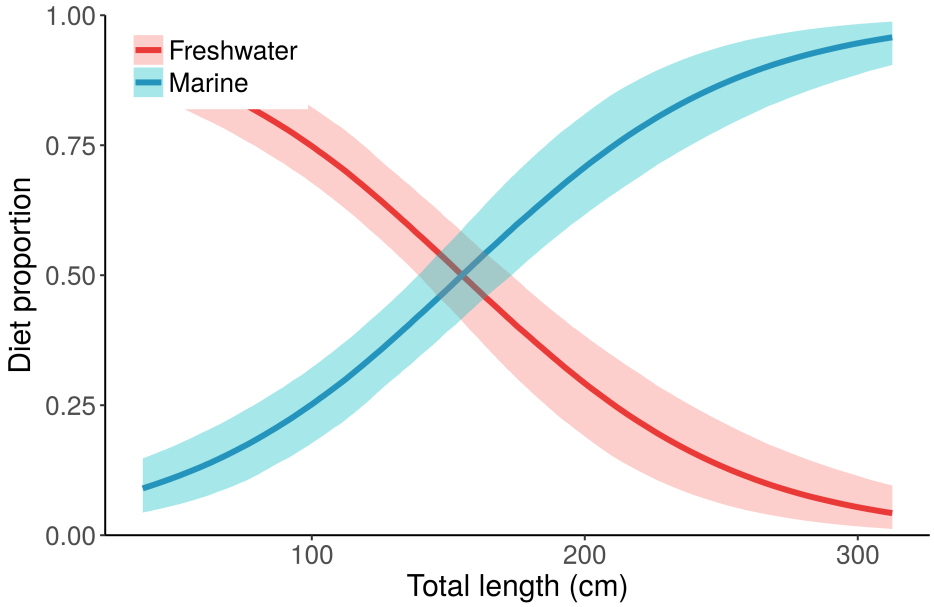


Figure 33: Posterior estimate of p_{marine} as a function of length from Model 5 ('Length'). Solid line indicates median, shading is the 95% CI.

You can calculate posteriors for derived quantities using the MCMC draws as well. Make sure to calculate the quantity for each individual draw. Here, we show the specialization index of Newsome et al. [16] as a function of Length:

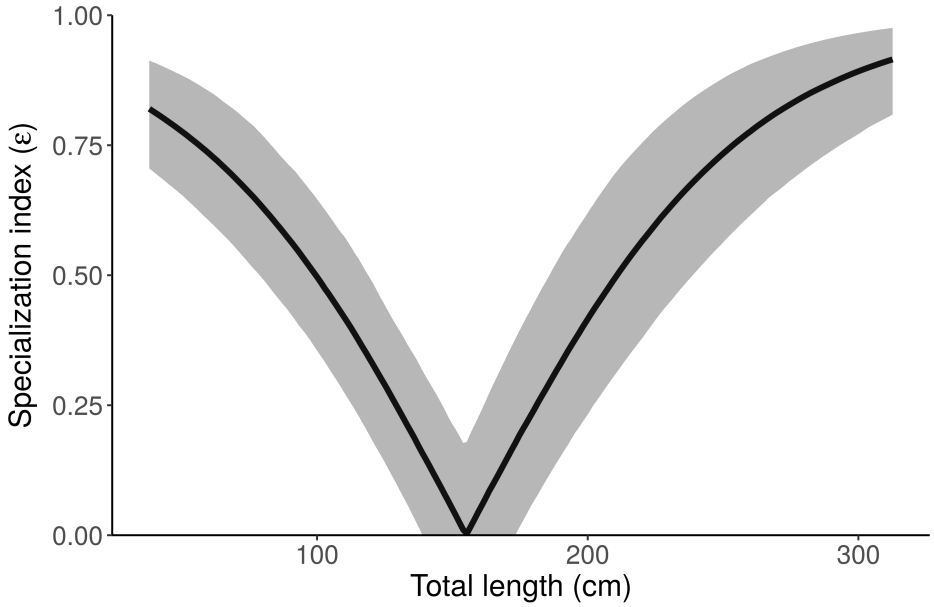


Figure 34: Posterior estimate of ε as a function of length from Model 5. Solid line indicates median, and shading is the 95% CI. Small and large alligators specialize on freshwater and marine prey, respectively, while average length alligators are generalists as they transition from freshwater to marine.

4 Using MixSIAR with your own data

After you've run the working examples, you should be confident that MixSIAR is installed and working correctly, and now you can begin using it to analyze your own biotracer data. You may find it more convenient to use the script version than the GUI.

4.1 Running your own data

The process for running MixSIAR on your data is the same as in the working examples:

1. Open R. Check to see that all of the necessary MixSIAR files and your data files are in your working directory (can use `getwd()`, `setwd()`, and `list.files()` commands in R). See Data file format and loading.
2. Create the MixSIAR GUI using `source("mixsiar_gui.r")` and `mixsiar_gui()`.
3. Load your files using the "Read-in data" buttons.
4. Plot your data with the "Make isospace plot" button. Refer to Section 3.1.4.
5. Choose your MCMC parameters ("# of Chains", "Chain Length", "Burn-in", and "Thin"). Refer to Section 3.1.6.
6. Choose your error structure. See Sections 3.1.7 and 3.7.7, and for more explanation, Stock and Semmens [25].
7. Choose your prior. Refer to Section 3.1.5 for the uninformative prior and Section 3.5.5 on how to construct an informative prior.
8. Plot your prior with the "Plot prior" button.
9. Choose your output options for summary statistics, plots, and diagnostics.
10. Click "RUN MODEL."
11. When the model is finished, click "Process output".
12. Check that the model has converged using the diagnostics. Refer to Section 3.1.9.
13. Look at your results—posterior density plots and summary statistics. See Section 3.1.12.
14. Save your workspace if you wish. See Section 3.1.13.

4.2 Data file format and loading

*Check to see that all of the necessary files are in the **correct format** and in your **working directory***. In all files, extra unused columns are not a problem and column order is not important. You need 3 .csv files:

1. **Mixture/consumer biotracer values** (with covariates if desired). After you load the file, you will tell MixSIAR which columns to use as "Isotopes", Random Effects, and Continuous Effects. Covariates can be numerical or text. If you include two Random Effects, you will be asked "Should MixSIAR run a hierarchical analysis?" You should answer "Yes" if your second covariate is clearly nested within the first (e.g. Pack within Region), and "No" if the two are not nested (e.g. Month and Sex). *The labels you use in the mixture file must match those in the source and discrimination files* (i.e. if you use 'd13C' here, then you must use 'Meand13C', 'SDd13C', and 'Concd13C' in the source and discrimination files). Here is the top of "wolves_consumer.csv":

d13C	d15N	Region	Pack
-23.68	7.96	1	1
-23.61	7.78	1	1
-23.76	7.72	1	1
-23.61	7.77	1	1
-24.37	7.33	1	1
-23.93	7.65	1	1

2. **Source isotope values** (with covariate and/or concentration dependence optional). There are several options for the source data file, depending on the type of data you have. Three different source data scenarios are demonstrated in the four working examples.

- *"Raw source data"—original measurements*. Column labels must match those in the mixture file (e.g. 'd13C'). The Lake Example had "Raw source data" without a covariate ("lake_sources.csv"):

Source	d13C	d15N
Surface POM	-27.861	-0.724
Surface POM	-25.534	-3.627
Surface POM	-29.447	1.63
Surface POM	-29.538	-3.245
Surface POM	-29.758	0.393
Subsurface POM	-35.225	-0.736
Subsurface POM	-34.928	-1.289
Subsurface POM	-34.325	1.921
Subsurface POM	-32.309	-0.515
Terrestrial Detritus	-31.933	-2.495
Terrestrial Detritus	-29.082	-2.504
Terrestrial Detritus	-29.997	-2.333
Terrestrial Detritus	-30.608	-1.481

- *"Source means+SDs"—summary statistics*. Add "Mean" and "SD" to the biotracer labels you used in the mixture file (i.e. if you used 'd13C' before, then you must use 'Meand13C' and 'SDd13C'). You cannot enter "0" as a standard deviation. **You must also have a column titled "n" with the sample size of each source estimate**. If you do not know the sample size, entering an arbitrary large number (e.g. 1000) will have MixSIAR effectively treat the means and SDs as known parameters instead of fitting them with the data. The Geese Example had "Source means+SDs" data with concentration dependence and without a covariate ("geese_sources.csv"):

Sources	Meand15N	SDd15N	Meand13C	SDd13C	Concd15N	Concd13C	n
Zostera	6.4889844701	1.4594632432	-11.170227684	1.2149561571	0.0297	0.3593	14
Grass	4.4321601033	2.2680708955	-30.8798439532	0.6413182104	0.0355	0.4026	14
U.lactuca	11.1926127957	1.1124384641	-11.170900037	1.9593305583	0.0192	0.2098	14
Enteromorpha	9.8162797509	0.8271039322	-14.0570109233	1.1724676718	0.0139	0.1844	14

- *"Concentration dependence"* (optional). Add "Conc" to the isotope labels you used in the mixture file (i.e. 'd13C' becomes 'Concd13C'). These can simply be the elemental concentrations for each source (e.g. [C], [N]), or can incorporate digestibility as in Koch and Phillips [11] (e.g. Digest [C], Digest [N]). Concentration dependence is built into the MixSIAR model following Equation 1 in Parnell et al. [19]. The Geese Example included concentration dependence for "Source means+SDs" ("geese_sources.csv", above), but you can also add 'Concd13C', etc. to "Raw source data" in the same way. Before loading the source data file, make sure "Do you have Concentration Dependence data?" is marked "Yes".

- *"Covariate"* (optional). Either "Source means+SDs" or "Raw source data" can have a covariate column, as in the Wolves Example ("wolves_sources.csv", below). The covariate label must match that in the mixture data file. Before loading the source data file, make sure "Does your source data vary by <covariate>?" is marked "Yes".

	Region	Meand13C	SDd13C	Meand15N	SDd15N	n
Deer	1	-26.88	1.1	3.07	1.35	24
Deer	2	-27.15	0.67	2.8	1.14	37
Deer	3	-27.47	0.75	2.76	2.32	9
Salmon	1	-18.58	1.34	12.26	1.18	6
Salmon	2	-22.38	2.85	11.92	1.12	5
Salmon	3	-22.38	2.85	11.92	1.12	5
Marine Mammals	1	-14.7	1.08	16.26	1.06	7
Marine Mammals	2	-14.47	0.95	15.55	1.69	6
Marine Mammals	3	-12.48	0.75	16.21	1.78	6

3. **Discrimination data**. Add "Mean" and "SD" to the isotope labels you used in the mixture file (i.e. if you used 'd13C' before, then you must use 'Meand13C' and 'SDd13C'). Discrimination data is assumed to be of the form Mean $\pm$ SD, and "0" SD is permitted, as in "wolves_discrimination.csv":

4.3 Using the MixSIAR script version

If you are familiar with R, you can use the script version of MixSIAR instead of the GUI. The script version has a couple advantages:

1. *Repeatability*: You can run different models and have a record of the commands that created each one. There are many reasons you'd want to do this. For example, you may want to compare model results with an uninformative prior vs. an informative prior, one error term option vs. another, grouping sources a priori vs. a posteriori, etc.

2. *Speed*: Clicking through the GUI buttons can get onerous after a while. You can modify the example .R files and run your entire analysis with `source("your_file.R")`.

3. *Installation ease*: Some users aren't able to install the GTK+ software that the GUI depends on. It may be worth figuring out the script version (R skills!) instead of figuring out how to get GTK+ installed.

The script examples are written as vignettes, which you can access via:

```
browseVignettes("MixSIAR")
```

We have also included clean, runnable .R scripts for all the working examples in the `example_scripts` folder of the MixSIAR package install. You can locate these scripts with:

```
# find MixSIAR directory
mixsiar.dir <- find.package("MixSIAR")
# folder with script files
paste0(mixsiar.dir,"/example_scripts")
# run Geese example
source(paste0(mixsiar.dir,"/example_scripts/mixsiar_script_geese.R"))
```

The script is largely the same as the GUI, with only a couple differences detailed below (using the Geese Example).

4.3.1 Load mix data

Fixed and random effects are both entered as "`factors=`", and then for a random effect "`fac_random=TRUE`", for a fixed effect "`fac_random=FALSE`".

```
?load_mix_data
mix <- load_mix_data(filename="geese_consumer.csv",
                             iso_names=c("d13C","d15N"),
                             factors="Group",
```

```
                                fac_random=FALSE,
                                fac_nested=FALSE,
                                cont_effects=NULL)
```

4.3.2 Load source data

```
?load_source_data
source <- load_source_data(filename="geese_sources.csv",
                           source_factors=NULL,
                           conc_dep=TRUE,
                           data_type="means", mix)
```

- `source_factors`: Are your source data by a factor (must match one of the `factors` in mixture data).
- `conc_dep`=TRUE/FALSE: Do you have concentration dependence data?
- `data_type`="means"/"raw": Do you have raw source data, or summary statistics (Means, SD, and n)?

4.3.3 Load discrimination data

```
discr <- load_discr_data(filename="geese_discrimination.csv", mix)
```

4.3.4 Make isospace plot

```
?plot_data
plot_data(filename="isospace_plot", plot_save_pdf=TRUE,
          plot_save_png=FALSE, mix,source,discr)
```

4.3.5 Calculate normalized surface area (if 2 biotracers)

```
if(mix$n.iso==2) calc_area(source=source,mix=mix,discr=discr)
```

Brett [2] described the interaction between the "uninformative" prior and the shape of the mixing polygon (which arises from the positions of the source means and their variances) as a bias of mixing models. This phenomenon may be better described as weakly informative data, but we agree that Brett's surface area metric may be useful in recognizing when this may be a problem for a mixing model.

The `calc_area` function calculates the normalized surface area as defined by Brett [2], except it combines the source and discrimination variance ($\sqrt{\sigma^2_{source} + \sigma^2_{discr}}$), as that is what is used in the mixing model equations.

4.3.6 Plot prior

Uninformative case:

```
plot_prior(alpha.prior=1,source)
```

Informative case (from Killer Whale Example):

```
kw.alpha <- c(10,1,0,0,3) # Our 14 fecal samples were 10, 1, 0, 0, 3
kw.alpha <- kw.alpha*length(kw.alpha)/sum(kw.alpha)
kw.alpha[which(kw.alpha==0)] <- 0.01
plot_prior(alpha.prior=kw.alpha,source=source,plot_save_pdf=TRUE,
           plot_save_png=FALSE,filename="prior_plot")
```

4.3.7 Write JAGS model

```
?write_JAGS_model
model_filename <- "MixSIAR_model.txt"
resid_err <- TRUE
process_err <- FALSE
write_JAGS_model(model_filename, resid_err, process_err, mix, source)
```

The GUI has 3 error structure options [25]. To create them in the script version:

1. Process * Resid: `resid_err=TRUE`, `process_err=TRUE`
2. Residual only: `resid_err=TRUE`, `process_err=FALSE`
3. Process only (N = 1): `resid_err=FALSE`, `process_err=TRUE`

4.3.8 Run JAGS model

Note 1: if using an informative prior, you need to set `alpha.prior` here.

```
?run_model
jags.1 <- run_model(run="test",mix,source,discr,model_filename,
                    alpha.prior = 1,resid_err,process_err)
```

Note 2: MixSIAR normalizes the mixture and source tracer data before running the JAGS model. This allows the same priors to be used regardless of scale of the tracer data, without using the data to select the prior (i.e. by setting the prior mean equal to the sample mean). Normalizing the tracer data does not affect the proportion estimates (p_k), but does affect users seeking to plot the posterior predictive distribution for their data. For each tracer, we calculate the pooled mean and SD of the mix and source data, then subtract these pooled means and SDs from the mix and source data, and divide by the pooled SD. For details, see lines 226-269 of `run_model.R`.

The diffuse Normal and Gamma priors on the source means and precisions:

$$\mu^s_{jk} \sim N(0, 1000) \text{ and } \frac{1}{\omega^{s\,2}_{jk}} \sim Gamma(.001, .001),$$

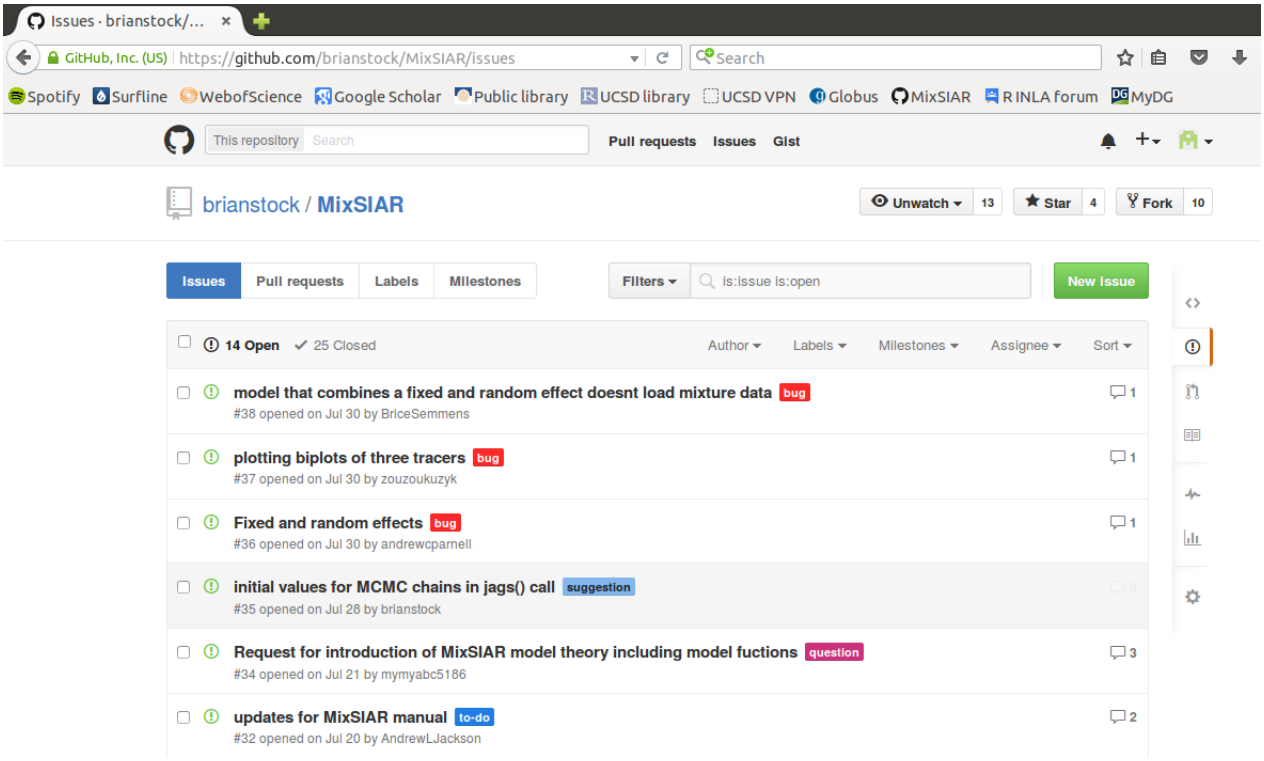
work well for stable isotope and fatty acid tracers, because their order of magnitude is 10^{-1}-10^{1}. For other tracer types that are of larger orders of magnitude (e.g. element concentrations used in sediment mixing can be 10^{3}-10^{5}, Nosrati et al. 2014), these priors would not work. Instead of using the data to set the prior, we scale the data so the same prior can be used regardless of data scale.

4.3.9 Process output

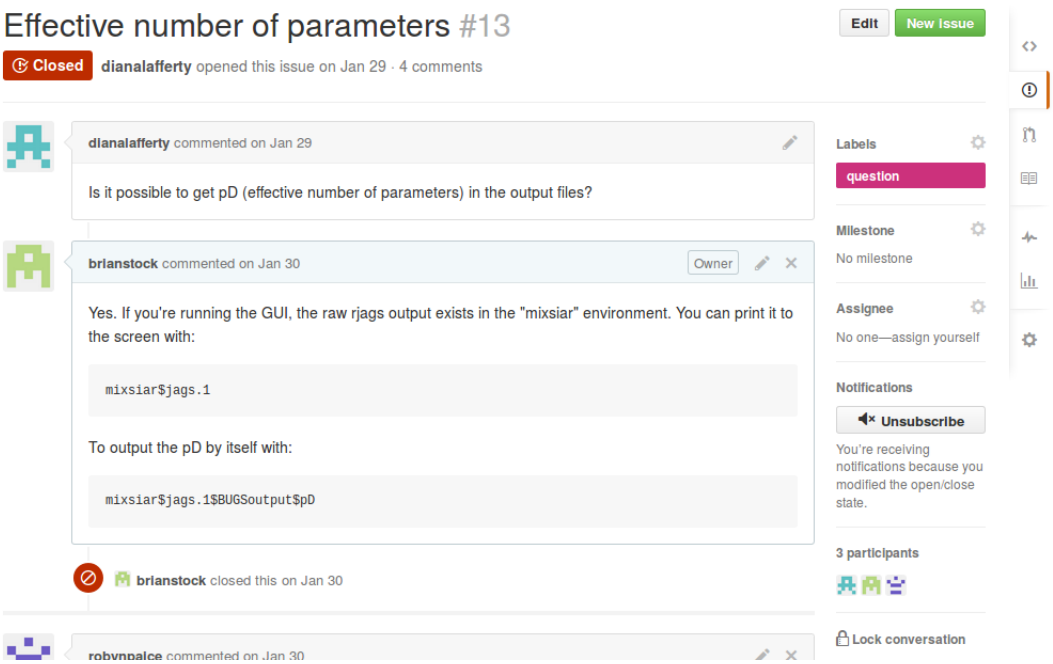
```
?output_JAGS
output_JAGS(jags.1, mix, source)
```

4.4 GitHub Issues page

We would love to hear feedback, positive or negative, if you use MixSIAR. Drawing our attention to errors helps us fix them, and hearing how people are using MixSIAR can help us improve the model in the future. If you have a question, bug report, or suggestion, please use the GitHub Issues page:



The GitHub Issues setup is a very convenient way to ask questions and file bug reports. Unlike email, it provides a community forum, where you can see previous questions/issues and get feedback from many people! See below for an example:



You can also contact the authors at b1stock@ucsd.edu or semmens@ucsd.edu.

4.5 Citing MixSIAR

If you use MixSIAR results in publications, please cite the MixSIAR manual as (similar to how you cite R) :

> Stock, B. C. and B. X. Semmens (2016). MixSIAR GUI User Manual. Version 3.1. https://github.com/brianstock/MixSIAR/. doi:10.5281/zenodo.47719.

A manuscript introducing MixSIAR will be available in the near future! Until then, most of the math underlying these models is in [18]:

> Parnell, A. C., Phillips, D. L., Bearhop, S., Semmens, B. X., Ward, E. J., Moore, J. W., Jackson, A. L., Grey, J., Kelly, D. J., and Inger, R. (2013). Bayesian stable isotope mixing models. Environmetrics, 24(6), 387-399.

The primary citation for Bayesian mixing models is [15]:

> Moore, J. W. and Semmens, B. X. (2008). Incorporating uncertainty and prior information into stable isotope mixing models. Ecology Letters, 11(5), 470-480.

The residual error term was introduced by [19]:

> Parnell, A. C., Inger, R., Bearhop, S., and Jackson, A. L. (2010). Source partitioning using stable isotopes: coping with too much variation. PLoS One, 5(3), e9672.

If you are using a hierarchical structure/random effects (Wolves Example), see [24]:

> Semmens, B. X., Ward, E. J., Moore, J. W., and Darimont, C. T. (2009). Quantifying inter-and intra-population niche variability using hierarchical Bayesian stable isotope mixing models. PLoS One, 4(7), e6187.

If you are using continuous effects, see [5]:

> Francis, T. B., Schindler, D. E., Holtgrieve, G. W., Larson, E. R., Scheuerell, M. D., Semmens, B. X., and Ward, E. J. (2011). Habitat structure determines resource use by zooplankton in temperate lakes. Ecology letters, 14(4), 364-372.

Hopkins and Ferguson [9] made the case for including covariance in the error structure, by default included in all MixSIAR models other than "Process Error only (N=1)":

> Hopkins, J. B., and J. M. Ferguson. 2012. Estimating the diets of animals using stable isotopes and a comprehensive Bayesian mixing model. PLoS One 7:e28478.

Stock and Semmens [25] explain and demonstrate the differences in the error terms ("Process Error only" vs. "Process * Resid" vs. "Resid only"):

> Stock, B. C., and Semmens, B. X. (2016). Unifying error structures in commonly used biotracer mixing models. Ecology, 97(10), 2562-2569.

5 Introductions to Bayesian Analysis

Markov Chain Monte Carlo and Applied Bayesian Statistics: a short course
http://www.stats.ox.ac.uk/~cholmes/Courses/BDA/bda_mcmc.pdf

Getting Started with JAGS, rjags, and Bayesian Modelling
http://jeromyanglim.blogspot.com/2012/04/getting-started-with-jags-rjags-and.html

Gibbs Sampling Made Easy
http://xcorr.net/2011/07/13/gibbs-sampling-made-easy-jags-rkward-coda/

A Primer on Bayesian Statistics in Health Economics and Outcomes Research
http://www.shef.ac.uk/content/1/c6/02/55/92/primer.pdf

Bayesian Modeling in the Social Sciences: an Introduction to Markov-Chain Monte Carlo
http://jackman.stanford.edu/mcmc/icpsr99.pdf

5.1 Convergence Diagnostics

Patrick Lam notes
http://www.people.fas.harvard.edu/~plam/teaching/methods/convergence/convergence_print.pdf

SAS Support (trace plot analysis description)
http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_introbayes_sect008.htm

References

[1] Bicknell, A. W. J., M. E. Knight, D. T. Bilton, M. Campbell, J. B. Reid, J. Newton, and S. C. Votier. 2014. Intercolony movement of pre-breeding seabirds over oceanic scales: Implications of cryptic age-classes for conservation and metapopulation dynamics. Diversity and Distributions **20**:160–168.

[2] Brett, M. T. 2014. Resource polygon geometry predicts Bayesian stable isotope mixing model bias. Marine Ecology Progress Series **514**:1–12.

[3] Burnham, K. P., and D. R. Anderson. 2002. Model selection and multimodel inference: a practical information-theoretic approach. Springer Science & Business Media.

[4] deVries, M. S., B. C. Stock, J. H. Christy, G. R. Goldsmith, and T. E. Dawson. 2016. Specialized morphology corresponds to a generalist diet: linking form and function in smashing mantis shrimp crustaceans. Oecologia **182**:429–442.

[5] Francis, T. B., D. E. Schindler, G. W. Holtgrieve, E. R. Larson, M. D. Scheuerell, B. X. Semmens, and E. J. Ward. 2011. Habitat structure determines resource use by zooplankton in temperate lakes. Ecology Letters **14**:364–372.

[6] Galloway, A., M. Eisenlord, M. Dethier, G. Holtgrieve, and M. Brett. 2014. Quantitative estimates of isopod resource utilization using a bayesian fatty acid mixing model. Marine Ecology Progress Series **507**:219–232.

[7] Galloway, A. W., S. J. Taipale, M. Hiltunen, E. Peltomaa, U. Strandberg, M. T. Brett, and P. Kankaala. 2014. Diet-specific biomarkers show that high-quality phytoplankton fuels herbivorous zooplankton in large boreal lakes. Freshwater Biology **59**:1902–1915.

[8] Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin. 2014. Bayesian data analysis. Taylor & Francis.

[9] Hopkins, J. B., and J. M. Ferguson. 2012. Estimating the diets of animals using stable isotopes and a comprehensive Bayesian mixing model. PLoS One **7**:e28478.

[10] Inger, R., G. D. Ruxton, J. Newton, K. Colhoun, J. A. Robinson, A. L. Jackson, and S. Bearhop. 2006. Temporal and intrapopulation variation in prey choice of wintering geese determined by stable isotope analysis. Journal of Animal Ecology **75**:1190–1200.

[11] Koch, P. L., and D. L. Phillips. 2002. Incorporating concentration dependence in stable isotope mixing models: a reply to Robbins, Hilderbrand and Farley (2002). Oecologia **133**:14–18.

[12] Marin, X. 2013. ggmcmc: Graphical tools for analyzing Markov Chain Monte Carlo simulations from Bayesian inference. URL: http://xavier-fim.net/packages/ggmcmc .

[13] McCauley, D. J., H. S. Young, R. B. Dunbar, J. A. Estes, B. X. Semmens, and F. Micheli. 2012. Assessing the effects of large mobile predators on ecosystem connectivity. Ecological Applications **22**:1711–1717.

[14] McElreath, R. 2016. Statistical rethinking: a Bayesian course with examples in R and Stan. CRC Press.

[15] Moore, J. W., and B. X. Semmens. 2008. Incorporating uncertainty and prior information into stable isotope mixing models. Ecology Letters **11**:470–480.

[16] Newsome, S. D., J. D. Yeakel, P. V. Wheatley, and M. T. Tinker. 2012. Tools for quantifying isotopic niche space and dietary variation at the individual and population level. Journal of Mammalogy **93**:329–341.

[17] Nifong, J. C., C. A. Layman, and B. R. Silliman. 2015. Size, sex and individual-level behaviour drive intrapopulation variation in cross-ecosystem foraging of a top-predator. Journal of Animal Ecology **84**:35–48.

[18] Parnell, A., D. L. Phillips, S. Bearhop, B. X. Semmens, E. J. Ward, J. W. Moore, A. L. Jackson, J. Grey, D. J. Kelly, and R. Inger. 2013. Bayesian stable isotope mixing models. Environmetrics **24**:387–399.

[19] Parnell, A. C., R. Inger, S. Bearhop, and A. L. Jackson. 2010. Source partitioning using stable isotopes: Coping with too much variation. PLoS ONE **5**.

[20] Phillips, D. L., and P. L. Koch. 2002. Incorporating concentration dependence in stable isotope mixing models. Oecologia **130**:114–125.

[21] Plummer, M., 2003. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. Pages 1–10 *in* Proceedings of the 3rd International Workshop on Distributed Statistical Computing. URL http://mcmc-jags.sourceforge.net/.

[22] R Core Team, 2015. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

[23] Semmens, B. X., B. C. Stock, A. Jackson, E. J. Ward, A. Parnell, D. Phillips, R. Inger, and S. Bearhop, in prep. MixSIAR: a new generation of Bayesian mixing model.

[24] Semmens, B. X., E. J. Ward, J. W. Moore, and C. T. Darimont. 2009. Quantifying inter- and intra-population niche variability using hierarchical Bayesian stable isotope mixing models. PloS one **4**:e6187.

[25] Stock, B., and B. Semmens. 2016. Unifying error structures in commonly used biotracer mixing models. Ecology **97**:2562–2569.

[26] Vehtari, A., A. Gelman, and J. Gabry. 2017. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Statistics and Computing **27**:1413–1432.

[27] Ward, E. J., B. X. Semmens, and D. E. Schindler. 2010. Including source uncertainty and prior information in the analysis of stable isotope mixing models. Environmental Science and Technology **44**:4645–4650.

[28] Yanes, Y., M. P. Asta, M. Ibáñez, M. R. Alonso, and C. S. Romanek. 2013. Paleoenvironmental implications of carbon stable isotope composition of land snail tissues. Quaternary Research **80**:596–605.