MixSIAR GUI User Manual

version 1.0

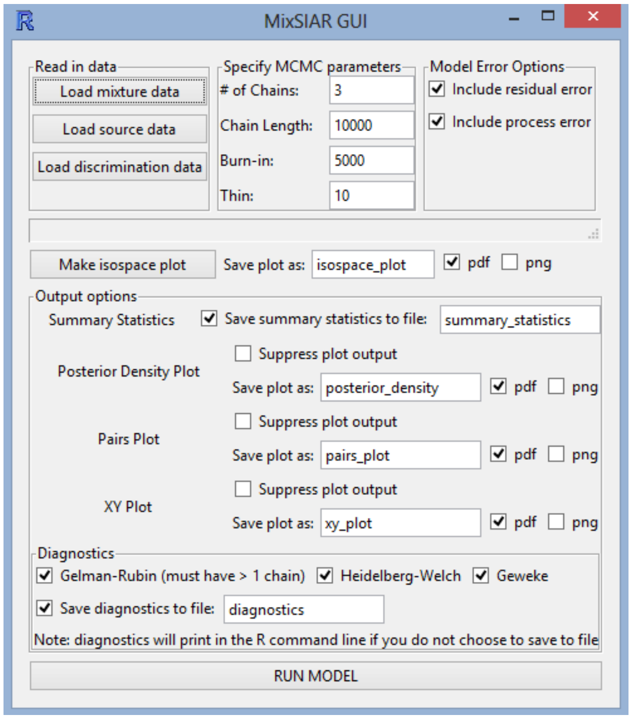
Brian Stock and Brice Semmens

October 2013

1 Introduction

1.1 What is the MixSIAR GUI?

The MixSIAR GUI is a graphical user interface (GUI) that allows you to analyze stable isotope data using the MixSIAR model framework [?]. The GUI and model code are written in the open source languages R and JAGS (Just Another Gibbs Sampler), which are freely available online and can be installed on machines running Mac OS X, Microsoft Windows, and Linux (jump to Installation).



Specify MCMC parameters

# of Chains: 3

Chain Length: 10000

Burn-in: 5000

Thin: 10

Model Error Options

☑ Include residual error

☑ Include process error

Save plot as: isospace_plot ☑ pdf □ png

Output options

Summary Statistics ☑ Save summary statistics to file: summary_statistics

Posterior Density Plot □ Suppress plot output

Save plot as: posterior_density ☑ pdf □ png

Pairs Plot □ Suppress plot output

Save plot as: pairs_plot ☑ pdf □ png

XY Plot □ Suppress plot output

Save plot as: xy_plot ☑ pdf □ png

Diagnostics

☑ Gelman-Rubin (must have > 1 chain) ☑ Heidelberg-Welch ☑ Geweke

☑ Save diagnostics to file: diagnostics

Note: diagnostics will print in the R command line if you do not choose to save to file

Figure 1: **MixSIAR GUI.**

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

Stable isotope analysis has become an important tool to ecologists, since it can be used to determine diet composition, population structure, and animal movement. MixSIAR is a

Bayesian mixing model that uses stable isotope data to estimate the proportions of source (prey) contributions to a mixture (consumer). Bayesian mixing models improve isotope analysis by explicitly taking into account uncertainty in source values [?], categorical and/or continuous covariates [?] [?] [?], and prior information [?].

While diet analysis is the most common application of Bayesian stable isotope 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 [?]. See the following papers for the theory behind each new development:

- MixSIR (original Bayesian mixing model, GUI in MATLAB) [?]
- SIAR (residual error, R package) [?]
- Population structure (categorical covariate) [?]
- Uncertainty in source and discrimination[1] values, Concentration dependence [?]
- Continuous covariate [?]
- Multivariate normal residual error, ILR transform, Changing sources [?]

1.3 Current and future features

The current implementation of the MixSIAR model in the MixSIAR GUI includes the following features:

- Any number of isotope values[2] (tested on real data with 1, 2, or 3)
- Source fitting (both for raw source data and means+SD data)
- Categorical covariates (up to 2, modeled as random effects, either nested or independent)
- Continuous covariates (up to 1, theoretically any number)
- Option to include Individual effects (as random effect)
- Error structure options (residual, combined source, or both–default)
- Concentration dependence

Future versions of MixSIAR will allow the user to incorporate the following:

- Prior specification (the current GUI uses uninformative priors)

[1] 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').

[2] We use *'isotope values'* to refer to δ-values throughout.

- Multivariate residual error structure (as in [?])
- Fit discrimination data in the model
- Allow use of data with missing values (e.g. have 3^{rd} isotope values for some consumers but not all)

1.4 How do I get started using the MixSIAR GUI?

We have written the MixSIAR GUI because we feel it can be a useful tool for ecologists wishing to conduct stable isotope analysis on their data. However, before blindly accepting the MixSIAR GUI results, you should be somewhat familiar with the theory behind the model. Take some time to understand, at least on a conceptual level, the assumptions, limitations, and algorithms of Bayesian analysis, Markov Chain Monte Carlo, and Gibbs Sampling 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. Then after you are confident the MixSIAR GUI is installed and working correctly, you can begin using it to analyze your own stable isotope data.

2 Installation

MixSIAR GUI is written using the `gWidgetsRGtk2` package in R. We think `gWidgetsRGtk2` is a pretty nifty, fairly painless way to create cross-platform GUIs in R–many thanks to John Verzani and other contributors. MixSIAR GUI has been tested and runs on Windows (8) and Mac OS X (Lion). It should also work on Linux, but no promises.

Most problems with the install process are due to 1) not having the latest version of R installed, or 2) the GTK+ installation and linking to the `RGtk2` package (which the MixSIAR GUI is dependent on). If you have install issues, obviously try to resolve them yourself first, but feel free to contact the authors at `bcstock@ucsd.edu` or `semmens@ucsd.edu`.

2.1 Windows

Luckily for you, the MixSIAR GUI looks and runs best on Windows!

1. *Check to make sure you have a recent version of R* (3.0.1 current at time of writing, but 2.15.x is ok). If you have never installed R on your computer, download it from `http://www.r-project.org/index.html`. 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, "for most people the best thing to do is to uninstall R, install the new version, copy any installed packages to the library folder in the new installation, run

`update.packages(checkBuilt=TRUE, ask=FALSE)` in the new R and then delete anything left of the old installation.” For more details, see `http://cran.r-project.org/bin/windows/base/rw-FAQ.html#What_0027s-the-best-way-to-upgrade_003f`.

2. *Download and install the latest version of JAGS* (3.4.0 at time of writing). You can download JAGS from `http://mcmc-jags.sourceforge.net/`.

3. *Download all of the files necessary to run MixSIAR GUI.* MixSIAR is currently hosted on EcologyBox: `http://conserver.iugo-cafe.org/user/brice.semmens/MixSIAR`.

4. *Make sure all of the MixSIAR GUI files are in one folder*:
   - build_mix_win.r
   - build_source_win.r
   - check.png
   - mixsiar.r
   - output_JAGS.r
   - plot_continuous_var.r
   - plot_data.r
   - plot_data_one_iso.r
   - red x.png
   - run_model.r
   - write_JAGS_model.r

5. *Open R* ***using the R console*** *and navigate to the folder with the MixSIAR files.* The easiest way to do this is simply double-click on the R Workspace in the .zip folder (.RData file with no name). You can check what your current working directory is with `getwd()`. Running the GUI code from R Studio causes some plotting issues (you have to press Enter for each plot to appear–better to use the R console).

6. *Enter* `source("mixsiar.r")`.

7. *Enter* `mixsiar()`.This will install and load the necessary R packages. **The first time you run the code to install the RGtk2 package, you will be prompted to install GTK+. Follow the automatic prompts and do not interrupt the GTK+ installation!**

8. *Restart R.*

9. *Enter* `source("mixsiar.r")` *and* `mixsiar()` *again.* If JAGS, GTK+, and the required R packages are installed properly, the MixSIAR GUI will 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 then it's tricky to get GTK+ and R to interface properly (you may have to adjust your PATH variable).

2.2 Mac OS X

Unfortunately, the latest version of R (3.0.x) and the `RGtk2` package do not work well on Mac OS X. This is a known problem with `RGtk2` and will hopefully be fixed, since there was no problem with the previous version (see `http://stackoverflow.com/questions/15868860/r-3-0-and-gtk-rgtk2-error`) Other R packages that depend on `RGtk2` have also had this Mac OS X install issue, notably `Rattle` (`http://rattle.togaware.com/rattle-install-troubleshooting.html` and `https://groups.google.com/forum/#!forum/rattle-users`).

This leaves you with 2 options:

1. *Use an older version of R (2.15) and proceed with normal install (#2 for Windows install).* Versions older than 2.15 have had problems with the ggplot2 package dependency though, so I recommend using 2.15.3. Non-current versions of R can be accessed from `http://www.r-project.org/index.html` (Download R > choose CRAN mirror > Download R for Mac OS X > "old" subdirectory > R-2.15.3.pkg). Note: the JAGS progress bar may not work well with this option - R appeared to be frozen but the model ran and the output was produced eventually.

2. *Use a current version of R (3.x) and manually install GTK+ using MacPorts.* This is not easy–you have to install Xcode, Command Line Tools, and MacPorts, then install gtk2 from Terminal, adjust your PATH, and install R packages from source. If you want to try it, read on... These instructions are from jverzani's answer at `http://stackoverflow.com/questions/15868860/r-3-0-and-gtk-rgtk2-error`. I assume that, like our lab computer, you don't already have MacPorts or Xcode installed. But if you do, skip ahead.

   (a) Go to `https://developer.apple.com/xcode/` to download and install Xcode.

   (b) Open Xcode and click on Xcode > Preferences > Downloads, and install Command Line Tools.

   (c) Go to `http://www.macports.org/install.php` and download the appropriate 'pkg' installer for your OS (Mountain Lion 10.8, Lion 10.7, or Snow Leopard 10.6). Install MacPorts. Check that MacPorts is working by opening Terminal and typing `sudo port selfupdate`.

   (d) Open Terminal and type `sudo port install gtk2`.

   (e) Assuming the gtk2 install process goes well, then (still in Terminal) type `export PATH=/opt/local/bin:/opt/local/sbin:$PATH`.

   (f) Download the source code for the R packages `RGtk2` and `cairoDevice` from CRAN (the tar.gz file, here `http://cran.r-project.org/web/packages/RGtk2/index.`

html, and here `http://cran.r-project.org/web/packages/cairoDevice/index.html`).

(g) In Terminal, navigate to the folder where you saved the source files (e.g. `cd Downloads`).

(h) Type `R CMD INSTALL packagefilename` to install `RGtk2` and `cairoDevice`.

(i) Open the R console and load the `RGtk2` and `cairoDevice` libraries to make sure they're installed correctly, by typing `library(RGtk2)` and `library(cairoDevice)`.

(j) Run MixSIAR following the normal install instructions (#2 for Windows install).

2.3 Linux

1. *Check to make sure you have a recent version of R* (3.0.1 current at time of writing, but 2.15.x is ok). If you have never installed R on your computer, download it from `http://www.r-project.org/index.html`. You can check which version of R (and any packages) you are running using the command `sessionInfo()`.
2. *Download and install the latest version of JAGS* (3.4.0 at time of writing). You can download JAGS from `http://mcmc-jags.sourceforge.net/`.
3. *Download and install GTK+.* From the terminal, try `sudo apt-get install libgtk2.0-dev`. You can also see `http://www.gtk.org/download/index.php`. Mac and Windows will automatically install GTK+ when `RGtk2` is loaded, but not Linux, and the GTK+ libraries need to be installed for the MixSIAR GUI to work.
4. *Check if GTK+ is installed correctly*, by opening R, installing and loading the `RGtk2` package. `install.packages("RGtk2")` and `library(RGtk2)`. If you are prompted to install GTK+, say no.
5. *Download all of the files necessary to run MixSIAR GUI.* MixSIAR is currently hosted on EcologyBox: `http://conserver.iugo-cafe.org/user/brice.semmens/MixSIAR`.
6. *Make sure all of the MixSIAR GUI files are in one folder*:
   - build_mix_win.r
   - build_source_win.r
   - check.png
   - mixsiar.r
   - output_JAGS.r
   - plot_continuous_var.r
   - plot_data.r
   - plot_data_one_iso.r

- red x.png
- run_model.r
- write_JAGS_model.r

7. *Open R* ***using the R console*** *and navigate to the folder with the MixSIAR files.* You can check what your current working directory is with `getwd()`. Running the GUI code from R Studio causes some plotting issues (you have to press Enter for each plot to appear–better to use the R console).

8. *Enter* `source("mixsiar.r")`.

9. *Enter* `mixsiar()`.This will install and load the necessary R packages.

10. *Restart R.*

11. *Enter* `source("mixsiar.r")` *and* `mixsiar()` *again.* If JAGS, GTK+, and the required R packages are installed properly, the MixSIAR GUI will 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.

3 Running the working examples

The following working examples will familiarize yourself with the MixSIAR GUI, and it's a good idea to walk through them before playing around with your own data. If you are familiar with SIAR, the Geese Example uses the same data as the SIAR demo and might be good to look at after the Wolves Example, which is aimed at a first-time user.

First check to see that all of the necessary files are in your working directory (File List). A summary of the working examples and MixSIAR features they demonstrate is below:

	Wolves	**Lake**	**Palmyra**	**Geese**
Isotope data	C, N	C, N	C, N	C, N
Categorical covariate(s)	Region, Pack	—	Taxa	Group
Continuous covariate	—	Secchi:Mixed	—	—
Source data	Means + SDs (by Region)	Raw	Raw	Means + SDs
Individual effect	Yes	Yes	Yes	—
Concentration dependence	—	—	—	Yes
Run time	15 min	10 min	10 min	40 min

Table 1: Working examples included with MixSIAR.

3.1 Wolves Example

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

- 2 Isotope values (δ^{13}C, δ^{15}N)
- 2 Categorical covariates (Region, Pack)
- Individual effect
- Source data as means and SDs (by Region)

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 C and 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).
4. Make sure the box at the bottom that says "Include 'Individual' as a random effect" is ticked, and then click "I'm finished"
5. Another box will appear asking if the data is nested/hierarchical (Pack within Region). Click "I'm finished" (default is yes).

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 source 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 means and SDs, 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!* 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'). MixSIAR does not currently take into account uncertainty in discrimination values, but it may in the future!

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 (default is to save the plot as "isospace_plot.pdf"). You can also change the name of the file here.

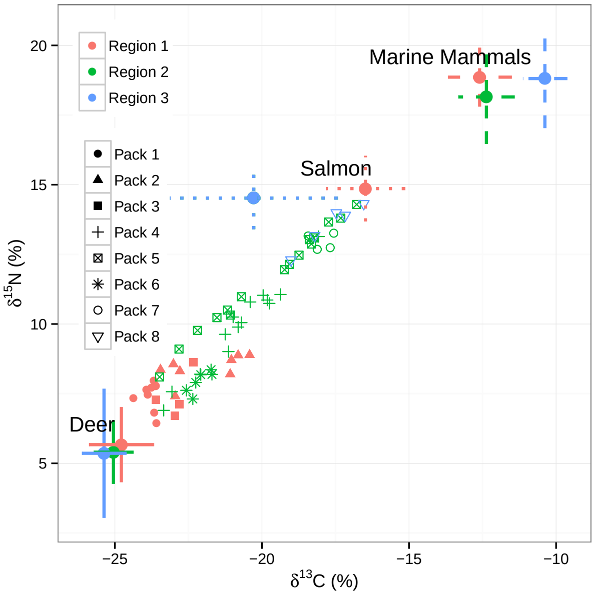


Figure 2: **Stable isotope input for the Wolves Example.** Mixture data (wolves) are by Region (color) and Pack (shape). Source data are by Region and have been adjusted by discrimination means and SDs. Error bars indicate ± 1 SD.

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 sources are way off from the mixture, MixSIAR may not be able to find a solution (if residual error is not included). If residual error is included, then MixSIAR will always find a solution even if it is nonsensical (mixture data outside of the source polygon, error term will be huge). 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.[3] Error bars indicate ± 1 SD.

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 isotope value, MixSIAR will create a 1-D plot. If you have more than 2 isotope values, MixSIAR will create all possible pairwise 2-D plots (e.g. C-N, C-S, and N-S).

3.1.5 Specify MCMC parameters

For the Wolves Example you can set "# of Chains" = 3, "Chain Length" = 50000, "Burn-in" = 25000, and "Thin" = 25. If you want to know more about what these are, see Section 4.2.

3.1.6 Error term options

In the Wolves Example we want both "residual error" and "process error", and they should both already be checked. Generally, you can leave both error terms in the model, unless you have a specific reason to exclude one.

For a discussion of what we mean by "residual error" and "process error", see Error Term Options.

3.1.7 Run MixSIAR

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:

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 3698

Initializing model
   |+++++++                                                                | 20%
```

This is good! It means MixSIAR has created a JAGS model file, passed your data to JAGS, and JAGS is fitting your model. Be patient, this took about 15 minutes on my laptop (Windows 8, Intel Core i5 1.7GHz with 8GB RAM, running other programs). If you're familiar with JAGS and want to check out the model file, "MixSIAR_model.txt" will be in your working directory folder.

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

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 3.

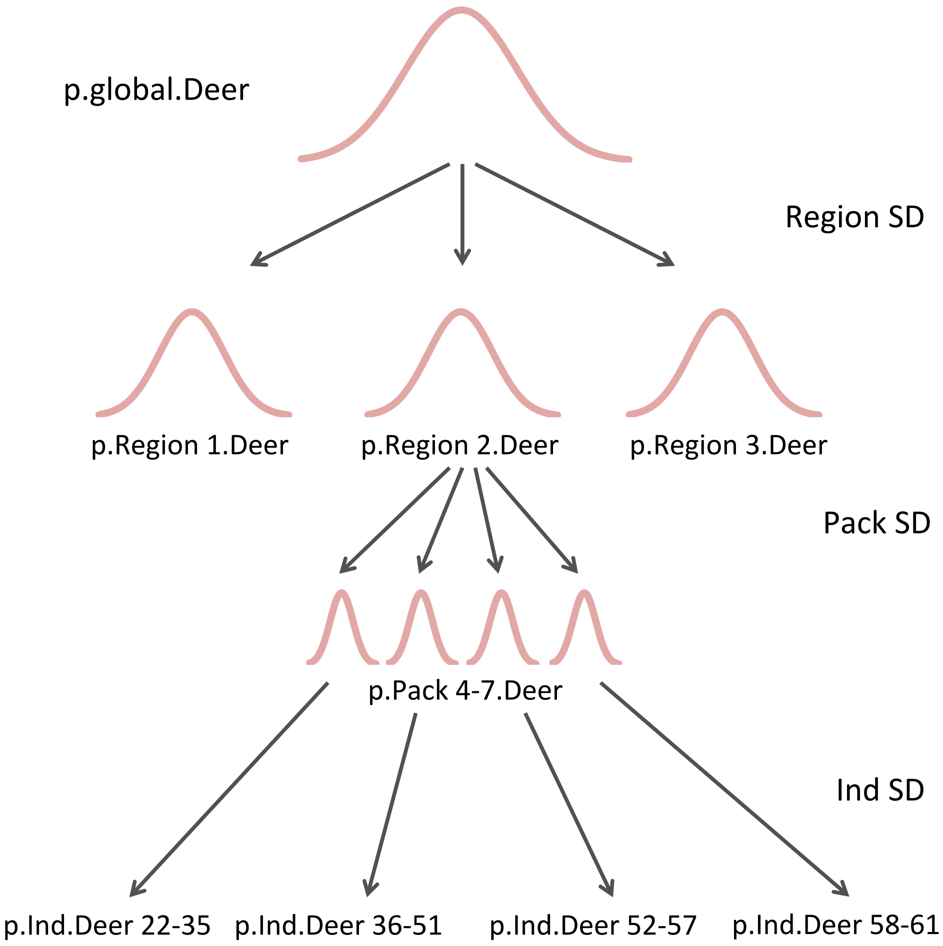


Figure 3: **Model structure and variables in the Wolf Example.** For the diet proportion, p, of each source (Deer shown here), we fit a distribution for all wolves (`p.global.Deer`). Means for each Region (`p.Region 1-3.Deer`) are drawn from this `p.global.Deer` distribution, with deviation `Region.SD`. The process repeats for each subsequent 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`. Means for each Individual are drawn from their Pack distributions (`p.Ind 22-35.Deer` from `p.Pack 4.Deer`), with deviation `Individual.SD`.

3.1.8 Using MCMC diagnostics

When the model is finished, a bunch of plots will be spit out. First, though, 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 trace plots (see Figure 9) and diagnostic tests in this effort. MixSIAR prints the diagnostics in the R console (above Summary Statistics) and saves them as "diagnostics.txt". Check them out!

MixSIAR displays 3 diagnostic tests by default: Gelman-Rubin, Heidelberger-Welch, 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" [**?**]. The Heidelberger-Welch test consists of two parts: a stationarity test and a half-width test, and prints out "passed" or "failed" for each test for each variable for each chain. 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: see 5.1.

3.1.9 Interpreting MixSIAR output

After the model is finished, several plots are created:

- Posterior plot comparing the variation in diet by each factor (Figure 4, "posterior_density_SD.pdf")
- Posterior plots of diet by Pack (Figure 6, "posterior_density_diet_p_Pack 4.pdf" etc.)
- Posterior plots of diet by Region (Figure 5, "posterior_density_diet_p_Region 1.pdf" etc.)
- Posterior plot of overall population diet (Figure 7, "posterior_density_diet_p_global.pdf")
- Pairs plot (Figure 8, "pairs_plot.pdf")
- XY/Trace plots of overall population diet and variation by factor (Figure 9, "xy_plot_diet_p.pdf" and "xy_plot_SD.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**, **Pack.SD**, and **Individual.SD**—these are the variation in diet for Factor 1 (Region), Factor 2 (Pack), and Individual. You should have values close to $\hat{\sigma}_{region} = 1.33$, $\hat{\sigma}_{pack} = 0.53$, and $\hat{\sigma}_{ind} = 0.30$. Thus, we can say that most of the variation in wolves diets was driven by Region. Check that the mean, SD, and quantiles for **Region.SD**, **Pack.SD**, and **Individual.SD** match their posterior density distributions (posterior_density_SD.pdf, Figure 4).

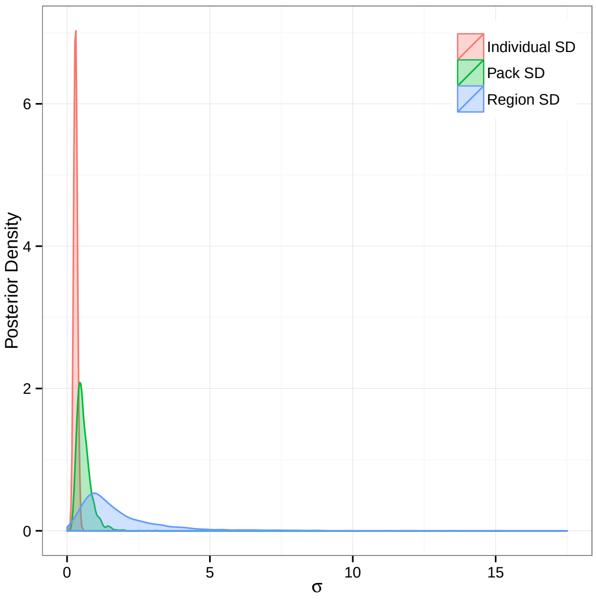


Figure 4: **Posterior plot comparing the variation in diet by each factor.** Including covariates as Random Effects in MixSIAR allows you to examine which explains more of the variation in consumer diet. In the Wolves Example here, $\hat{\sigma}_{region} = 1.33$, $\hat{\sigma}_{pack} = 0.53$, and $\hat{\sigma}_{ind} = 0.30$, indicating that the majority of the total variation in wolves' diets was driven by Region. Median values are printed in Summary Statistics; see 50% values for **Region.SD**, **Pack.SD**, and **Individual.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 (81.3% Deer, 7.8% Marine Mammals, and 9.0% Salmon), look in Summary Statistics for the 50% quantiles of **p.Region 1.Deer**, **p.Region 1.Marine Mammals**, and **p.Region 1.Salmon**. 95% credible intervals[4] 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**. Check that these values agree with the posterior plot of Region 1 wolves' diet (posterior_density_diet_p_Region_1.pdf, Figure 5).

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

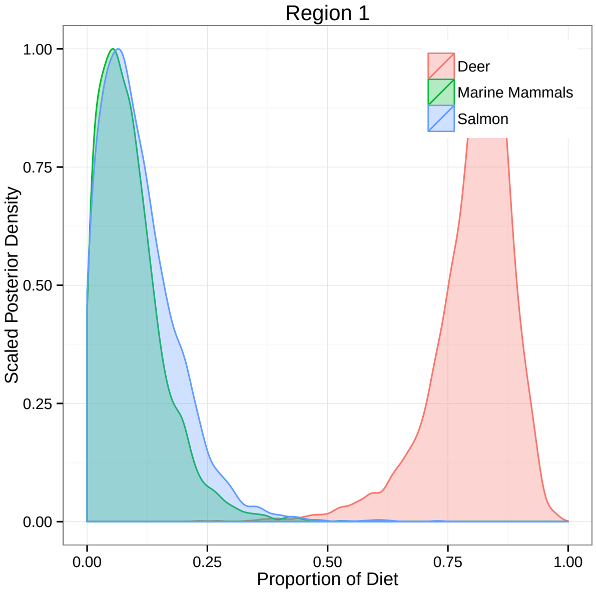


Figure 5: **Posterior plot of Region 1 wolves' diet.** MixSIAR plots the diet of each level of each categorical covariate. The diet proportions of Region 1 are in Summary Statistics as **p.Region 1.Deer**, **p.Region 1.Marine Mammals**, and **p.Region 1.Salmon**.

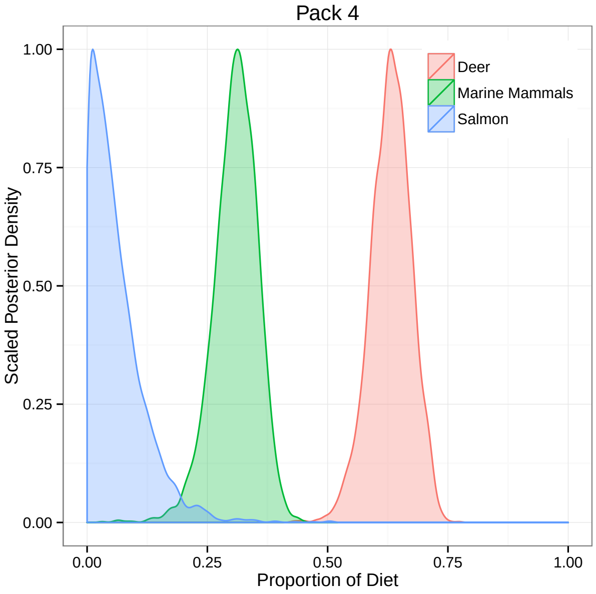


Figure 6: **Posterior plot of Pack 4 wolves' diet.** MixSIAR plots the diet of each level of each categorical covariate. The diet proportions of Pack 4 are in Summary Statistics as **p.Pack 4.Deer**, **p.Pack 4.Marine Mammals**, and **p.Pack 4.Salmon**.

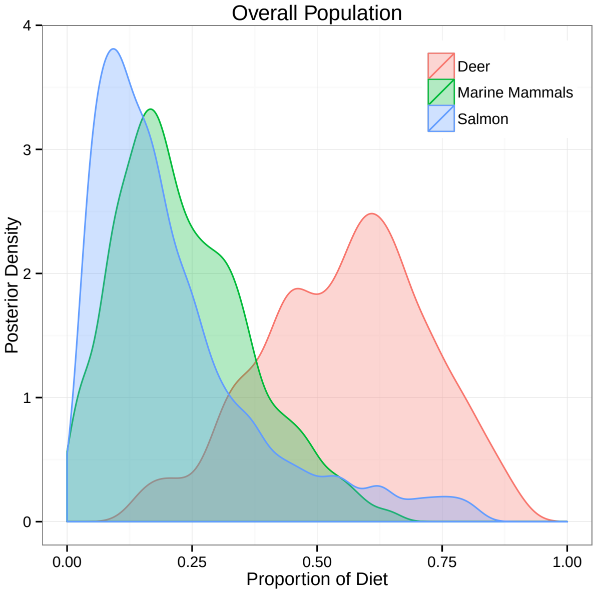


Figure 7: **Posterior plot of overall wolves' diet.** MixSIAR also plots the diet of the overall consumer population, **p.global**.

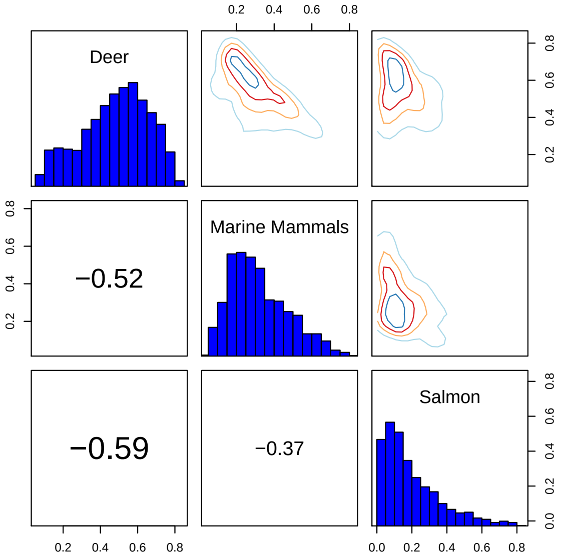


Figure 8: **Pairs plot of the posterior diet proportions of the overall wolf population.** The upper-diagonal shows contour plots, the diagonal shows histograms, and the lower-diagonal shows the correlations between the different sources.

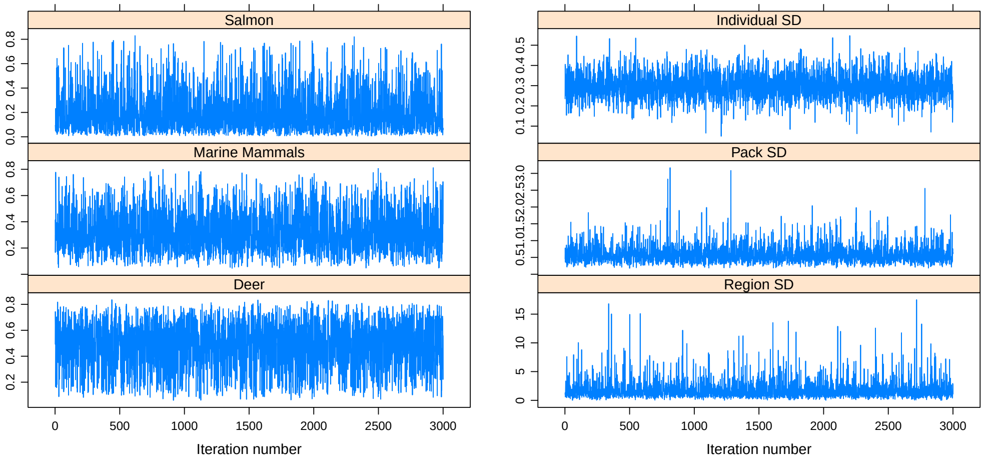


Figure 9: **XY/Trace plots of overall population diet (left) and variation by factor (right).** Trace plots help in assessing convergence of Markov chains—look for constant mean and variance, and the chain's ability to traverse its distribution quickly.

3.1.10 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".

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. `jags.1` is the `rjags` object and `jags1.mcmc` is the `mcmc.list` object. Objects holding MCMC chains you may be interested in are:

- `p.ind`: individual consumer diet proportions
- `p.fac1`: Region consumer diet proportions
- `p.fac2`: Pack consumer diet proportions
- `p.global`: overall consumer diet proportions
- `fac1.sig`: variation in diet among Regions
- `fac2.sig`: variation in diet among Packs
- `ind.sig`: variation in diet among Individuals

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 [Region, Source]. So p.fac1[1,1] is the proportion of Region 1 wolves' diet that was Deer, p.fac1[1,2] is the proportion of Region 1 wolves' diet that was Marine Mammals, etc). 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.

For example, to find the median diet proportions for Region 1 wolves (81.3% Deer, 7.8% Marine Mammals, and 9.0% Salmon), look in Summary Statistics for the 50% values of **p.fac1[1,1:3]**. 95% credible intervals 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.fac1[1,1:3]**.

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(jags.1)` 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 Lake 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 "Lake Example" data is simulated based on Francis et al [**?**] and looks at the diet of zooplankton in 21 lakes using:

- 2 Isotope values (δ^{13}C, δ^{15}N)
- 1 Continuous covariate (Secchi Depth : Mixed Layer Depth)
- Individual effect
- Raw source data

Fitting a model with a continuous covariate is more complex than categorical covariates and can be a bit finicky.

3.2.1 Loading mixture 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 a continuous effect (Secchi:Mixed).
4. Make sure the box at the bottom that says "Include 'Individual' as a random effect" is ticked, and then click "I'm finished". When analyzing a continuous effect, Individual

must be included in the model. Here, “Individuals” are zooplankton samples from 21 unique lakes in this example, not individual zooplankton.

3.2.2 Loading source data

1. Click “Load source data”
2. There is no concentration dependence data in “lake_sources.csv”, so “Do you have Concentration Dependence data?” should be “No”.
3. In this example we have the original “raw” data—Click “Load raw source data”.
4. Choose “lake_sources.csv” when prompted for the data file, and click “Ok”.
5. Click “I’m finished”.

3.2.3 Loading discrimination data

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

3.2.4 Making an isospace plot

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

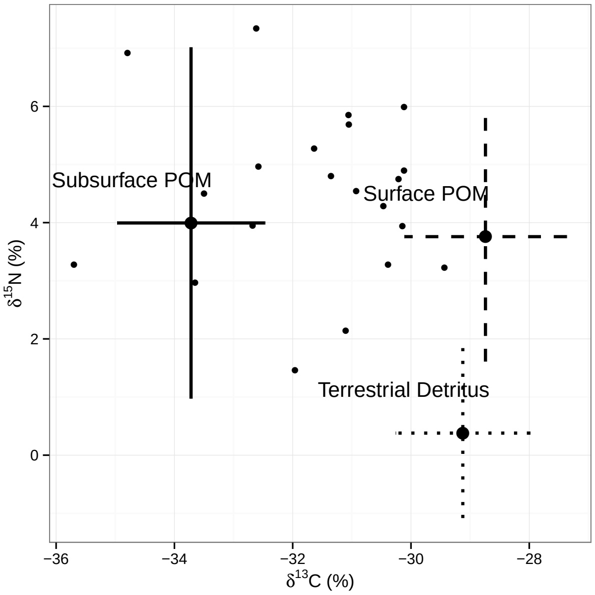


Figure 10: **Stable isotope input for the Lake Example.** Consumer data (zooplankton) are smaller black dots and source data are labeled. Error bars indicate combined source and discrimination uncertainty ± 1 SD.

3.2.5 Specify MCMC parameters

For the Lake Example you can set "# of Chains" = 3, "Chain Length" = 100000, "Burn-in" = 50000, and "Thin" = 50. For more description of these parameters, see Section 4.2.

3.2.6 Error term options

In the Lake Example we want both "residual error" and "process error", and they should both already be checked.

3.2.7 Run MixSIAR

Click the "RUN MODEL" button at the bottom. The Lake Example model took my computer about 10 minutes to run (Windows 8, Intel Core i5 1.7GHz with 8GB RAM, running other programs).

3.2.8 Using MCMC diagnostics

When the model is finished, use the trace plots (see Figure 16) and diagnostic tests to determine if the model has converged. See Section 3.1.8 if you need a refresher on how to do this.

3.2.9 Interpreting MixSIAR output

After the model is finished, the following plots should be produced:

- Posterior plot showing the variation in individual diet (Figure 14, "posterior_density_SD.pdf")
- Posterior plot of diet by Secchi Depth : Mixed Layer Depth (Figure 11, "posterior_density_diet_p_Cont1.pdf")
- Posterior plots of diet of the min(Secchi), median(Secchi), and max(Secchi) individuals (Figure 12, "posterior_density_diet_min_Cont1.pdf", "posterior_density_diet_median_Cont1.pdf", and "posterior_density_diet_max_Cont1.pdf")
- Posterior plot of overall population diet (Figure 13, "posterior_density_p_global.pdf")
- Pairs plot (Figure 15, "pairs_plot.pdf")
- XY/Trace plots of overall population diet and individual variation (Figure 16, "xy_plot_diet_p.pdf" and "xy_plot_SD.pdf)

Summary statistics are also printed out and saved, as in the Wolves Example.

There are two main ways we can determine if the continuous effect, Secchi Depth : Mixed Layer Depth, has a significant effect on diet:

1. Look at the posterior plot of diet changing with Secchi Depth : Mixed Layer Depth (Figure 11). If the continuous effect is significant, the curves in this plot will have non-zero slope.
2. Compare the DIC values of the models with and without the continuous effect (DIC is at the top of the Summary Statistics). Generally, models with smaller DIC are preferred over those with larger DIC. If removing the continuous effect causes the DIC to decrease more than one or two points, then the continuous effect probably should be included (and is significant). See Semmens et al [**?**] for an example of model selection in a hierarchical Bayesian context.

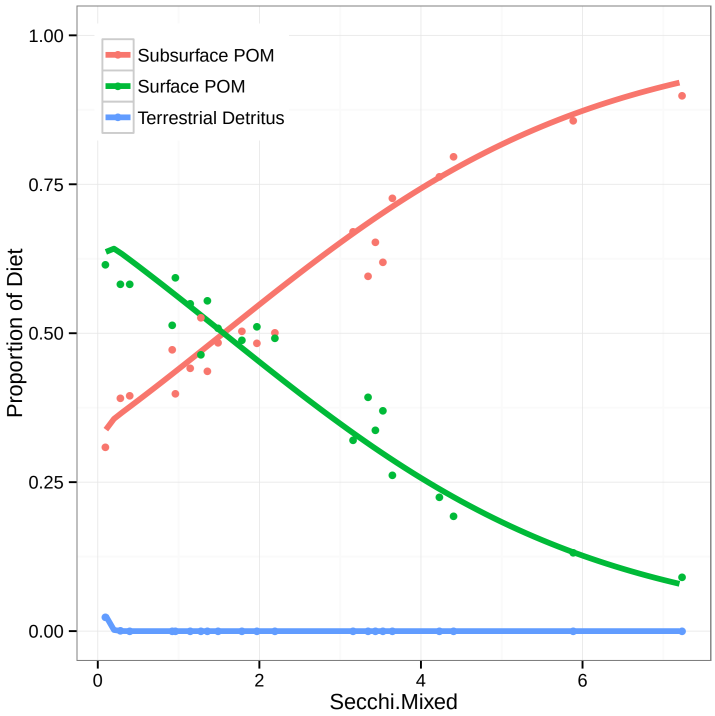


Figure 11: **Posterior plot showing how zooplankton diet changes with Secchi Depth : Mixed Layer Depth.** See that zooplankton diet at the lowest Secchi:Mixed values is roughly 65% Surface POM, 33% Subsurface POM, and 2% Terrestrial Detritus, while zooplankton diet at the highest Secchi:Mixed values is 8% Surface POM, 92% Subsurface POM, and 0% Terrestrial Detritus. Points represent median individual consumer (lake zooplankton) diet proportions by source (color), and lines represent regressions on these individual consumer diet proportions. For more explanation, see [**?**] and [**?**].

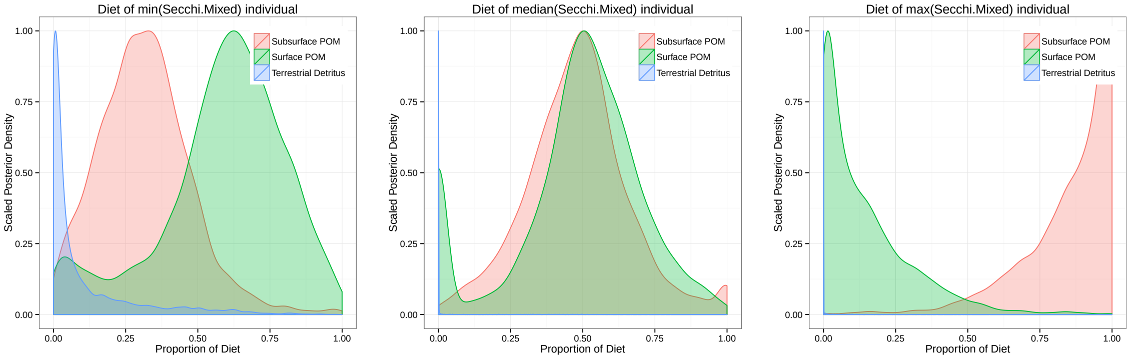


Figure 12: **Posterior plots showing the shift in individual lake zooplankton diet.** From L to R: Diet of zooplankton from the lowest Secchi:Mixed lake, median Secchi:Mixed lake, and max Secchi:Mixed lake.

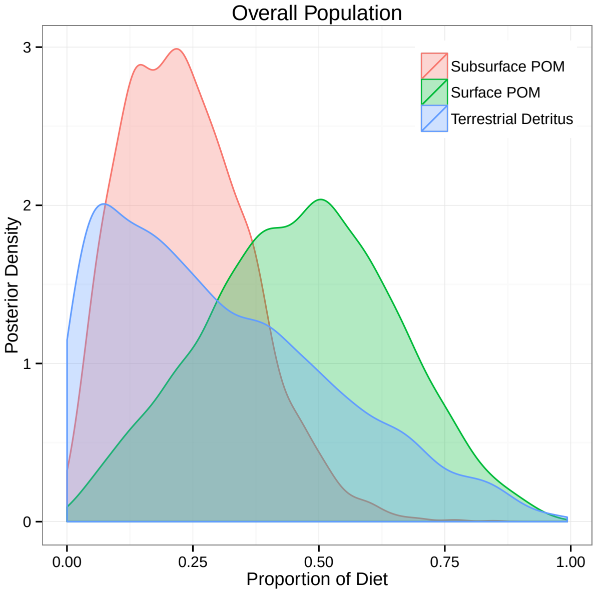


Figure 13: **Posterior plot of overall zooplankton diet.**

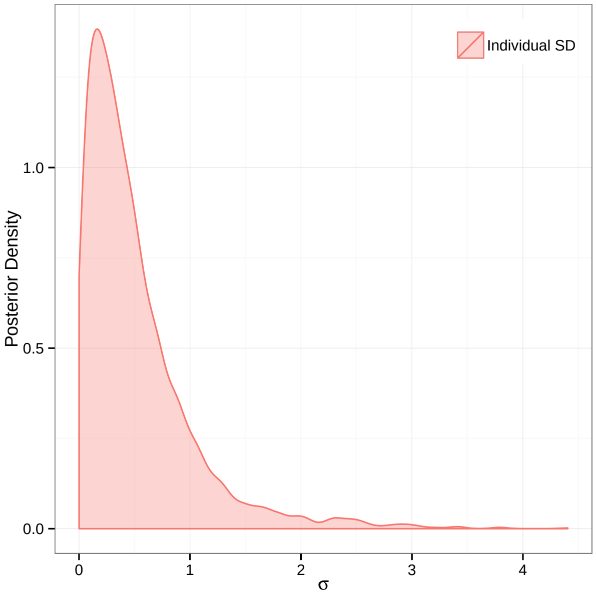


Figure 14: **Posterior plot of the variation in individual diet.** This plot is more useful and interesting if you have categorical covariates, where you would compare the mean of the SD of each.

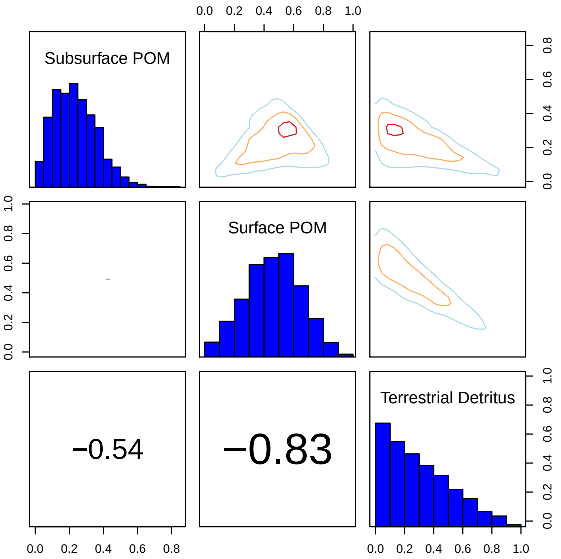


Figure 15: **Pairs plot of the posterior diet proportions of the overall zooplankton population.** The upper-diagonal shows contour plots, the diagonal shows histograms, and the lower-diagonal shows the correlations between the different sources.

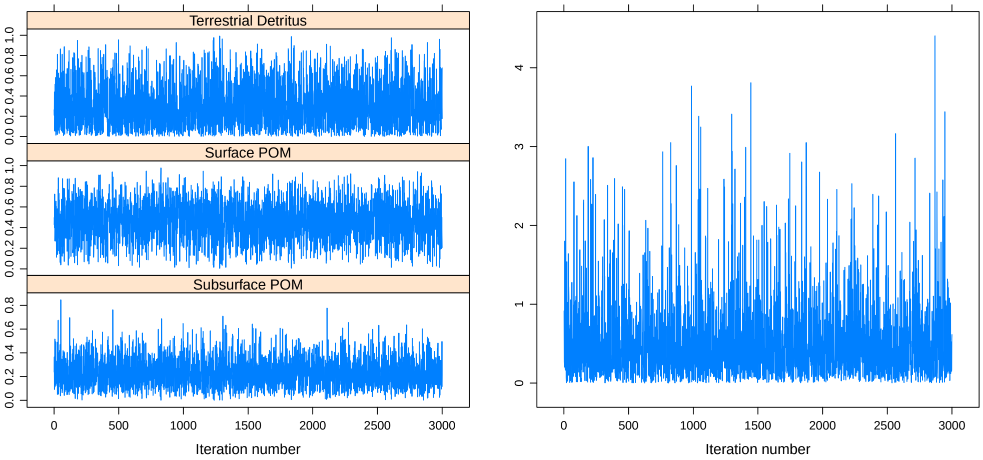


Figure 16: **XY/Trace plots of overall population diet (left) and individual variation (right).** Trace plots help in assessing convergence of Markov chains—look for constant mean and variance, and the chain's ability to traverse its distribution quickly.

3.3 Palmyra Example

For this walk-through I assume you've already done the other two examples (Wolves and Lake). The "Palmyra Example" data is actual data from McCauley et al [**?**] and analyzes the diet of large reef predators around the Palmyra Atoll with:

- 2 Isotope values (δ^{13}C, δ^{15}N)
- 1 Categorical covariate (Taxa)
- Individual effect
- Raw source data

3.3.1 Loading mixture 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 categorical covariate (Taxa).
3. Select which columns are isotopes (d13C and d15N) and which is a random effect (Taxa).
4. Make sure the box at the bottom that says "Include 'Individual' as a random effect" is ticked, and then click "I'm finished".

3.3.2 Loading source data

1. Click "Load source data"
2. There is no concentration dependence data in "palmyra_sources.csv", so "Do you have Concentration Dependence data?" should be "No".
3. In this example we have the original "raw" data—Click "Load raw source data".
4. Choose "palmyra_sources.csv" when prompted for the data file, and click "Ok".
5. Click "I'm finished".

3.3.3 Loading discrimination data

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

3.3.4 Making an isospace plot

Click "Make isospace plot". Your plot should match that of Figure 17.

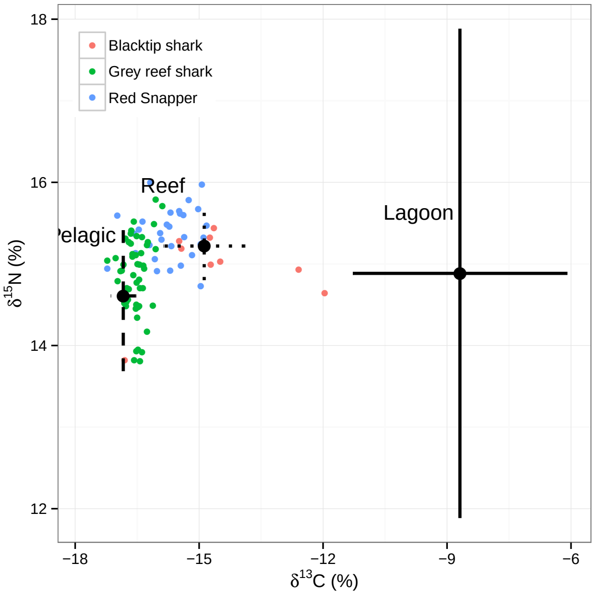


Figure 17: **Stable isotope input for the Palmyra Example.** Consumer data are by Taxa (color) and source data are labeled. Error bars indicate combined source and discrimination uncertainty ± 1 SD.

3.3.5 Specify MCMC parameters

For the Palmyra Example you can set "# of Chains" = 3, "Chain Length" = 50000, "Burn-in" = 25000, and "Thin" = 25. For more description of these parameters, see Section 4.2.

3.3.6 Error term options

In the Palmyra Example we want both "residual error" and "process error", and they should both already be checked.

3.3.7 Run MixSIAR

Click the "RUN MODEL" button at the bottom. The Palmyra Example model took my computer about 10 minutes to run (Windows 8, Intel Core i5 1.7GHz with 8GB RAM, running other programs).

3.3.8 Using MCMC diagnostics

When the model is finished, use the trace plots (see Figure 22) and diagnostic tests to determine if the model has converged. See Section 3.1.8 if you need a refresher on how to do this.

3.3.9 Interpreting MixSIAR output

After the model is finished, the following plots should be created:

- Posterior plot comparing the variation in diet by each factor (Figure 18, "posterior_density_SD.pdf")
- Posterior plots of diet by Taxa (Figure 19, "posterior_density_diet_p_Grey reef shark.pdf", etc.)
- Posterior plot of overall population diet (Figure 20, "posterior_density_diet_p_global.pdf")
- Pairs plot (Figure 21, "pairs_plot.pdf")
- XY/Trace plots of overall population diet and variation by factor (Figure 22, "xy_plot_diet_p.pdf" and "xy_plot_SD.pdf)

As in the other working examples, MixSIAR also prints and saves summary statistics. As an example, to find the median diet proportions for Grey Reef Shark (0.1% Lagoon, 84.5% Pelagic, and 15.1% Reef), look in Summary Statistics for the 50% values of **p.Grey reef shark.Lagoon**, **p.Grey reef shark.Pelagic**, and **p.Grey reef shark.Reef**. 95% credible intervals for Lagoon, Pelagic, and Reef habitat contribution to Grey Reef Shark diet would be found in Summary Statistics, 2.5% and 97.5% of the same variables. See that these values agree with the posterior plot of Grey Reef Shark diet in Figure 19.

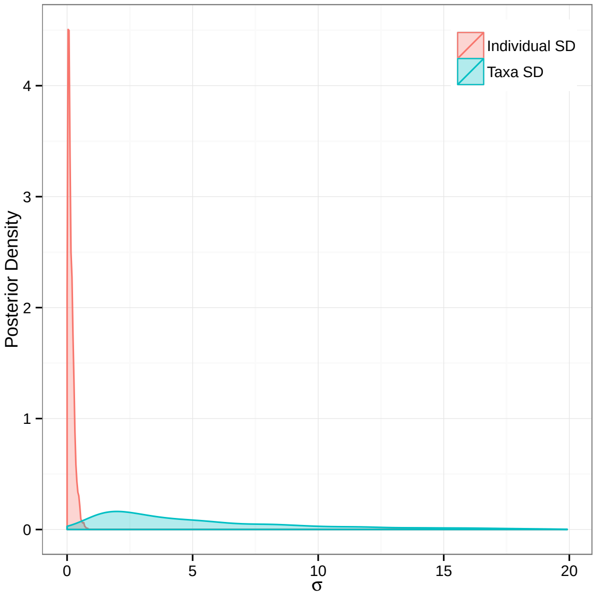


Figure 18: **Posterior plot comparing the variation in diet by each factor.** In the Palmyra Example, we can compare the variation in diet between species and individuals. $\hat{\sigma}_{taxa} = 3.59$ and $\hat{\sigma}_{ind} = 0.13$, indicating that the vast majority of the total variation in consumer diet was driven by Taxa. Median values are printed in Summary Statistics; see 50% values for **Taxa.SD**, and **Individual.SD**.

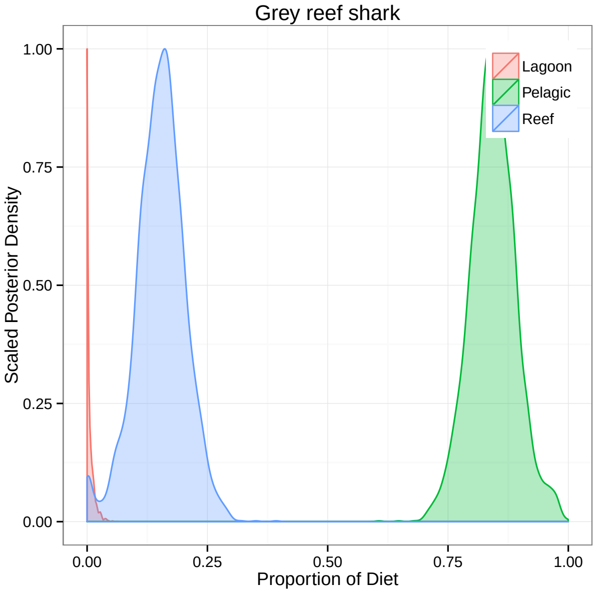


Figure 19: **Posterior plot of Grey Reef Shark diet.** Median values of **p.Grey reef shark.Lagoon**, **p.Grey reef shark.Pelagic**, and **p.Grey reef shark.Reef** indicate that Grey Reef Shark diet is composed of 0.1% Lagoon, 84.5% Pelagic, and 15.1% Reef sources.

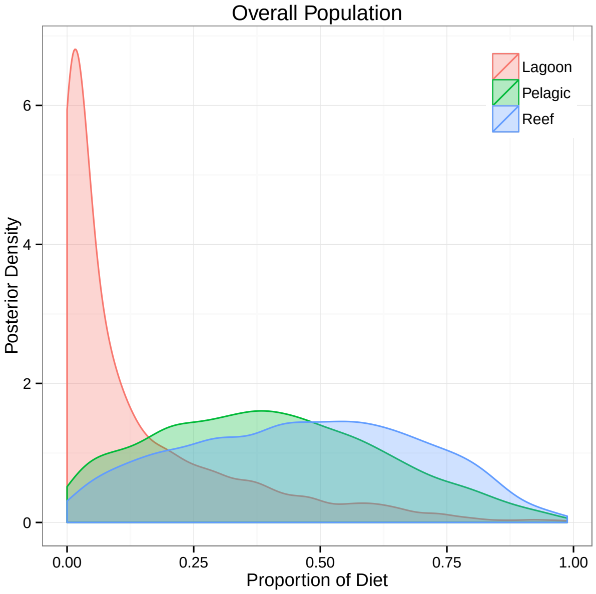


Figure 20: **Posterior plot of overall Palmyra consumer diet.** MixSIAR also plots the diet of the overall consumer population, **p.global**.

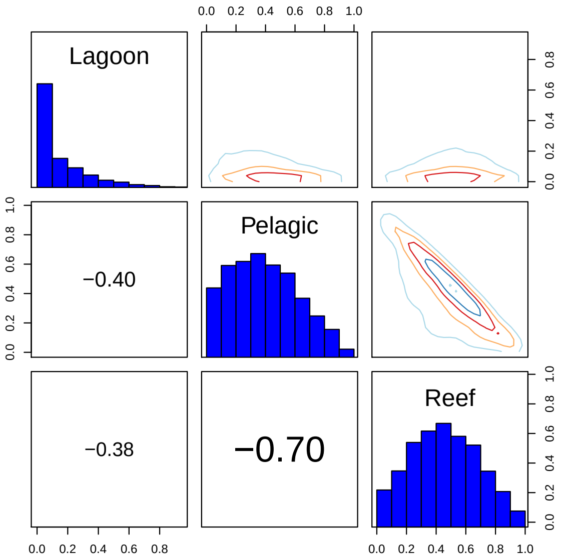


Figure 21: **Pairs plot of the posterior diet proportions of the overall Palmyra consumer population.** The upper-diagonal shows contour plots, the diagonal shows histograms, and the lower-diagonal shows the correlations between the different sources.

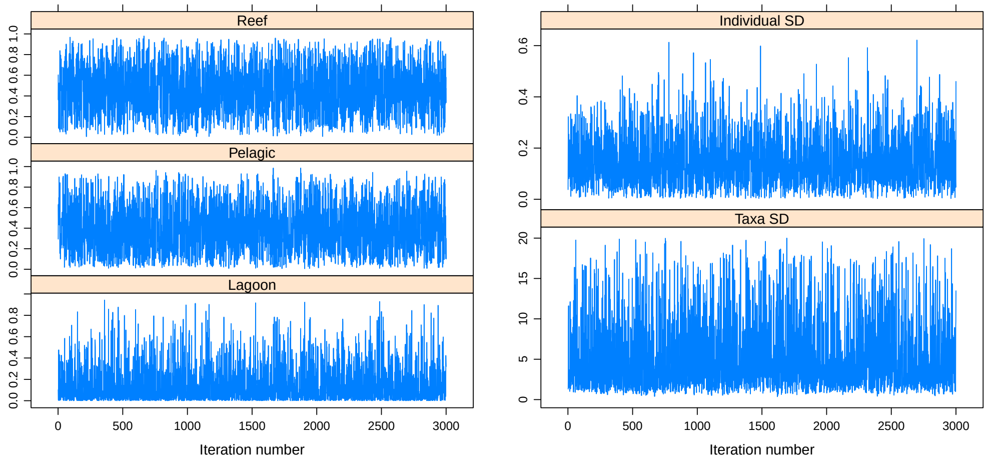


Figure 22: **XY/Trace plots of overall population diet (left) and variation by factor (right).** Trace plots help in assessing convergence of Markov chains—look for constant mean and variance, and the chain's ability to traverse its distribution quickly.

3.4 Geese Example

The "Geese Example" uses data from Inger et al [?] 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 [?]:

- 2 Isotope values (δ^{13}C, δ^{15}N)
- 1 Categorical covariate (Group)
- Individual effect
- Source data as means and SDs
- Concentration dependence

3.4.1 Loading mixture 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 random effect (Group).
4. Make sure the box at the bottom that says "Include 'Individual' as a random effect" is ticked, and then click "I'm finished".

3.4.2 Loading 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 and SDs, 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".

3.4.3 Loading discrimination data

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

3.4.4 Making an isospace plot

Click “Make isospace plot”. Your plot should match that of Figure 23.

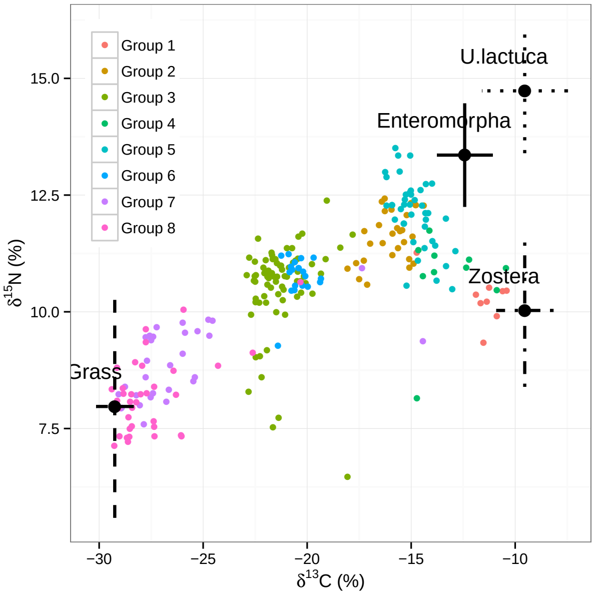


Figure 23: **Stable isotope input for the Geese Example.** Consumer data are by Group (color) and source data are labeled. Error bars indicate combined source and discrimination uncertainty ± 1 SD.

3.4.5 Specify MCMC parameters

For the Geese Example you can set “# of Chains” = 3, “Chain Length” = 50000, “Burn-in” = 25000, and “Thin” = 25. For more description of these parameters, see Section 4.2.

3.4.6 Error term options

In the Geese Example we want both “residual error” and “process error”, and they should both already be checked.

3.4.7 Run MixSIAR

Click the “RUN MODEL” button at the bottom. The Geese Example is larger than the others, so it will take significantly longer to run (about 40 minutes on my laptop running Windows 8, Intel Core i5 1.7GHz with 8GB RAM).

3.4.8 Using MCMC diagnostics

When the model is finished, use the trace plots (see Figure 28) and diagnostic tests to determine if the model has converged. See Section 3.1.8 if you need a refresher on how to do this.

3.4.9 Interpreting MixSIAR output

After the model is finished, the following plots should be created:

- Posterior plot comparing the variation in diet by each factor (Figure 24, "posterior_density_SD.pdf")
- Posterior plots of diet by Group (Figure 25, "posterior_density_diet_p_Group_6.pdf", etc.)
- Posterior plot of overall population diet (Figure 26, "posterior_density_diet_p_global.pdf")
- Pairs plot (Figure 27, "pairs_plot.pdf")
- XY/Trace plots of overall population diet and variation by factor (Figure 28, "xy_plot_diet_p.pdf" and "xy_plot_SD.pdf)

As in the other working examples, MixSIAR also prints and saves summary statistics. As an example, to find the median diet proportions for Group 6 geese (0.1% , 84.5% , and 15.1%), look in Summary Statistics for the 50% values of **p.Group 6.Grass**, **p.Group 6.Enteromorpha**, **p.Group 6.U.lactuca**, and **p.Group 6.Zostera**. The 95% credible interval for Grass contribution to Group 6 geese diet would be found in Summary Statistics, 2.5% and 97.5% of **p.Group 6.Grass**. See that these values agree with the posterior plot of Group 6 diet in Figure 25.

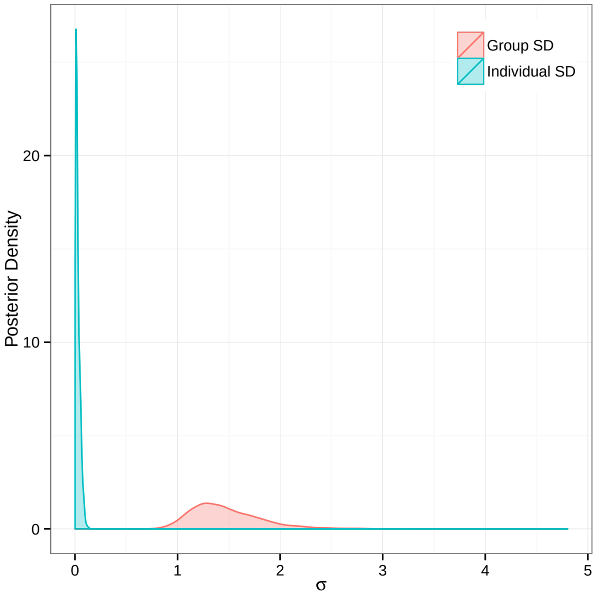


Figure 24: **Posterior plot comparing the variation in diet by Group and Individual.** In the Geese Example, we can compare the variation in diet between Group and individuals, $\hat{\sigma}_{group} = 1.39$ and $\hat{\sigma}_{ind} = 0.03$. Because $\hat{\sigma}_{ind}$ is so small compared to $\hat{\sigma}_{group}$, we may want to remove Individual from the model. Median values are printed in Summary Statistics; see 50% values for **Group.SD** and **Individual.SD**, if you included Individual Effects in the model).

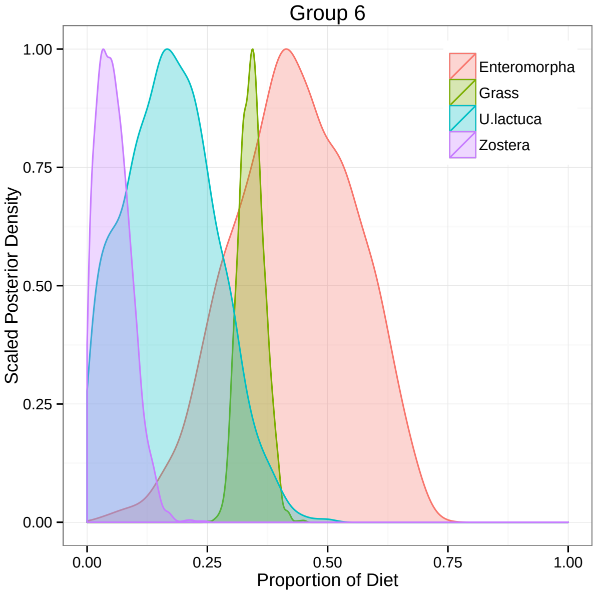


Figure 25: **Posterior plot of Group 6 geese diet.** For medians and credible interval limits, look in the Summary Statistics at **p.Group 6.Grass**, **p.Group 6.Enteromorpha**, **p.Group 6.U.lactuca**, and **p.Group 6.Zostera**.

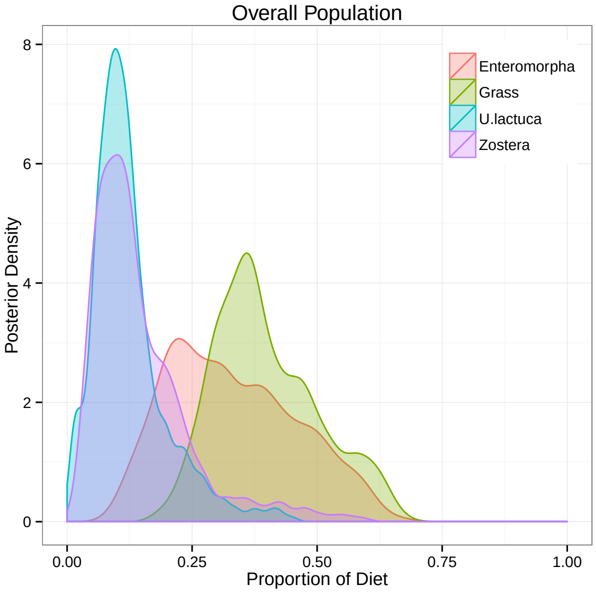


Figure 26: **Posterior plot of overall geese diet.** MixSIAR also plots the diet of the overall consumer population, **p.global**.

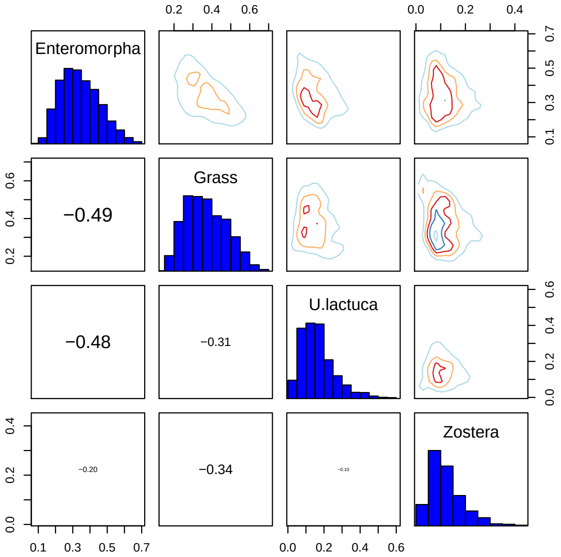


Figure 27: **Pairs plot of the posterior diet proportions of the overall goose population.** The upper-diagonal shows contour plots, the diagonal shows histograms, and the lower-diagonal shows the correlations between the different sources.

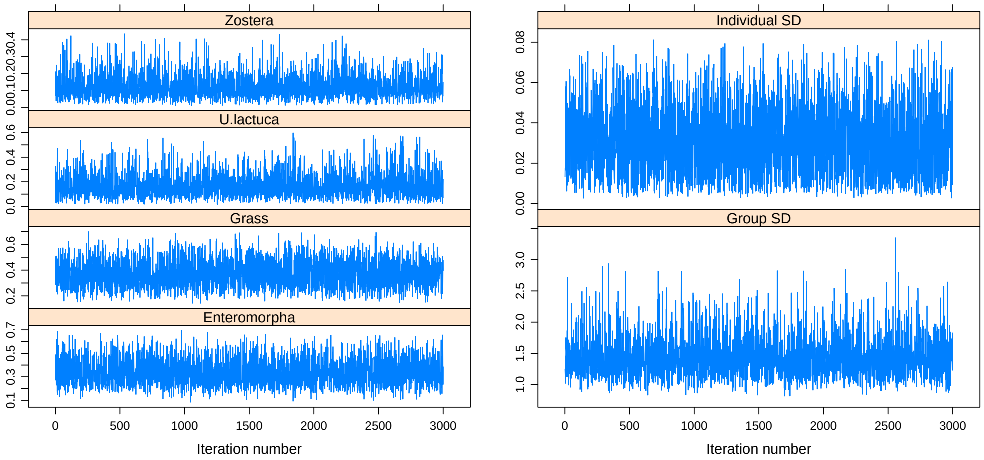


Figure 28: **XY/Trace plots of overall population diet (left) and variation by factor (right).** Trace plots help in assessing convergence of Markov chains—look for constant mean and variance, and the chain's ability to traverse its distribution quickly.

4 Using MixSIAR GUI on your own data

After you've run the working examples, you should be confident the MixSIAR GUI is installed and working correctly, and now you can begin using it to analyze your own stable isotope data. Below, we discuss some details of the way the GUI is coded that will hopefully allow you to avoid some potential pitfalls. If the GUI still isn't working for you, either try modifying the code to fit your needs or contact the authors at `bcstock@ucsd.edu` or `semmens@ucsd.edu`. We would love to hear feedback, positive or negative, if you use the MixSIAR GUI. Drawing our attention to errors helps us fix them, and hearing how people are using the GUI can help us improve the model in the future.

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

`B. C. Stock and B. X. Semmens (2013). MixSIAR GUI User Manual, version 1.0. http://conserver.iugo-cafe.org/user/brice.semmens/MixSIAR.`

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

1. Check to see that all of the necessary MixSIAR files (#4 on Page 4) and your data files are in your working directory. See Data file format and loading.

2. Run `source("mixsiar.r")` and `mixsiar()` in R to create the MixSIAR GUI.

3. Load your files using the "Read-in data" buttons.

4. Choose your MCMC parameters ("# of Chains", "Chain Length", "Burn-in", and "Thin"). Section 4.2 is a basic introduction to MCMC.

5. Plot your isotope data with the "Make isospace plot" button (Making an isospace plot).

6. Choose your output options for summary statistics, plots, and diagnostics.

7. Click "RUN MODEL."

8. Check that the model has converged with diagnostics (Using MCMC diagnostics).

9. Look at your results—posterior density plots and summary statistics (Interpreting MixSIAR output).

4.1 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 isotope 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 isotope 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

   ⋮

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 isotope 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
⋮		
Subsurface POM	-35.225	-0.736
Subsurface POM	-34.928	-1.289
⋮		
Terrestrial Detritus	-31.933	-2.495
Terrestrial Detritus	-29.082	-2.504
⋮		

- *"Source means+SDs"—summary statistics.* 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'). 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. 100000) 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 without a covariate ("geese_sources.csv"):

Sources	Meand15N	SDd15N	Meand13C	SDd13C	Concd15N	Concd13C	n
Zostera	6.48898	1.45946	-11.1702	1.2149	0.0297	0.3593	14
Grass	4.43216	2.26807	-30.8798	0.6413	0.0355	0.4026	14
U.lactuca	11.1926	1.11243	-11.1709	1.9593	0.0192	0.2098	14
Enteromorpha	9.81627	0.82710	-14.0570	1.1724	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 [**?**] (e.g. Digest $[C]$, Digest $[N]$). Concentration dependence is built into the MixSIAR model following Equation 1 in Parnell et al [**?**]. 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

⋮

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”:

	Meand13C	SDd13C	Meand15N	SDd15N
Deer	2.1	0	2.6	0
Salmon	2.1	0	2.6	0
Marine Mammals	2.1	0	2.6	0

4.2 MCMC Parameters

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

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 Introductions to Bayesian Models, MCMC, and Gibbs Sampling). 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 Using MCMC diagnostics), 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.

You can pretty much always leave “# of Chains” at 3. “Chain Length” will depend on the size (# data points) and complexity (# of covariates, # of isotopes) of your model and its coherence with your data. I advise using a short “Chain Length” first, just to see that the model is working. Then if the diagnostics show that the chains have not converged, increase the chain length. “Burn-in” is the first section of the chain that we throw away, because it can be heavily influenced by initial values and not representative of the true posterior distribution. Conservatively you can set “Burn-in” at about 1/2 of “Chain Length”. Finally, we “Thin” the chains to reduce auto-correlation and save memory (thinning by 25 means we use every 25^{th} value in the chain). “Thin” so that the total # of saved draws is about 1,000 [?].

$$draws = \frac{length - burnin}{thin}$$

4.3 Error Term Options

The original MixSIR model did not have a residual error term [?], and the variance of the jth isotope mixture was *given* as a combination of the source and discrimination variances:

$$\sigma_j^2 = \sum_{i=1}^{n} \left[f_i^2 * \left(s^2_{j_{source_i}} + s^2_{j_{discr_i}} \right) \right] \tag{1}$$

This is what we mean by *process error*, and this term is included when the “Include process error” checkbox is checked. Process error is the propagation of the estimated uncertainty in the source and discrimination values.

Parnell et al. proposed replacing this process error term with a *residual error* term to account for unknown sources of error, apart from the uncertainties in the source and discrimination values [?]. In the Parnell et al. formulation, the variance of the jth isotope mixture, $\sigma^2_{resid_j}$, is an additional estimated parameter in the model. Then the residual error of the observed isotope value j of consumer i is:

$$\epsilon_{ij} \sim N\left(0, \sigma^2_{resid_j}\right) \tag{2}$$

We call this *residual error*, and this term is included when the “Include residual error” checkbox is checked.

The default in MixSIAR is to include *both* process and residual error, which accounts for the estimated uncertainty in source and discrimination values (process error) and unknown sources of error (residual error). We add an error term, $\sigma^2_{resid_j}$, to Equation 1, which will be estimated by our model and given a inverse gamma prior: $\frac{1}{\sigma^2_{resid_j}} \sim gamma(.001, .001)$:

$$\sigma_j^2 = \sum_{i=1}^{n} \left[f_i^2 * \left(s^2_{j_{source_i}} + s^2_{j_{discr_i}} \right) \right] + \sigma^2_{resid_j} \tag{3}$$

When both the “Include residual error” and “Include process error” boxes are checked, the model uses Equation 3. For a more detailed explanation of the residual error term, see Moore and Semmens [?], Jackson et al. [?], Semmens et al. [?], and Parnell et al. [?]

5 Introductions to Bayesian Models, MCMC, and Gibbs Sampling

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's 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] T. B. Francis, D. E. Schindler, G. W. Holtgrieve, E. R. Larson, M. D. Scheuerell, B. X. Semmens, and E. J. Ward. Habitat structure determines resource use by zooplankton in temperate lakes. *Ecology letters*, 14(4):364–372, 2011.

[2] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. CRC press, 2003.

[3] R. Inger, G. D. Ruxton, J. Newton, K. Colhoun, J. A. Robinson, A. L. Jackson, and S. Bearhop. Temporal and intrapopulation variation in prey choice of wintering geese determined by stable isotope analysis. *Journal of Animal Ecology*, 75(5):1190–1200, 2006.

[4] A. L. Jackson, R. Inger, S. Bearhop, and A. Parnell. Erroneous behaviour of MixSIR, a recently published Bayesian isotope mixing model: a discussion of Moore & Semmens (2008). *Ecology Letters*, 12(3):E1–E5, 2009.

[5] P. L. Koch and D. L. Phillips. Incorporating concentration dependence in stable isotope mixing models: a reply to Robbins, Hilderbrand and Farley (2002). *Oecologia*, 133(1):14–18, 2002.

[6] D. J. McCauley, H. S. Young, R. B. Dunbar, J. A. Estes, B. X. Semmens, and F. Micheli. Assessing the effects of large mobile predators on ecosystem connectivity. *Ecological Applications*, 22(6):1711–1717, 2012.

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

[8] A. C. Parnell, R. Inger, S. Bearhop, and A. L. Jackson. SIAR: Stable isotope analysis in R. *The Comprehensive R Archive Network (Available at http://cran.r-project.org/web/packages/siar/index.html [Verified 15 July 2012])*, 2008.

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

[10] A. C. Parnell, D. L. Phillips, S. Bearhop, A. L. Jackson, B. X. Semmens, E. J. Ward, J. W. Moore, and R. L. Inger. Bayesian stable isotope mixing models. *Environmetrics*, doi: 10.1002/env.2221, 2013.

[11] B. X. Semmens, J. W. Moore, and E. J. Ward. Improving Bayesian isotope mixing models: a response to Jackson et al.(2009). *Ecology Letters*, 12(3):E6–E8, 2009.

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

[13] E. J. Ward and B. X. Semmens D. E. Schindler. Including source uncertainty and prior information in the analysis of stable isotope mixing models. *Environmental Science & Technology*, 44(12):4645–4650, 2010.