

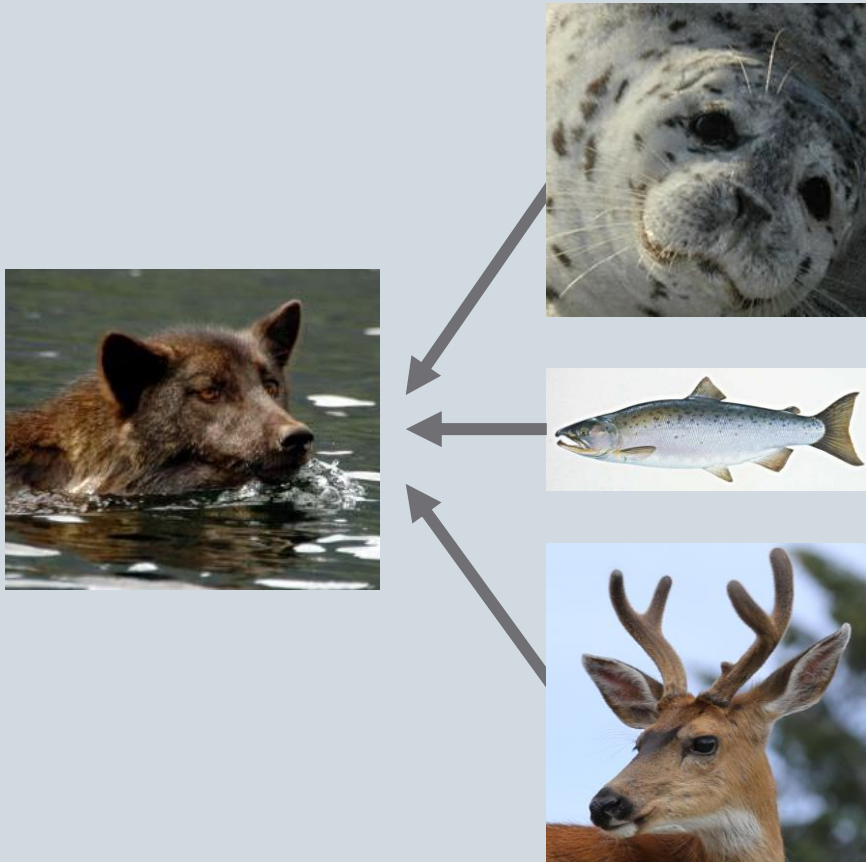
# Bayesian stable isotope mixing models and the MixSIAR R package

---

BRIAN STOCK

APRIL 3, 2023

# Calculate source % to a mixture



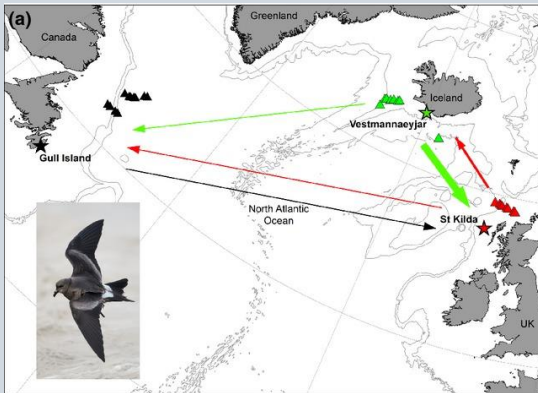
# Calculate **prey** % to a **diet**



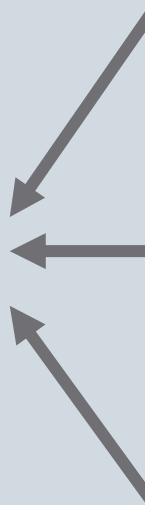
MixSIAR wolves example

[http://brianstock.github.io/MixSIAR/articles/wolves\\_ex.html](http://brianstock.github.io/MixSIAR/articles/wolves_ex.html)

# Calculate colony % to a bird



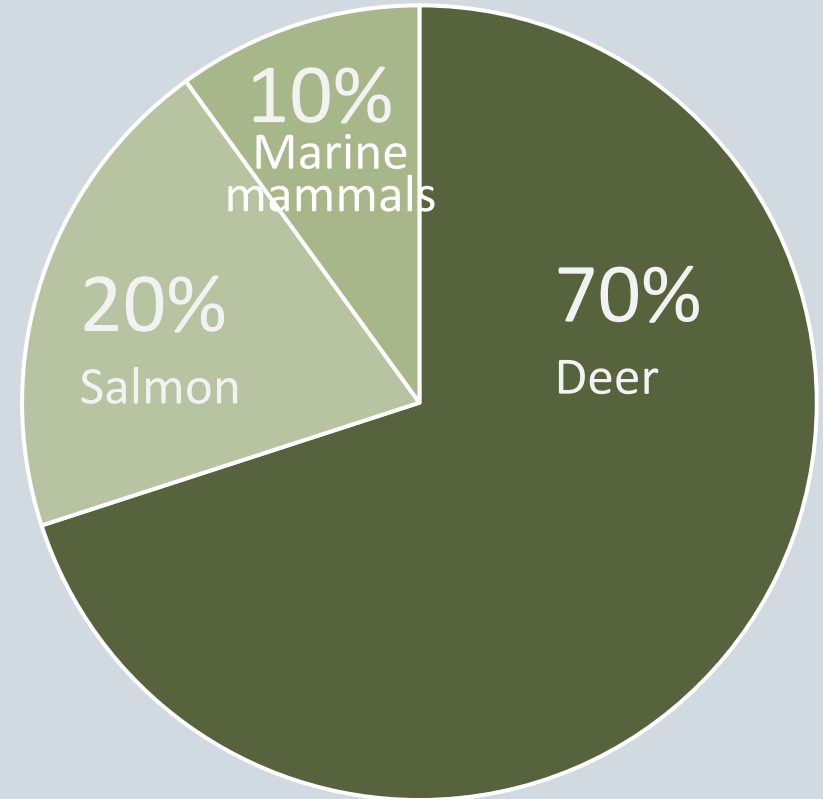
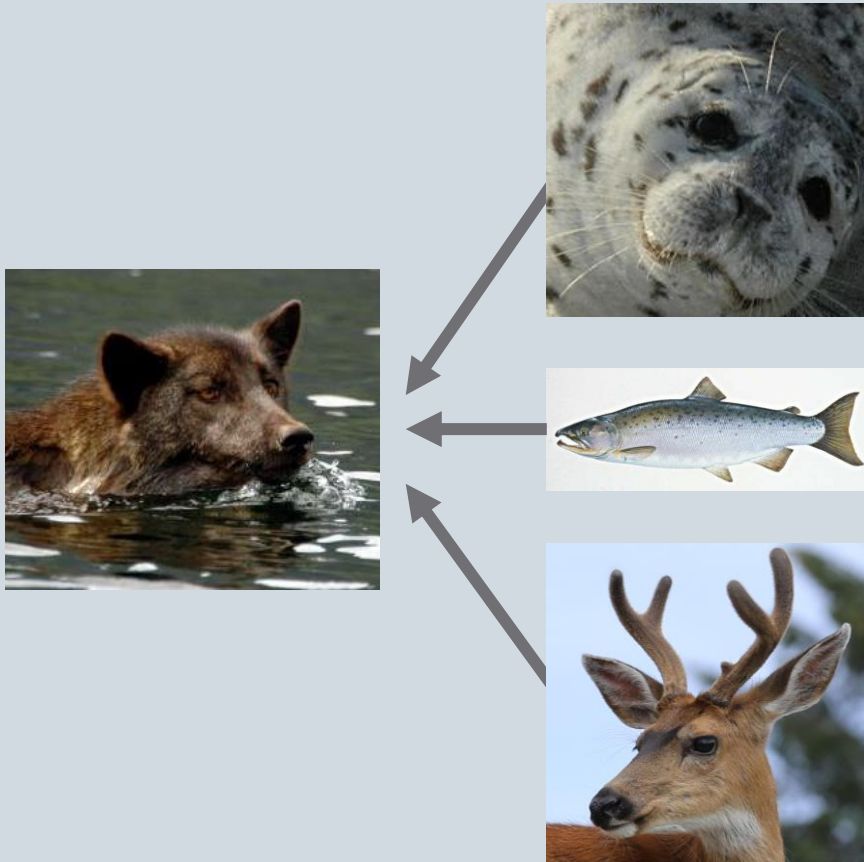
# Calculate **soil** % to a **sediment**



# Calculate **prey** % to a **diet**

Goal:

estimate source proportions,  $p$

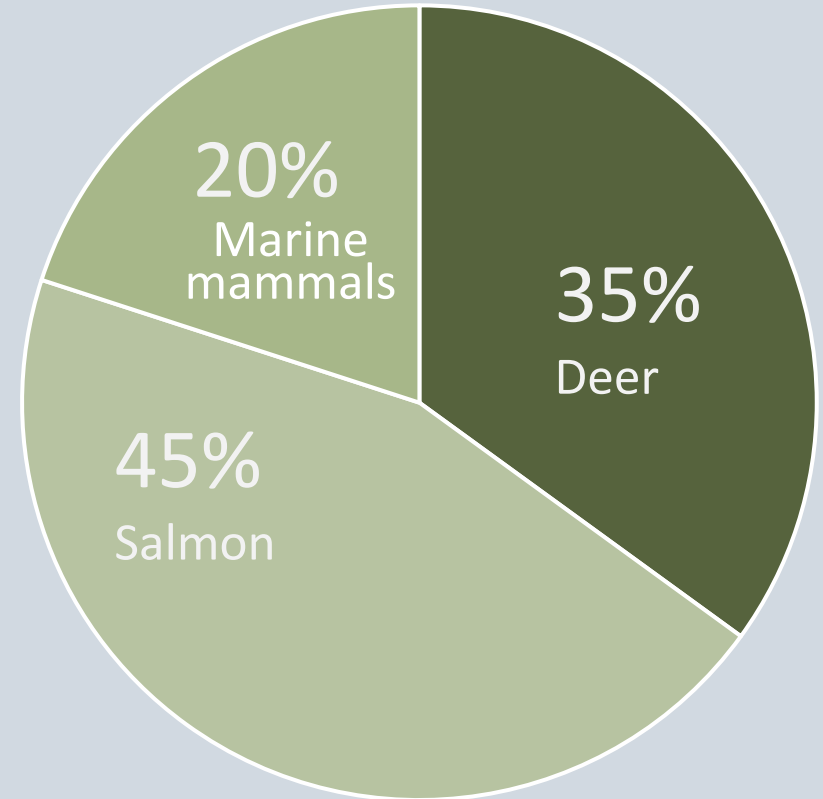
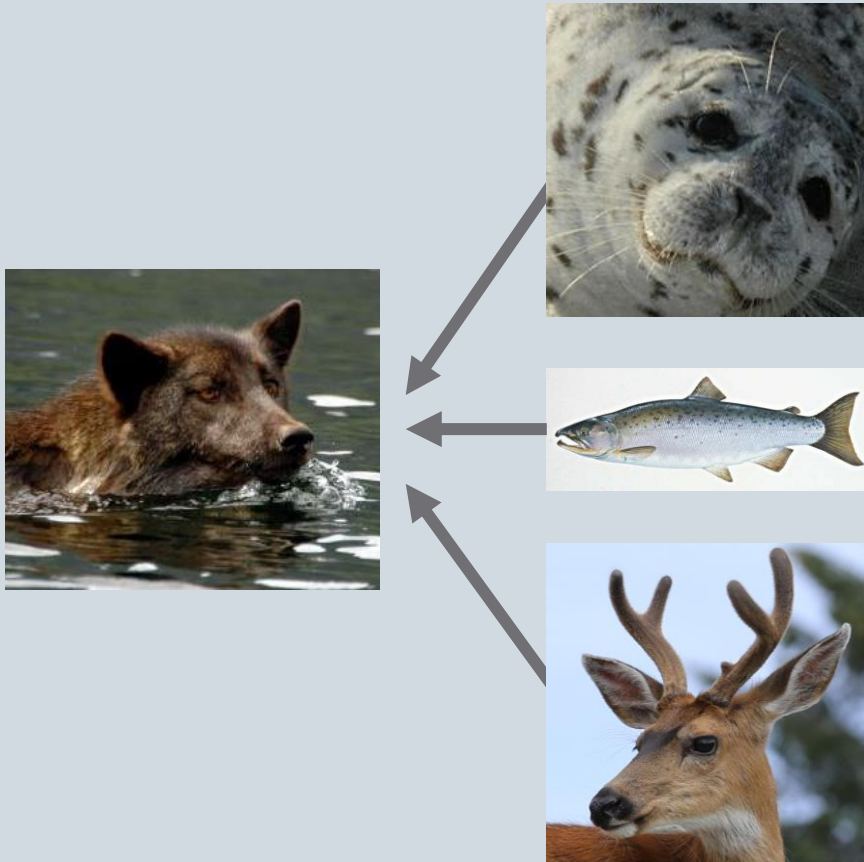




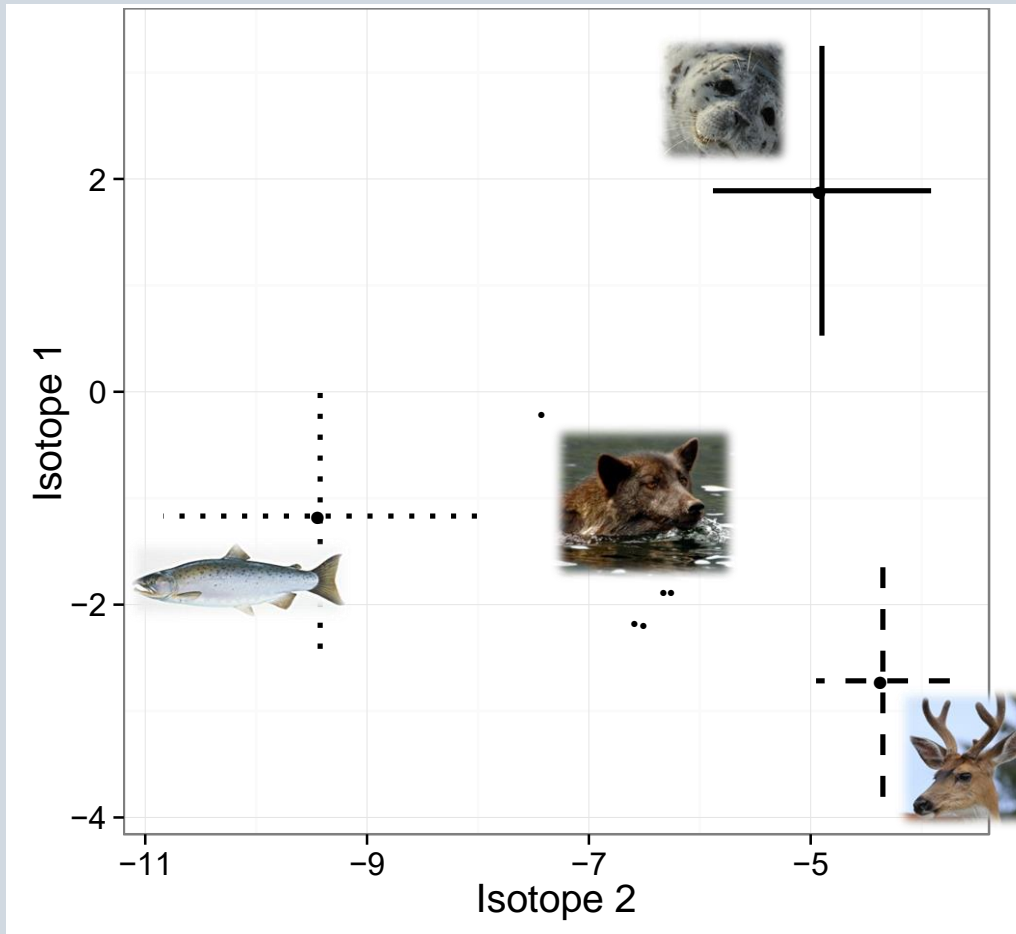
# Calculate **prey** % to a **diet**

Goal:

estimate source proportions,  $p$

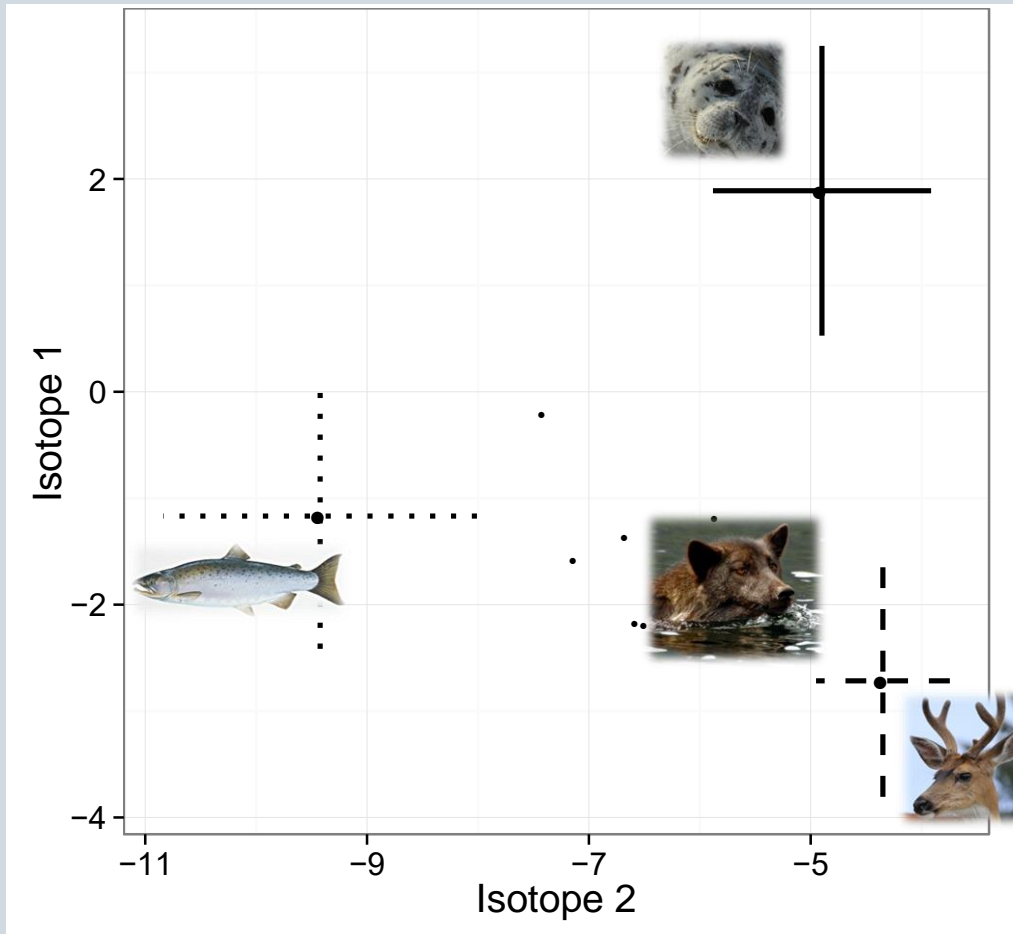


# Using stable isotope data “biotracer”



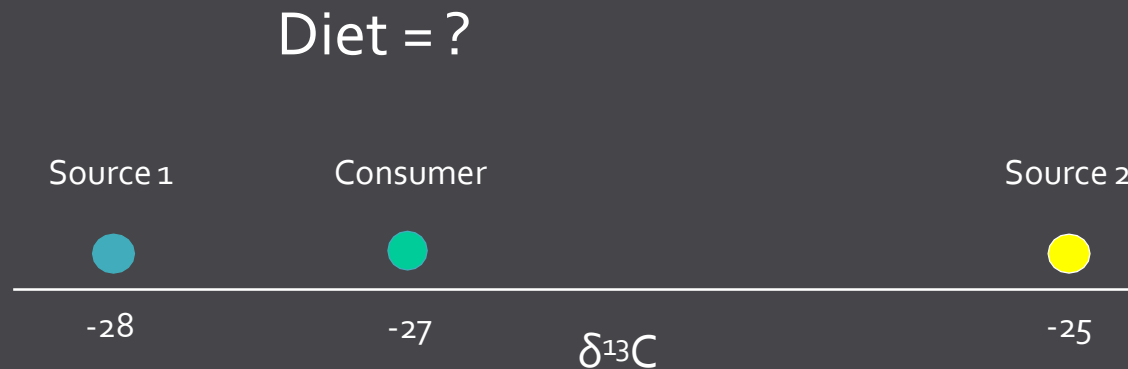


# Using stable isotope data “biotracer”



# How mixing models work

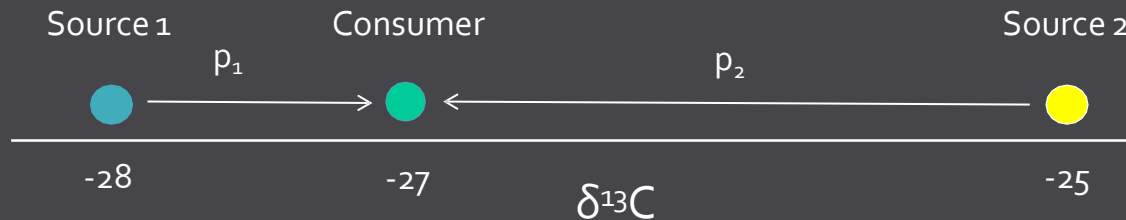
Linear mixing model:



# How mixing models work

Linear mixing model:

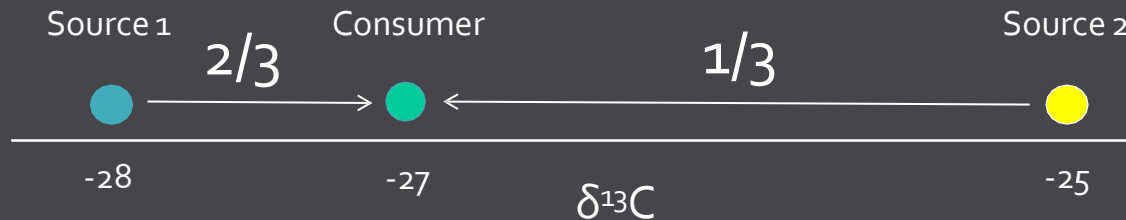
$$\text{Consumer} = p_1 * s_1 + p_2 * s_2 \quad (p_1 + p_2 = 1)$$



# How mixing models work

Linear mixing model:

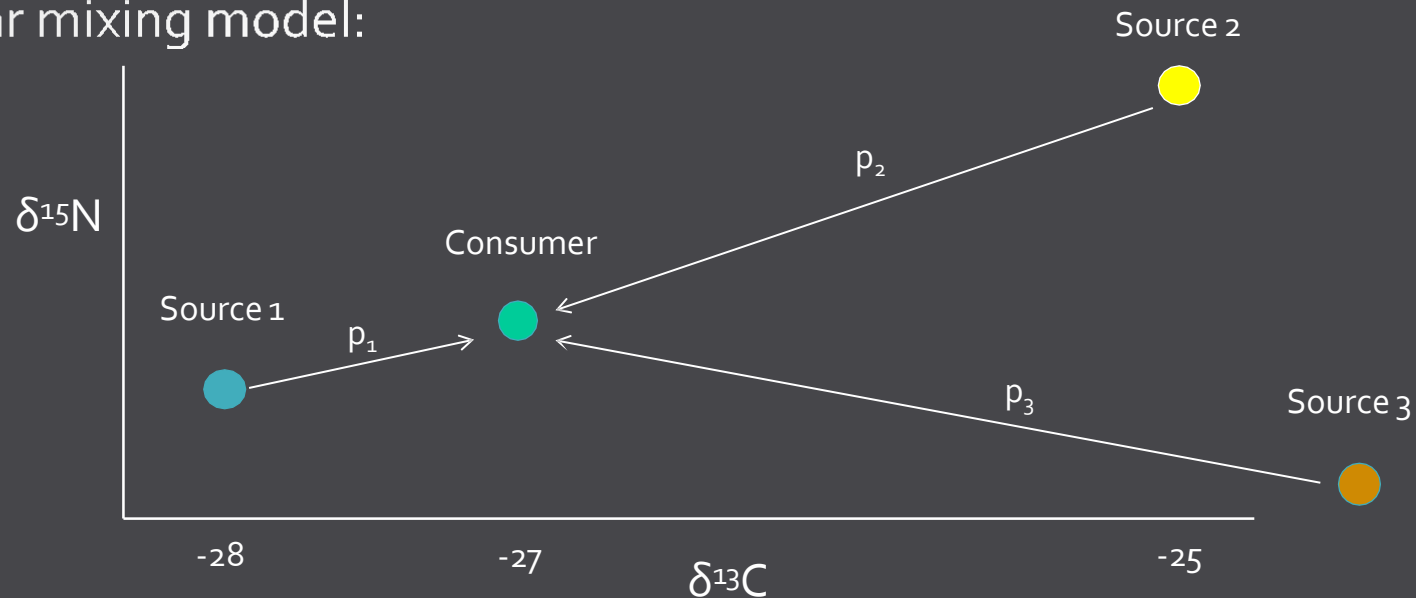
$$-27 = \frac{2}{3}(-28) + \frac{1}{3}(-25)$$



Twice as close to source 1 than source 2

# How mixing models work

Linear mixing model:



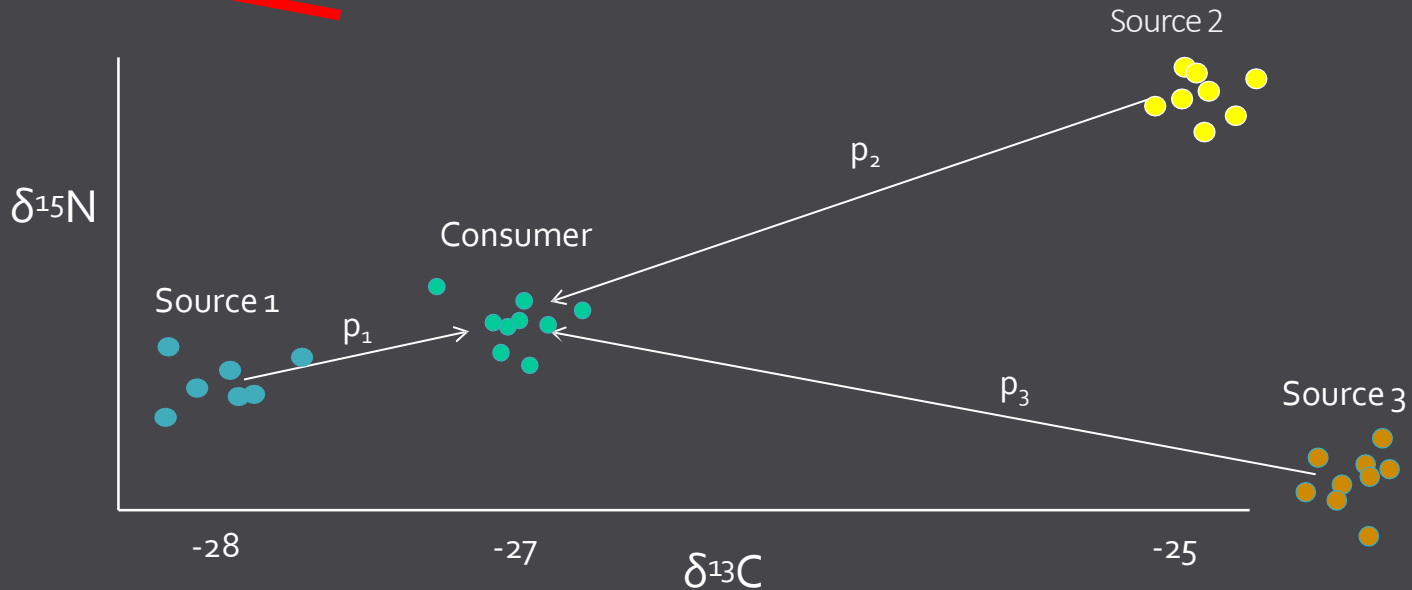
$$\text{Consumer}_C = p_1 s_{1C} + p_2 s_{2C} + p_3 s_{3C}$$

$$\text{Consumer}_N = p_1 s_{1N} + p_2 s_{2N} + p_3 s_{3N}$$

$$p_1 + p_2 + p_3 = 1$$

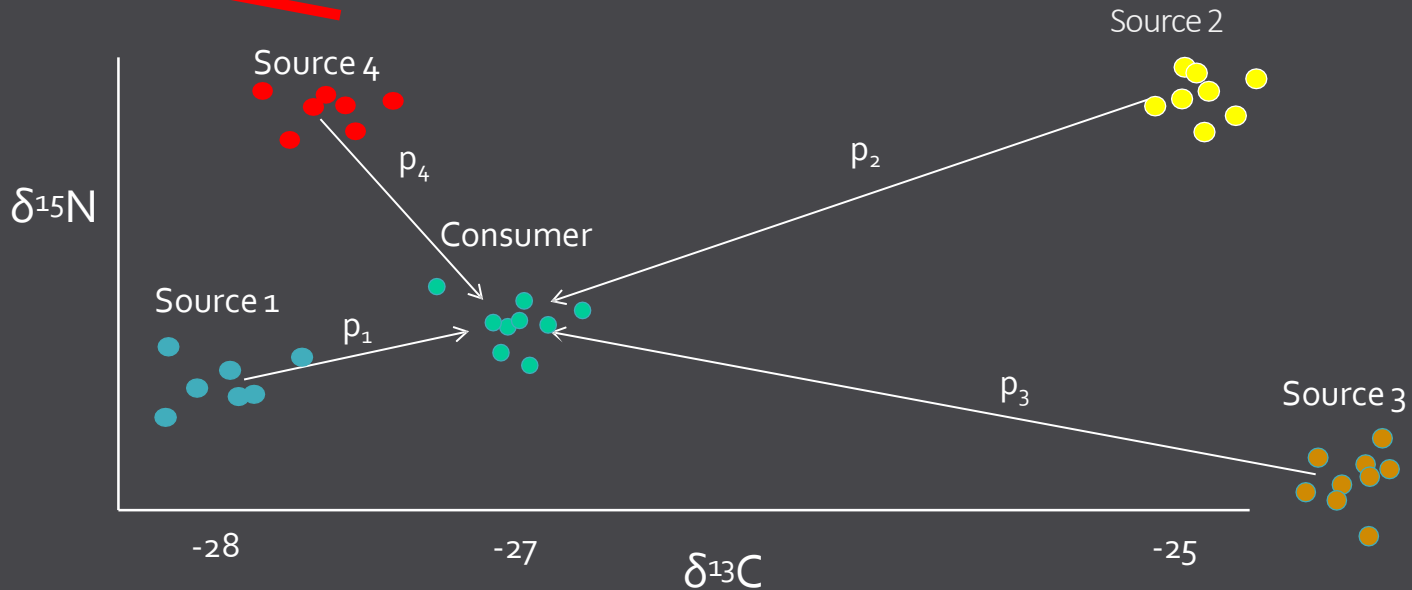
# How mixing models work

~~Linear mixing model:~~



# How mixing models work

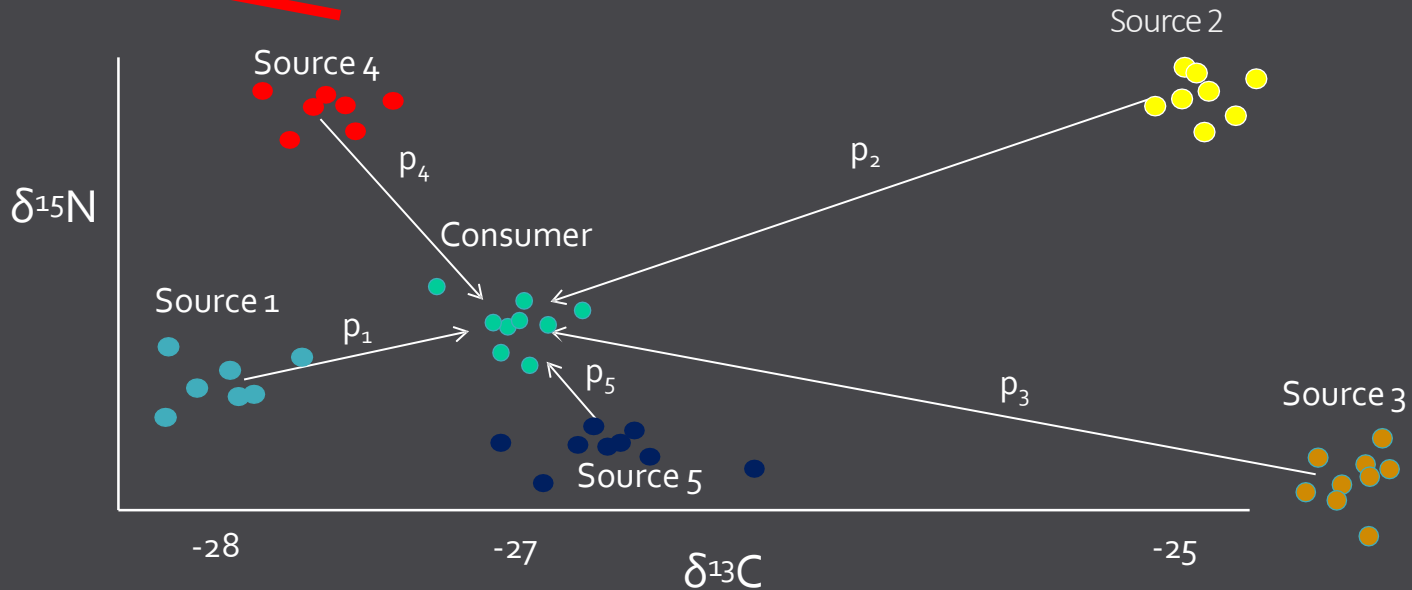
~~Linear mixing model:~~





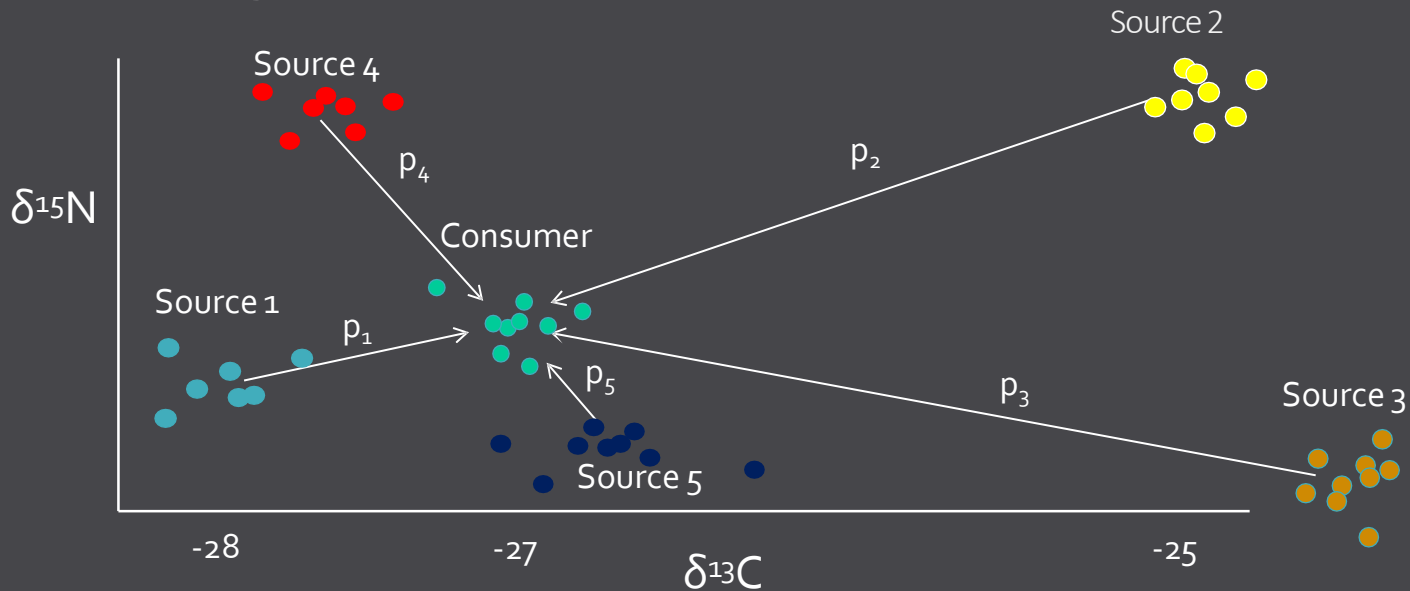
# How mixing models work

~~Linear mixing model:~~



# How mixing models work

Bayesian mixing model:



$$\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha})$$

$$Y_{ij} \sim \text{MVN} \left( \sum_k p_k \mu_{jk}^s, \Sigma \right)$$

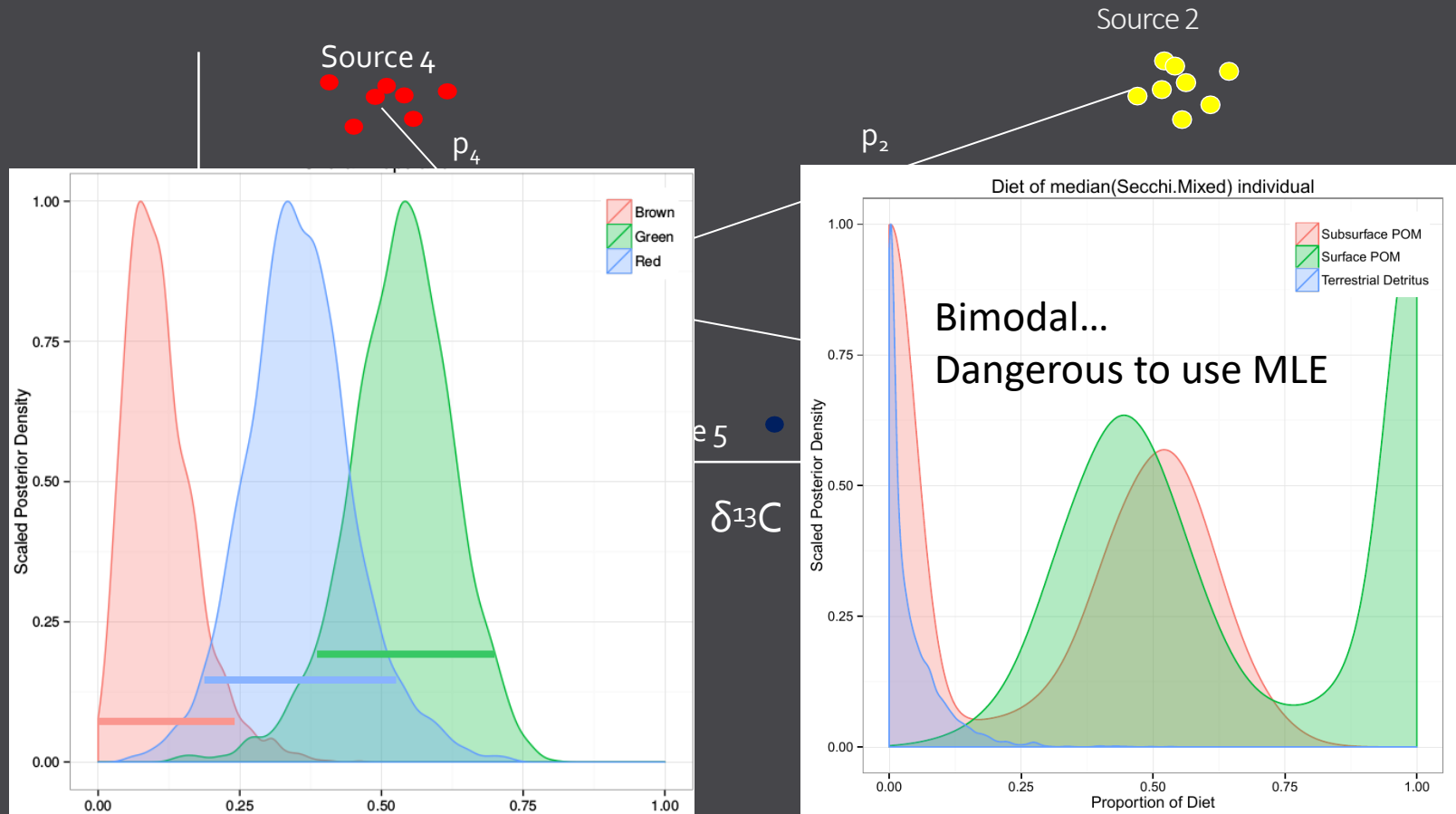
$$Y_{jk}^s \sim \text{MVN} (\mu_{jk}^s, \Sigma_k^s)$$

$$\mu_{jk}^s \sim \mathcal{N}(0, .001)$$

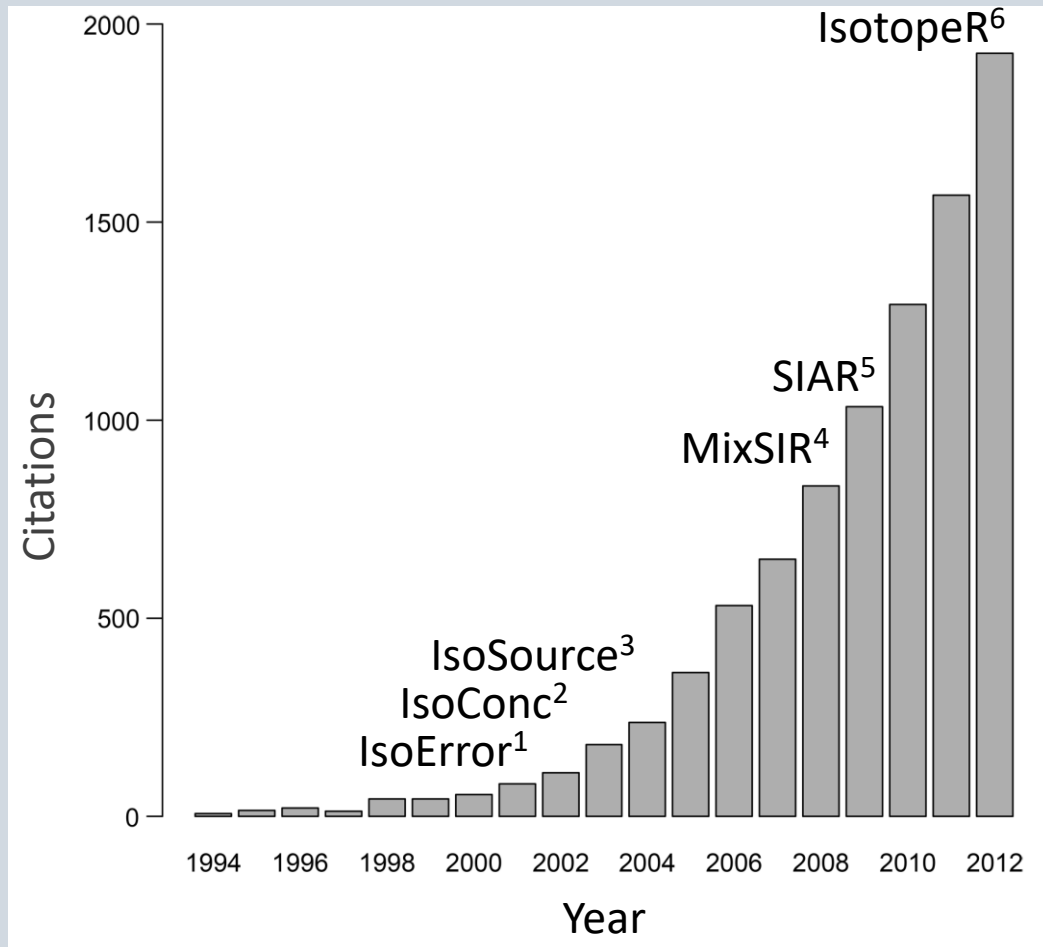
$$\tau_{jk} \sim \text{gamma}(.001, .001)$$

# How mixing models work

Bayesian mixing model:



# MixSIAR incorporates years of methods development



<sup>1</sup>Phillips and Gregg 2001a, 2001b

- Uncertainty (ad hoc)

<sup>2</sup>Phillips and Koch 2002

- **Concentration dependence**

<sup>3</sup>Phillips and Gregg 2003

- Underdetermined system

<sup>4</sup>Moore and Semmens 2008

- **Bayesian**, MATLAB GUI

<sup>5</sup>Parnell et al. 2010

- Bayesian, **residual error**

Ward et al. 2010

- **Fit source means**

<sup>6</sup>Hopkins and Ferguson 2012

- **Covariance**

Parnell et al. 2013

- **Covariate effects**

# MixSIAR: the general idea

1. Load data in R
2. Choose model options

```
if(mix$n.re > 0){ # at least 1 random effect
cat("
  fac1.sig ~ dunif(0,20);
  fac1.invSig2 <- 1/(fac1.sig*fac1.sig);
  # draw the fac1 (region) specific ILR terms (random effect)
  for(f1 in 1:factor1_levels) {
    for(src in 1:(n.sources-1)) {
      ilr.fac1[f1,src] ~ dnorm(0,fac1.invSig2);
    }
  }
", file=filename, append=T)}
```

```
if(resid_err){
cat("
  # Additive residual error (SIAR)
  for(iso in 1:n.iso){
    resid.prcsn[iso] ~ dgamma(.001,.001);
    resid.var[iso] <- 1/resid.prcsn[iso];
  }
", file=filename, append=T)}
```

3. write\_JAGS\_model.r

4. Run model in JAGS
5. Diagnostics + output in R

# The plan

~~1. Introduction~~

2. Source fitting

3. Covariate effects

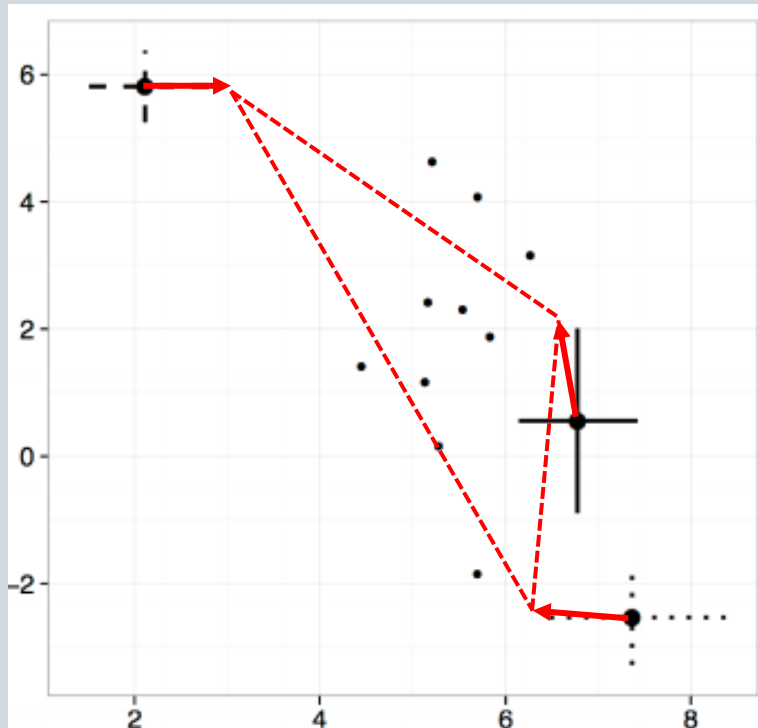


Don't think other software  
currently includes these (simmr)

4. Effects of priors

5. GitHub site + resources

# Source fitting



Account for *sampling error of the sources* (allow source means to deviate from the sample means)

1. Raw data
  - Parnell et al. 2013
  - includes covariance
2. Mean / SD / n
  - Ward et al. 2010
3. Fixed
  - set  $n = 10000$

\*Source and TDF variance indistinguishable

$$\sqrt{\sigma_{source}^2 + \sigma_{discr}^2}$$



# Source fitting

<http://dx.doi.org/10.7717/peerj.5096>



## Supplement 1

### MixSIAR Model Description

#### Contents

Preamble	.....
Source data	.....
Raw source data	.....
Mean / SD / n source data	.....

Account for *sampling error of the sources* (allow source means to deviate from the sample means)

1. Raw data
  - Parnell et al. 2013
  - includes covariance
2. Mean / SD / n
  - Ward et al. 2010
3. Fixed
  - set n = 10000

\*Source and TDF variance indistinguishable

$$\sqrt{\sigma_{source}^2 + \sigma_{discr}^2}$$

## 2. Source fitting

Parnell et al. (2013)

<https://doi.org/10.1002/env.2221>

Ward et al. (2010)

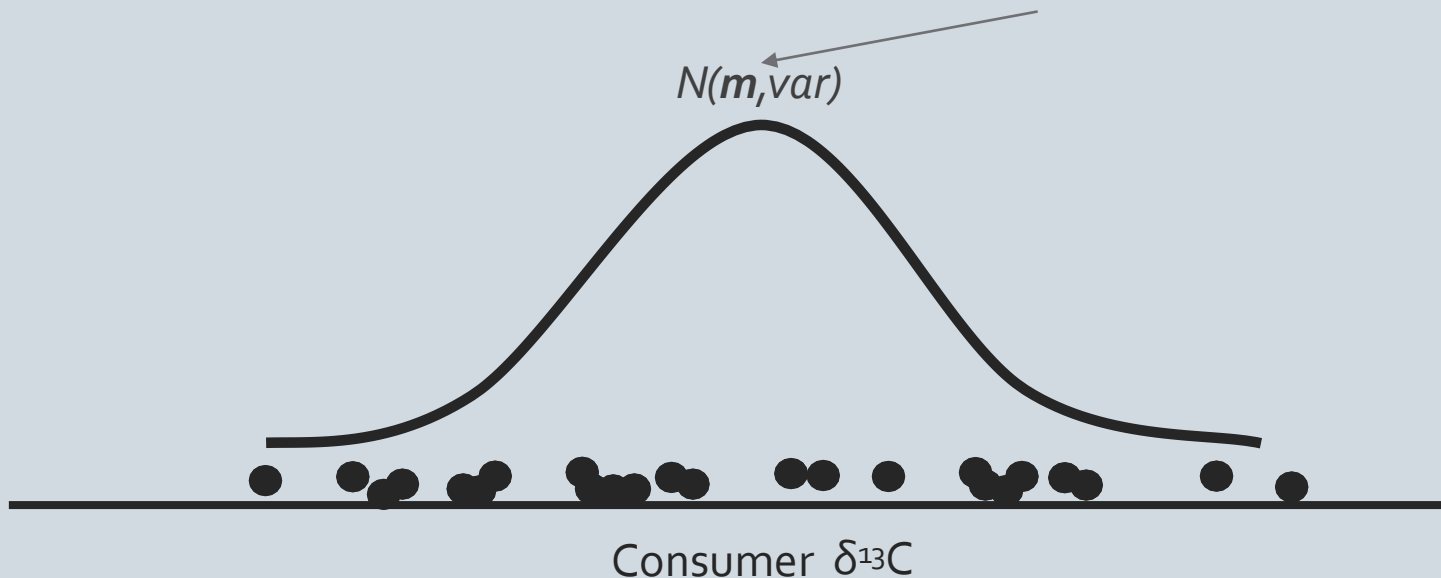
<https://doi.org/10.1021/es100053v>

# Covariate effects in MixSIAR

No covariate effects...

$$\mathbf{p} = [20\%, 50\%, 20\% \ 10\%]$$

Assumes that all consumers have the *same diet*



## 3. Covariate effects

# Covariate effects in MixSIAR

Transform  $\mathbf{p}$ 's

Linear regression in ILR-space

$$\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha})$$

$$p_{ik} = \text{inverseILR}(\text{ilr.global}_k + \text{ilr.fac1}_{mk} + \text{ilr.cont1}_k \text{Cont1}_i)$$

# Covariate effects in MixSIAR

Transform  $\mathbf{p}$ 's

Linear regression in ILR-space

$$\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha})$$

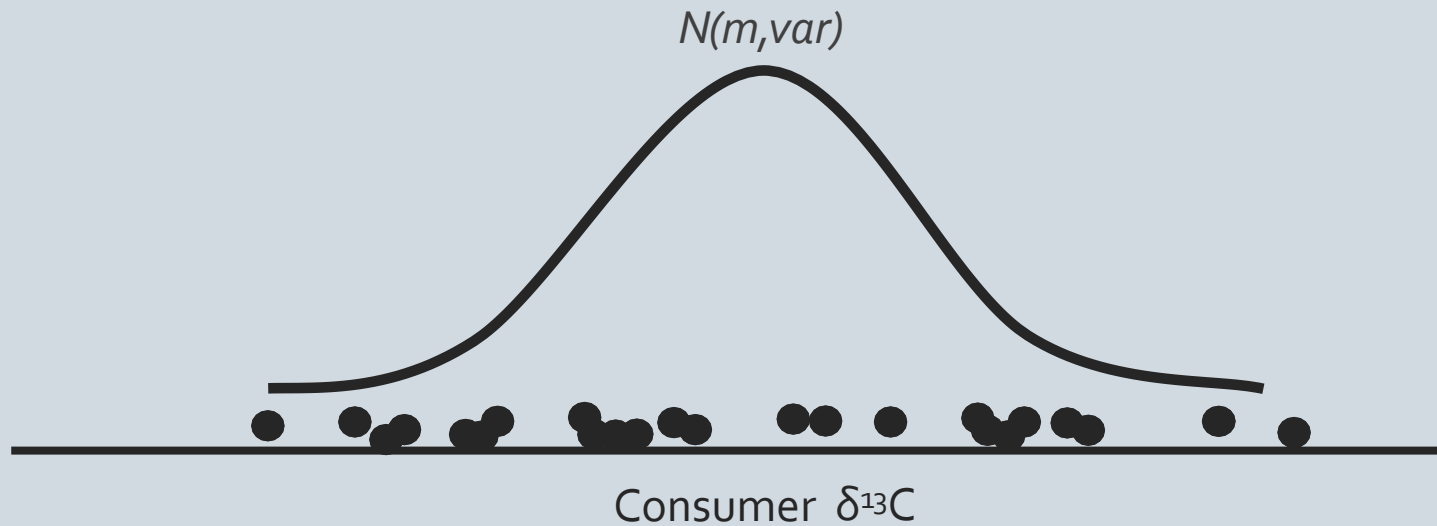
$$p_{ik} = \text{inverseILR}(\text{ilr.global}_k + \text{ilr.fac1}_{mk} + \text{ilr.cont1}_k \text{Cont1}_i)$$

↑  
Intercept/mean

↑  
Fixed/random effect

↑  
Continuous effect  
("slope")

# Fixed effects



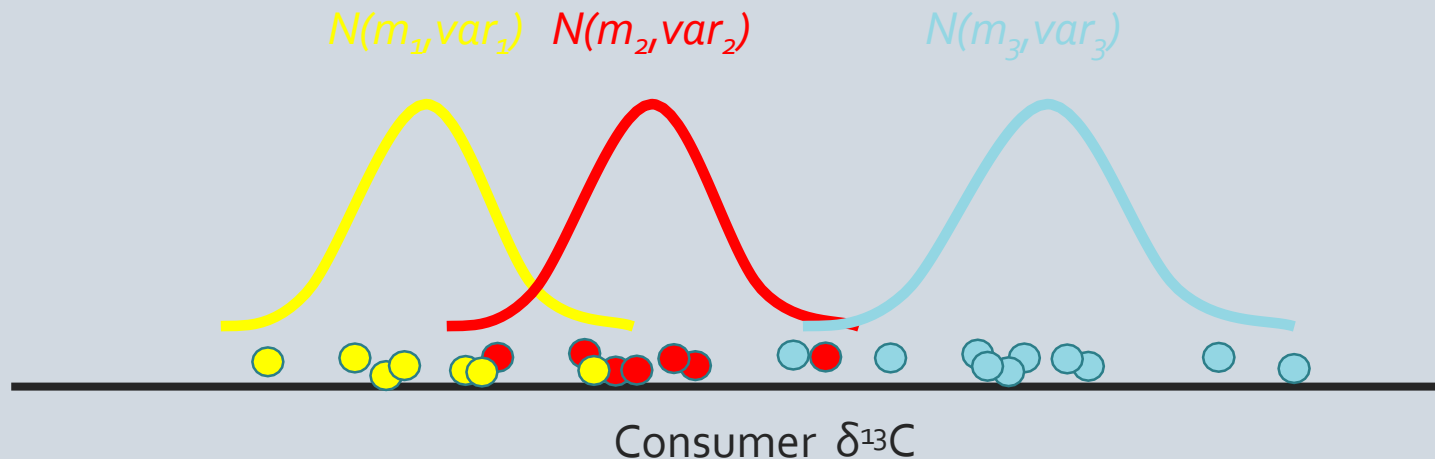
## 3. Covariate effects

# Fixed effects

Simplest, estimate mean  $\mu$  for different groups independently

Similar to running separate mixing models (e.g. SIAR, simmr), but

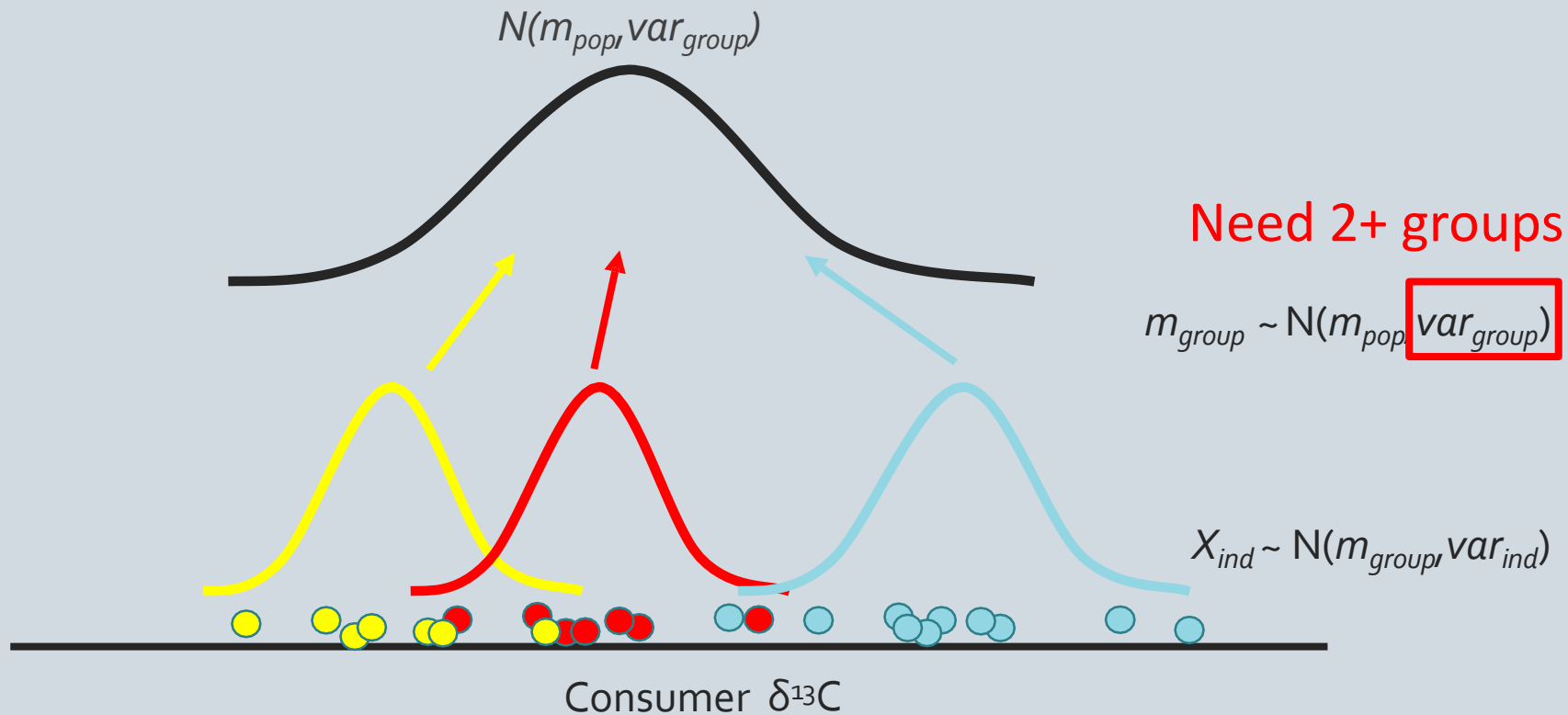
- Source means, error terms fit 1x and shared across groups



## 3. Covariate effects

# Random effects

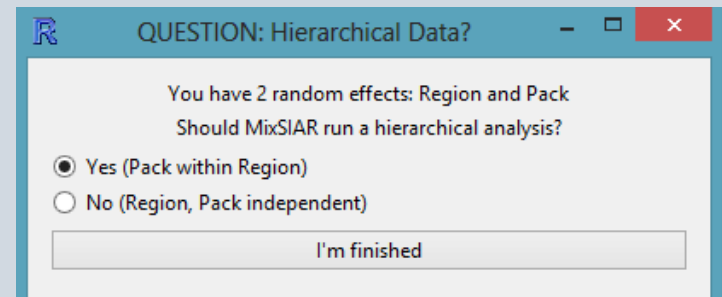
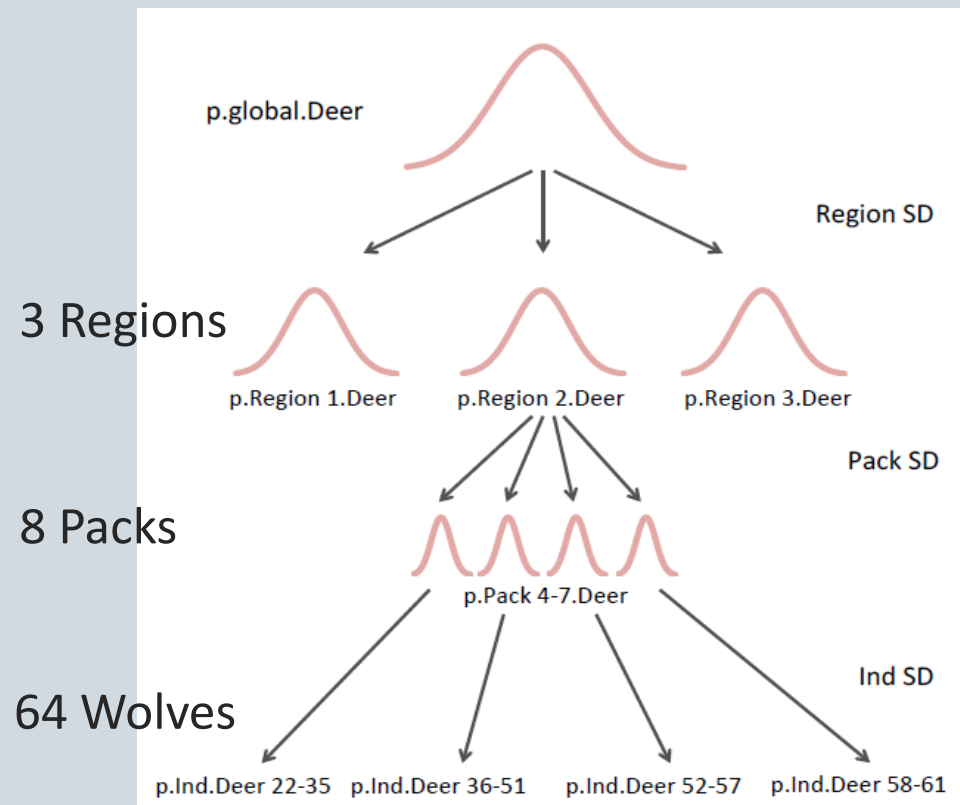
More complex



## 3. Covariate effects



# Nested/hierarchical random effects



Hierarchical

```
ilr.fac2.tot[f2,src] <- ilr.global[src] + ilr.fac1[factor1_lookup[f2],src] + ilr.fac2[f2,src];
```

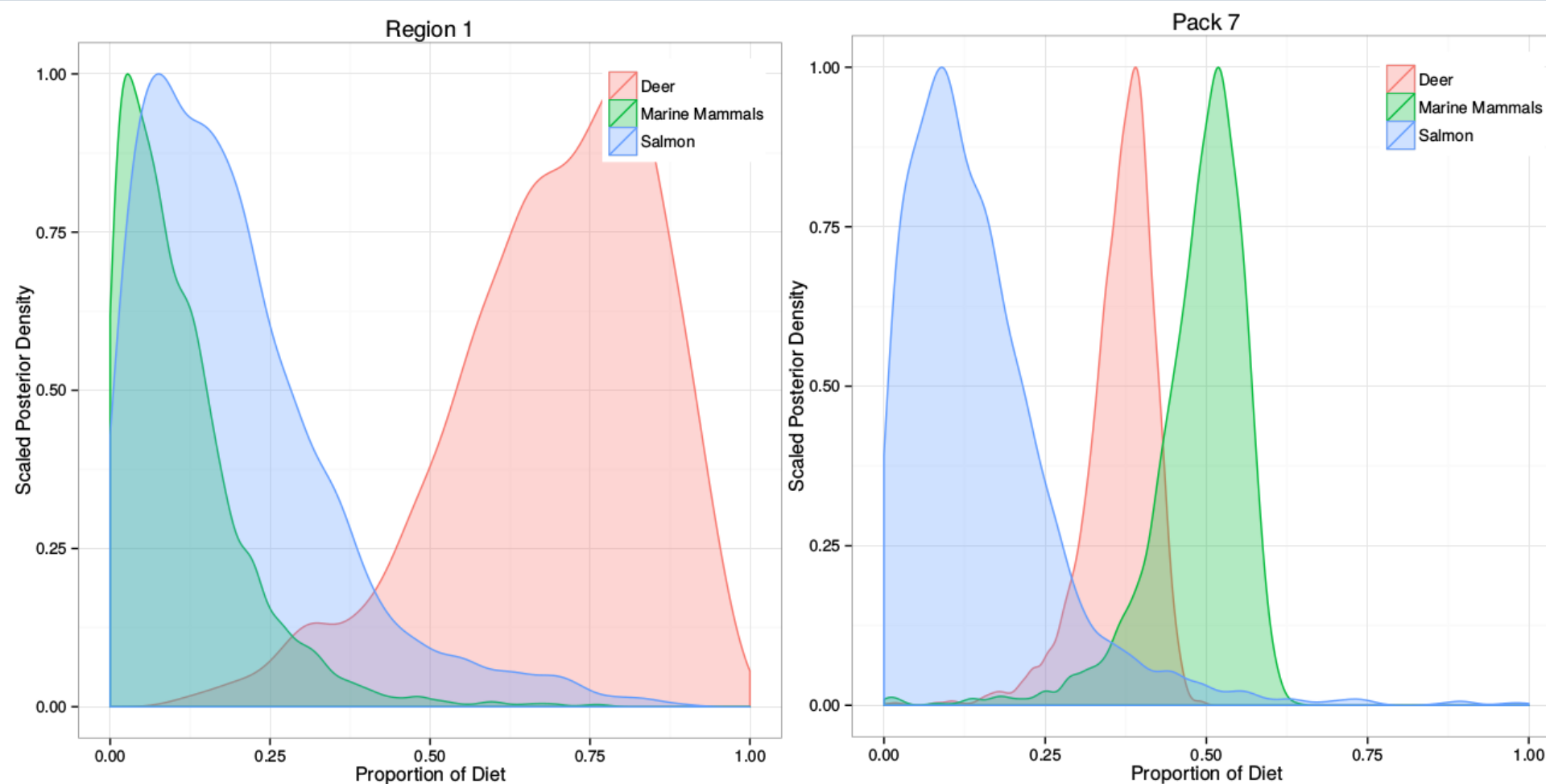
Independent

```
ilr.fac2.tot[f2,src] <- ilr.global[src] + ilr.fac2[f2,src];
```

## 3. Covariate effects

Semmens et al. (2009)

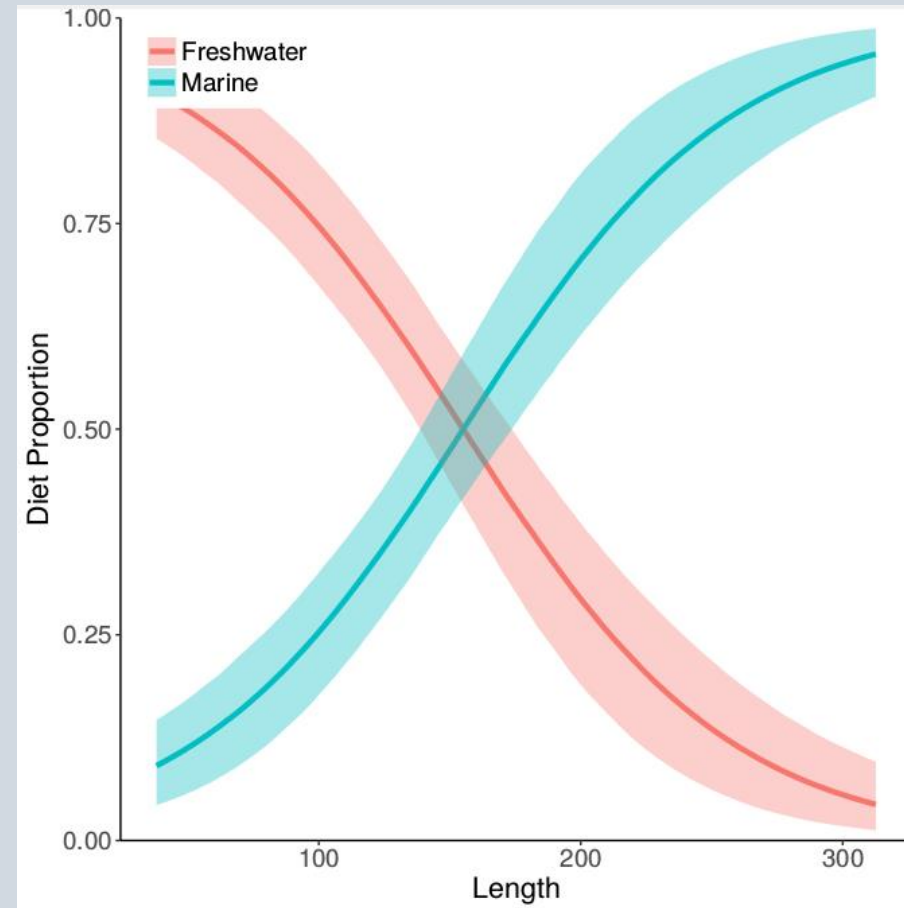
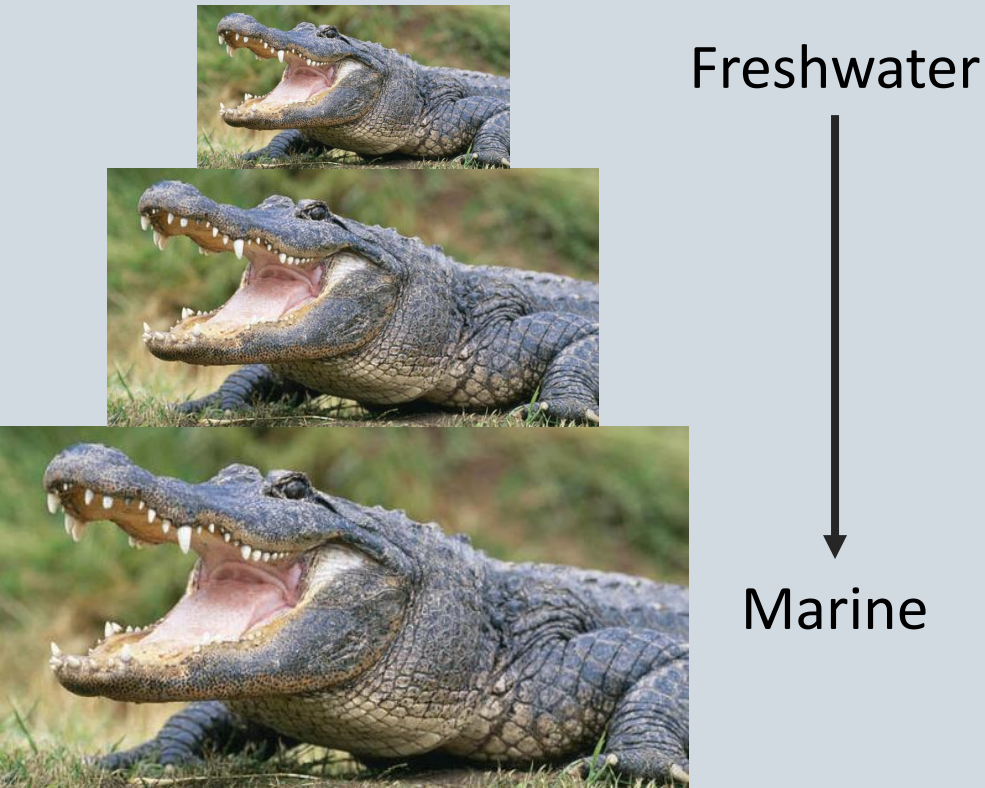
# (Nested) Random effects: Wolves Ex



## 3. Covariate effects

Semmens et al. (2009)

# Continuous effect: Alligator length



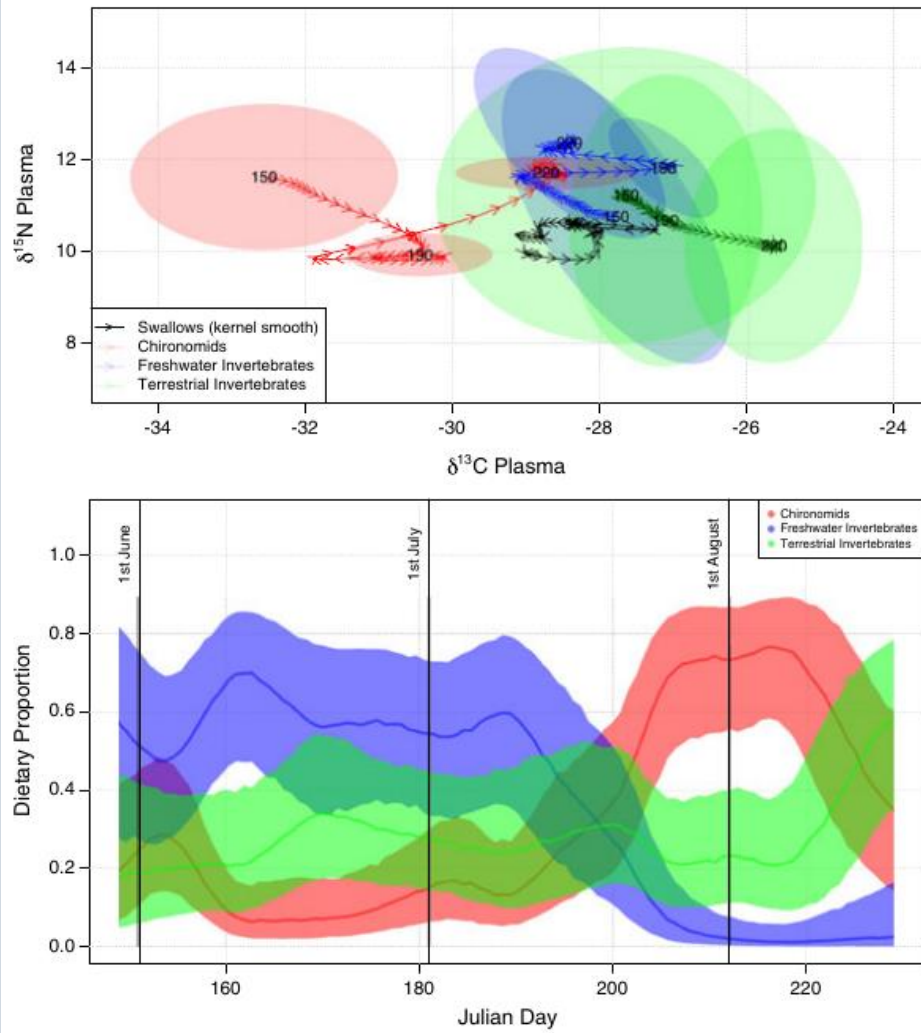
## 3. Covariate effects

Nifong et al. (2015)  
Stock et al. (2018)

# More complexity is possible ...way beyond MixSIAR

## Multivariate spline model

1. Time-varying source means and covariances
2. Time-varying diet proportions



## 3. Covariate effects

Parnell et al. (2013)

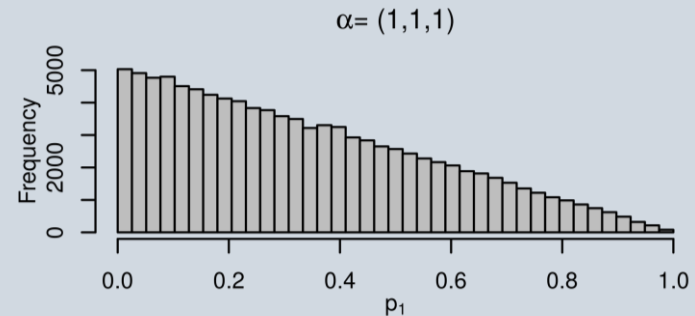
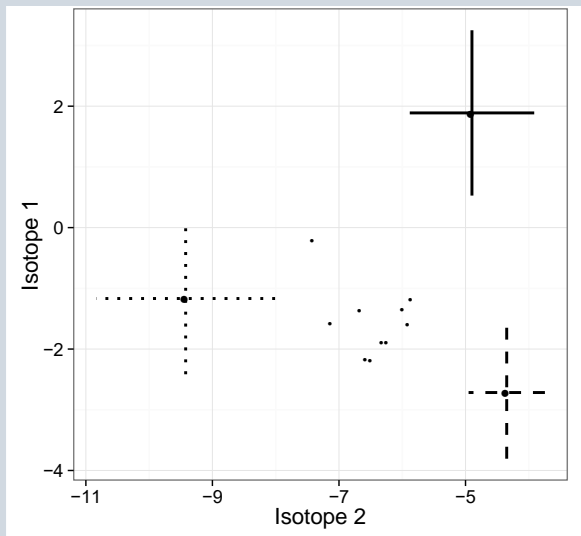
<https://doi.org/10.1002/env.2221>

## 4. Effect of priors/ “Bayesian mixing models are biased”

---

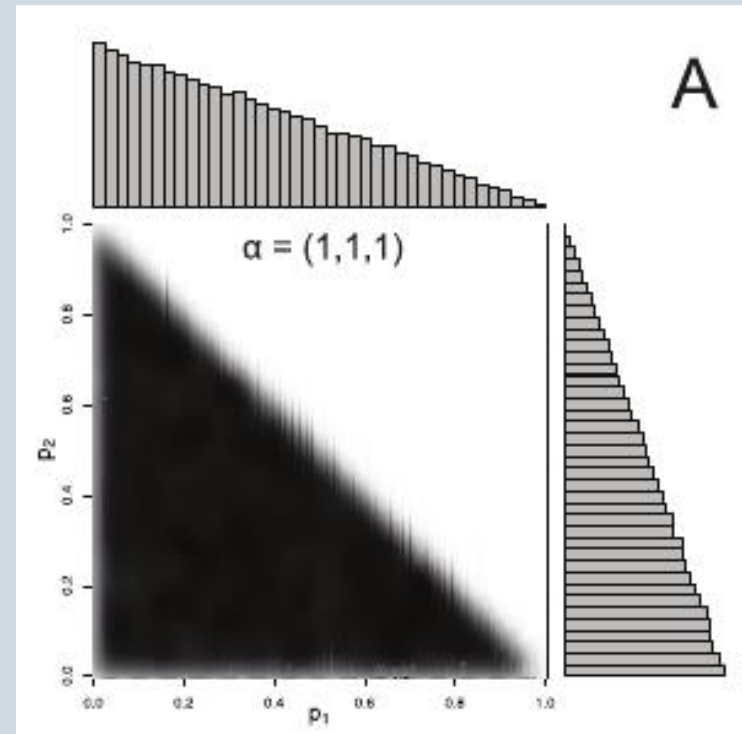
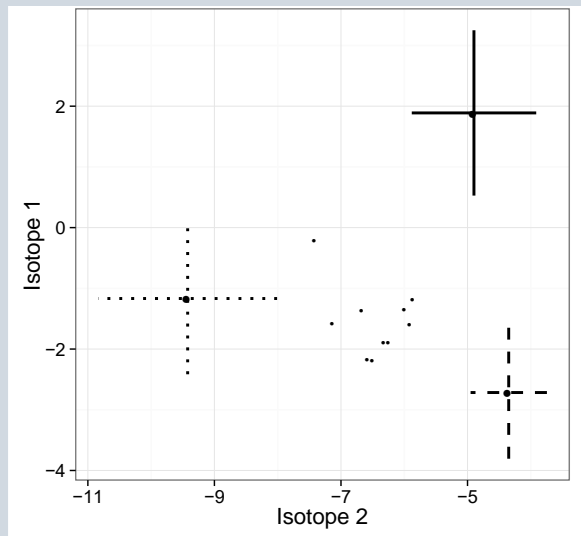
# a. There is no “uninformative” prior

Problem: proportions are not independent!



## a. There is no “uninformative” prior

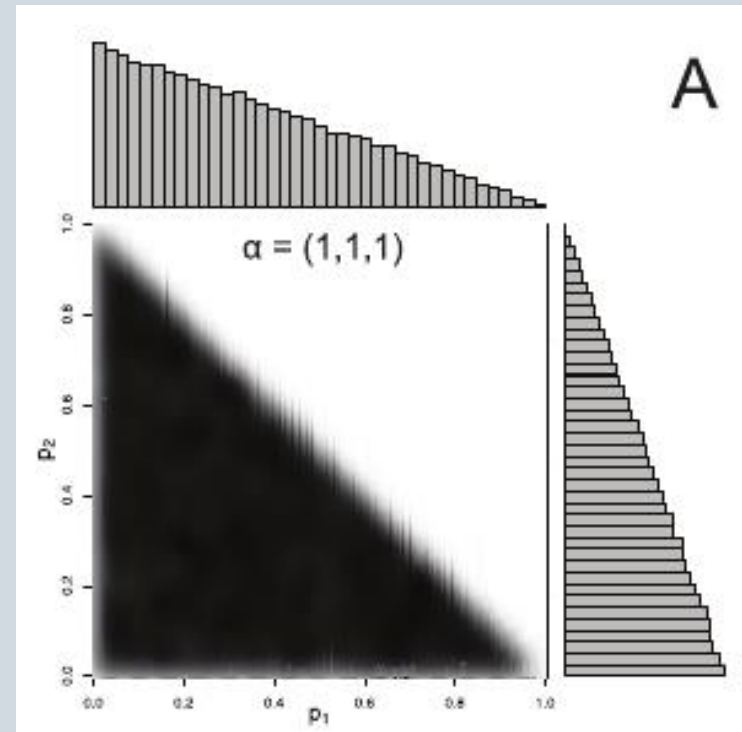
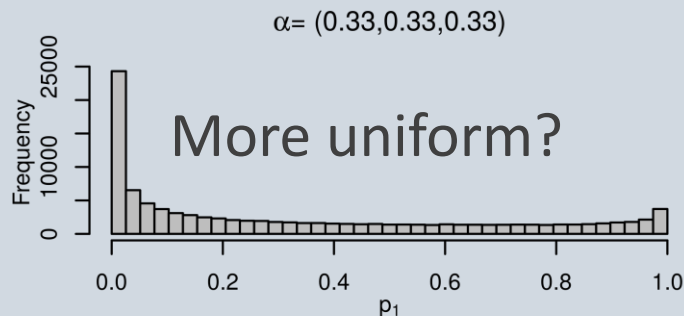
Problem: proportions are not independent!



All  $\mathbf{p}$  equally likely  
“uninformative” / generalist

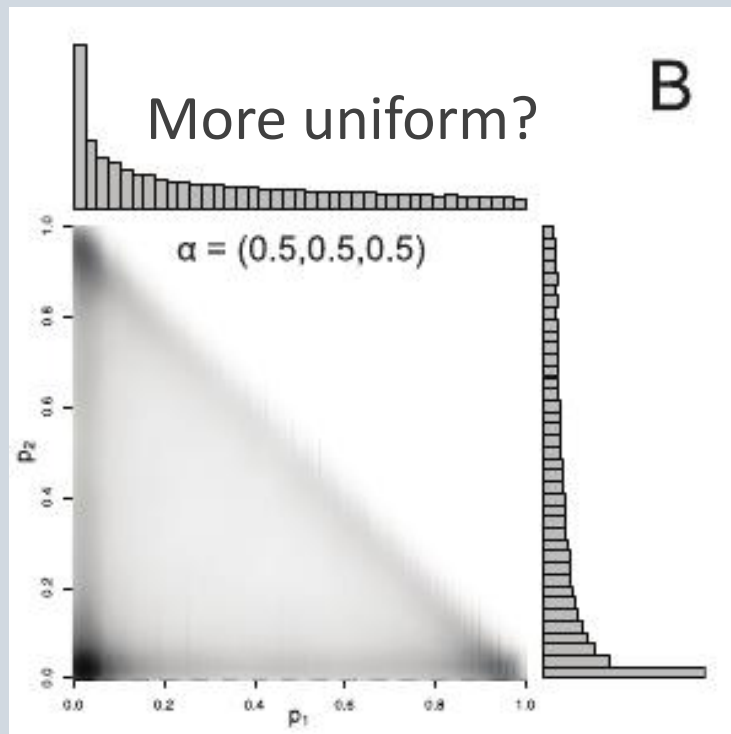


a. There is no “uninformative” prior

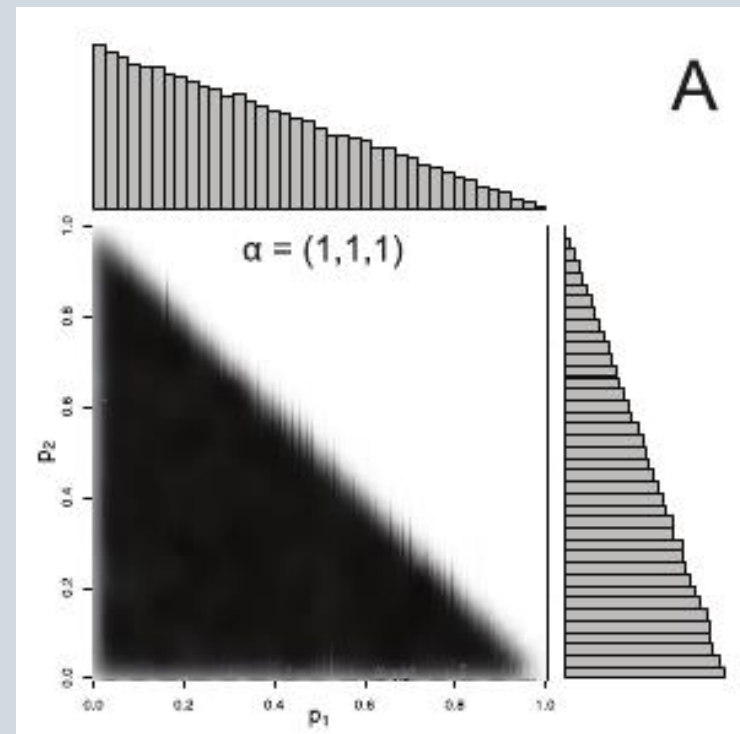


All  $\mathbf{p}$  equally likely  
“uninformative” / generalist

a. There is no “uninformative” prior



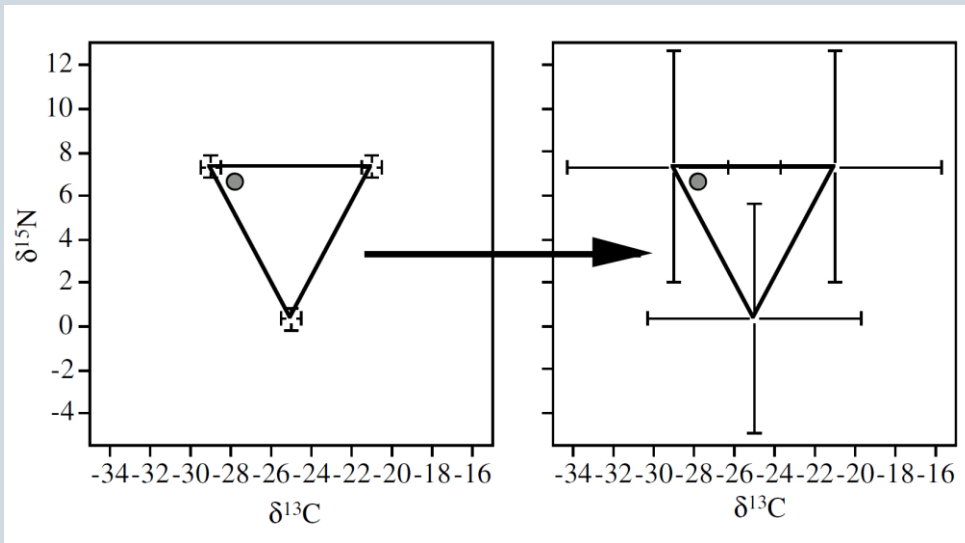
Extreme  $\mathbf{p}$  heavily weighted



All  $\mathbf{p}$  equally likely  
“uninformative” / generalist

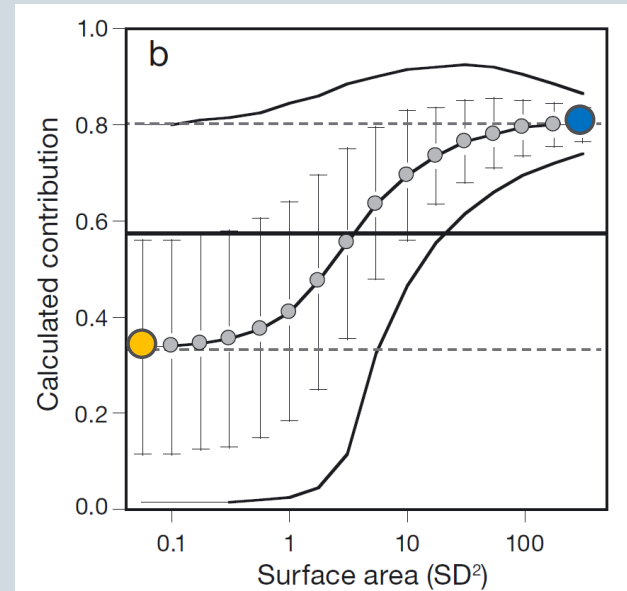
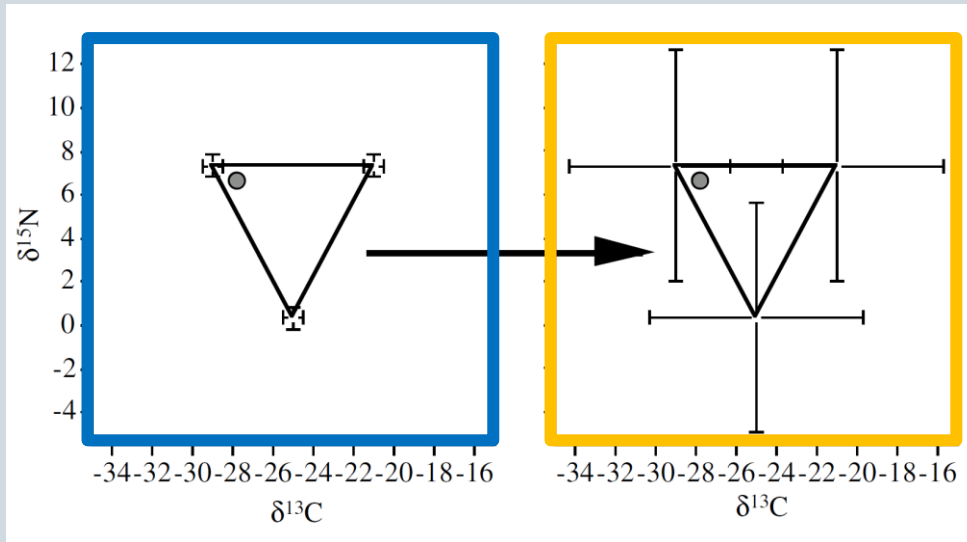
## b. Effect of the “uninformative” prior

### 1. How good is your data?



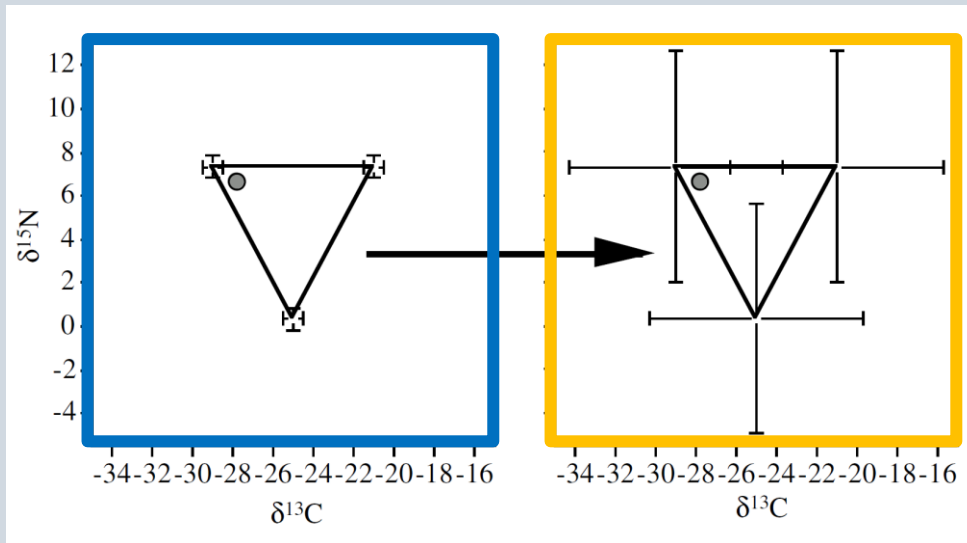
## b. Effect of the “uninformative” prior

### 1. How good is your data?



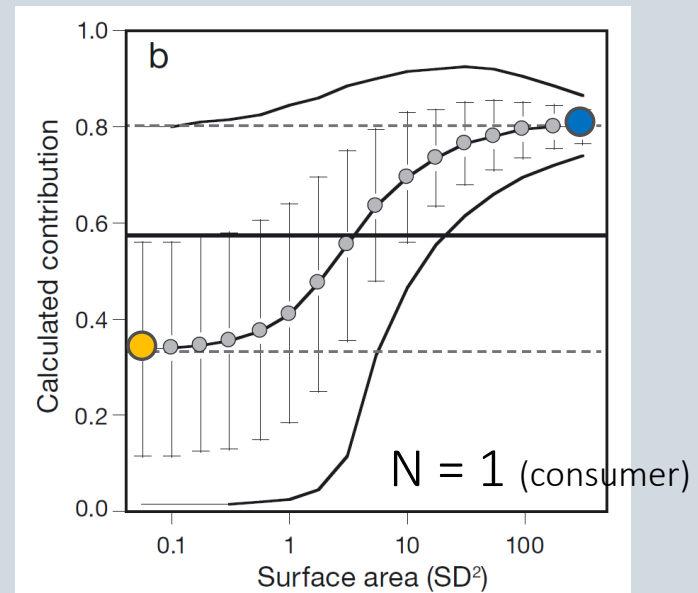
## b. Effect of the “uninformative” prior

1. How good is your data?



- Consumer sample size and variance
- Source sample size and variance
- Source geometry (are you unlucky?)

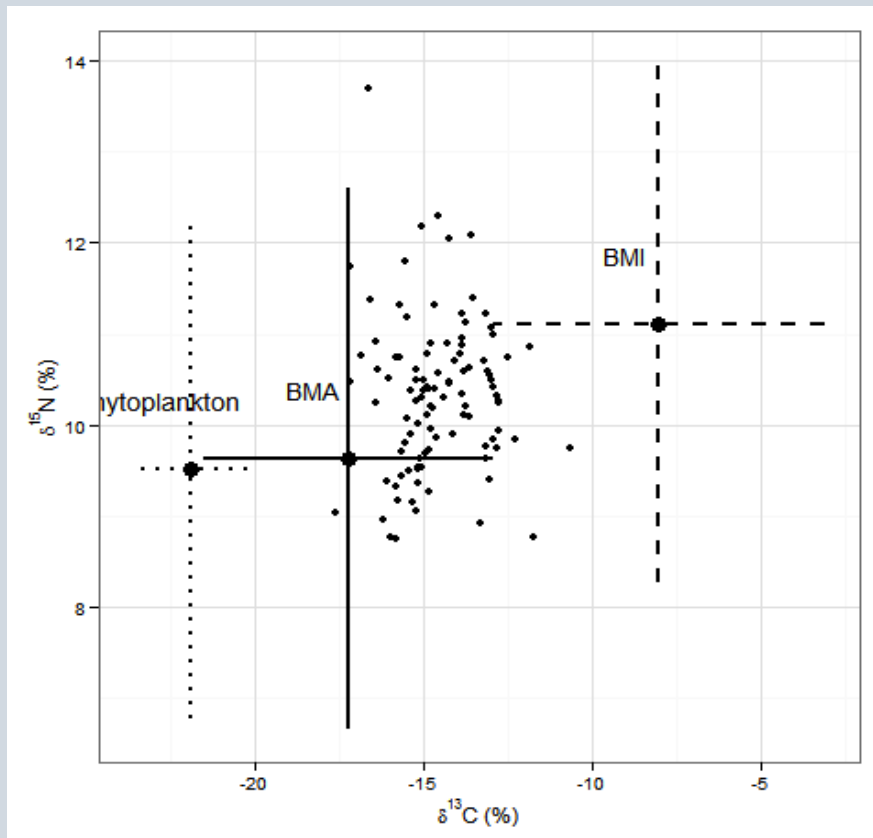
2. How much data do you have?



# c. Source geometry

High source uncertainty\*

- Collect more samples
- TDF experiment



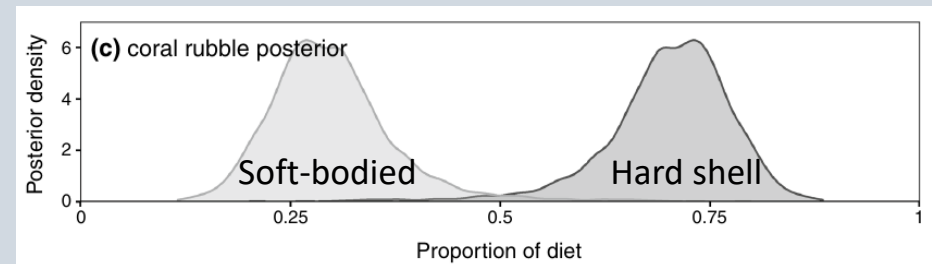
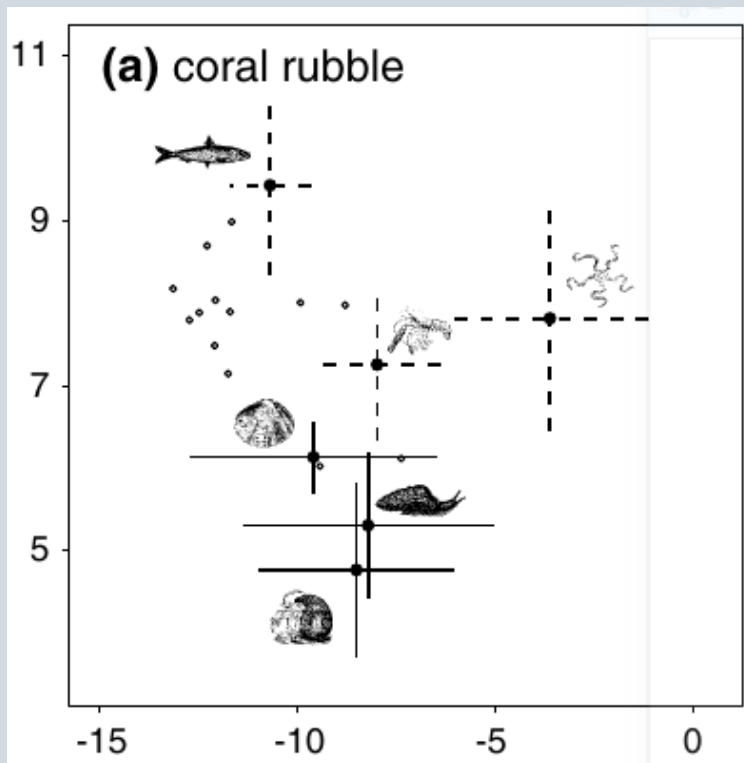
\*Source and TDF variance indistinguishable

$$\sqrt{\sigma_{\text{source}}^2 + \sigma_{\text{discr}}^2}$$

## c. Source geometry

Too many sources...

- combine (before/after)
- informative prior(s)
- more tracers

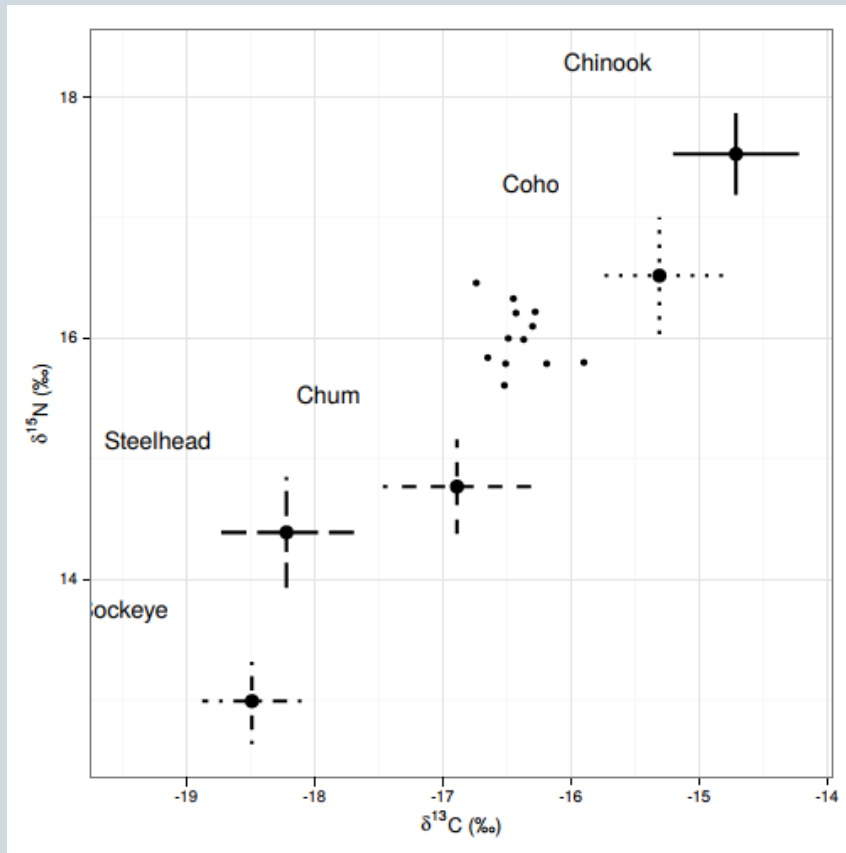


[https://github.com/brianstock/MixSIAR/blob/master/inst/example\\_scripts/mixsiar\\_script\\_mantis.R](https://github.com/brianstock/MixSIAR/blob/master/inst/example_scripts/mixsiar_script_mantis.R)

## c. Source geometry

Sources confounded, multiple solutions...

- informative prior (stomach or fecal contents, visual obs, opinion)



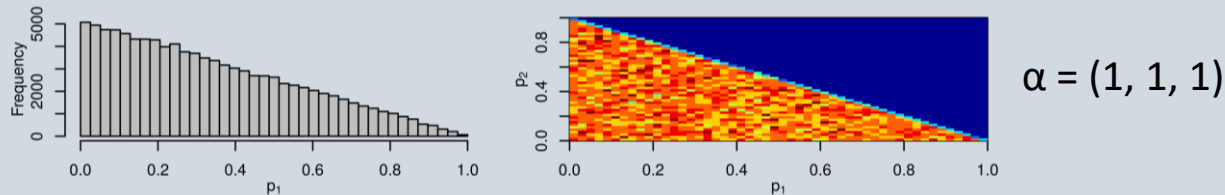
\*Source and TDF variance indistinguishable

$$\sqrt{\sigma_{\text{source}}^2 + \sigma_{\text{discr}}^2}$$



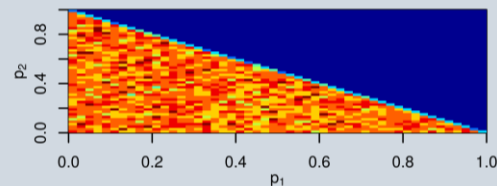
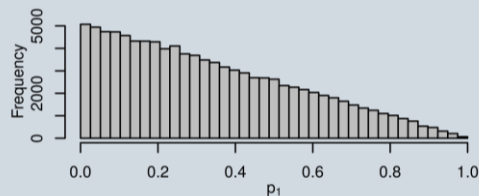
## d. Constructing informative priors

You control the mean proportions AND the variance (“informativeness”)

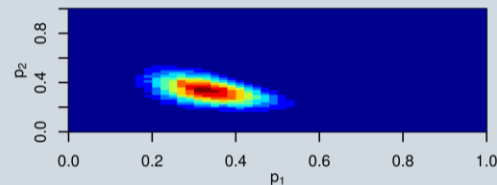
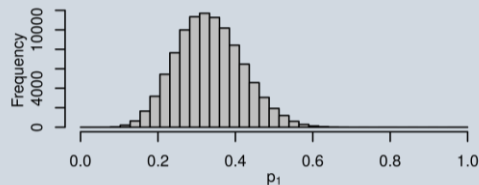


# d. Constructing informative priors

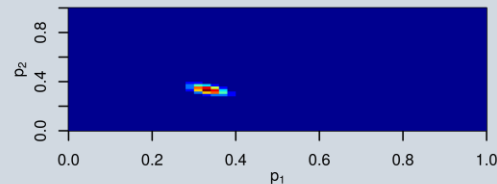
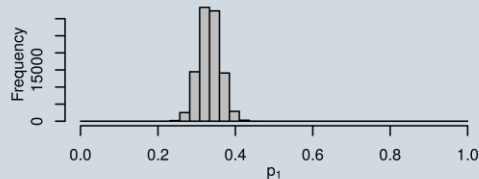
You control the mean proportions AND the variance (“informativeness”)



$\alpha = (1, 1, 1)$



$\alpha = (10, 10, 10)$



$\alpha = (100, 100, 100)$

## d. Constructing informative priors

You control the mean proportions AND the variance (“informativeness”)

30



8



25



# d. Constructing informative priors

You control the mean proportions AND the variance (“informativeness”)

30



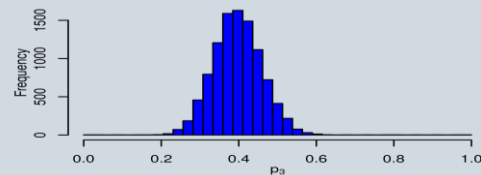
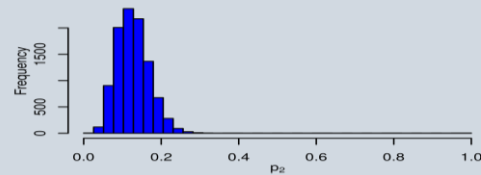
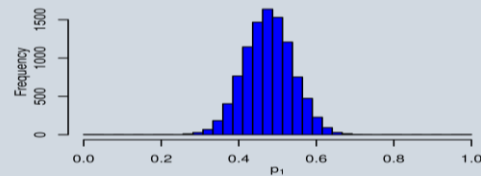
8



25



$$\alpha = (30, 8, 25)$$



# d. Constructing informative priors

You control the mean proportions AND the variance (“informativeness”)

30



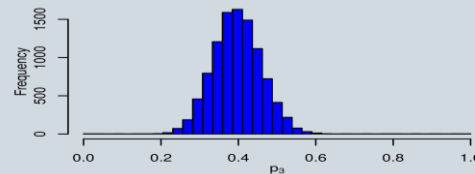
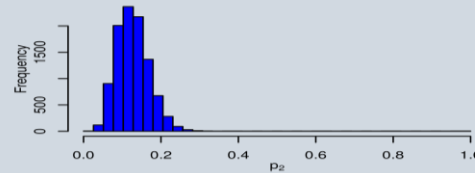
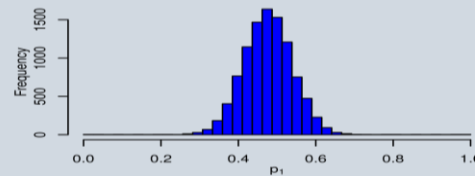
8



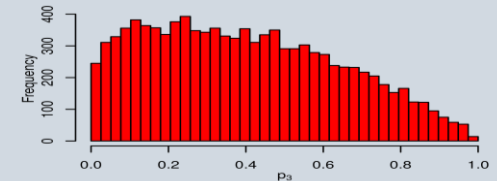
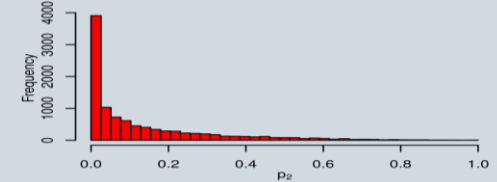
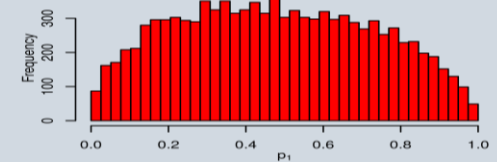
25



$$\alpha = (30, 8, 25)$$



$$\alpha = \frac{3 * (30, 8, 25)}{63}$$



# Where do I get MixSIAR?

## CRAN (older)

1. Download and install/update **R**
2. Download and install **JAGS**
3. Open R and run:

```
install.packages("MixSIAR")
```

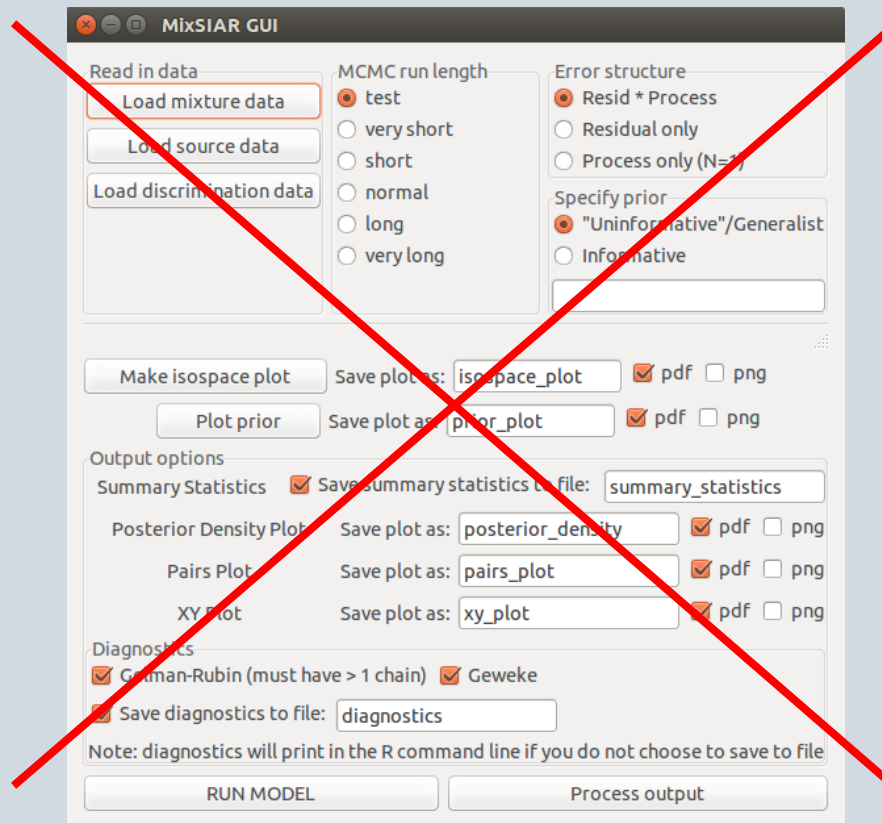
## GitHub (latest)

1. Download and install/update **R**
2. Download and install **JAGS**
3. Open R and run:

```
library(devtools)  
install_github("brianstock/MixSIAR")
```

# MixSIAR: GUI RIP

Graphical User Interface (GUI) deprecated... RGtk2...



# MixSIAR manual

[https://github.com/brianstock/MixSIAR/blob/master/inst/mixsiar\\_manual\\_small.pdf](https://github.com/brianstock/MixSIAR/blob/master/inst/mixsiar_manual_small.pdf)

## Contents

<b>1 Introduction</b>	<b>3</b>
1.1 What is the MixSIAR GUI?	3
1.2 What is the MixSIAR model framework? How does it relate to SIAR and MixSIR?	3
1.3 Model options	4
1.4 Great! How do I get started using MixSIAR?	5
<b>2 Installation</b>	<b>5</b>
2.1 Windows	6
2.2 Mac OS X	7
2.3 Linux	8
<b>3 Running the working examples</b>	<b>8</b>
3.1 Wolves Example	11
3.2 Geese Example	23
3.3 Lake Example	27
3.4 Palmyra Example	31
3.5 Killer Whale Example	35
3.6 Isopod Example	39
3.7 Cladocera Example	43
3.8 Storm-petrel Example	48
3.9 Snail Example	48
3.10 Mantis Example	49
3.11 Alligator Example	58
<b>4 Using MixSIAR with your own data</b>	<b>61</b>
4.1 Running your own data	61
4.2 Data file format and loading	62
4.3 Using the MixSIAR script version	64
4.4 GitHub Issues page	67
4.5 Citing MixSIAR	69
<b>5 Introductions to Bayesian Analysis</b>	<b>70</b>
5.1 Convergence Diagnostics	70

## 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”).

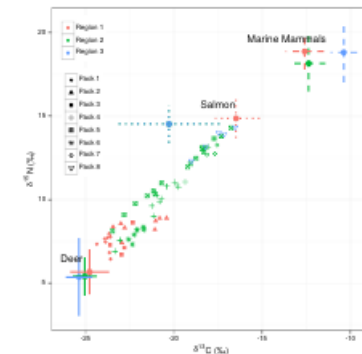


Figure 2: **Stable isotope input for the Wolves Example.** Mixture data (wolves) are by Region (color) and Pack (shape). Source data are by Region and have been adjusted by discrimination means and SDs. Error bars indicate  $\pm 1$  SD, the combined source+discrimination SD calculated as  $\sqrt{\sigma_{\text{source}}^2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



# MixSIAR model description

<http://dx.doi.org/10.7717/peerj.5096>



## Analyzing mixing systems using a new generation of Bayesian tracer mixing models

Brian C. Stock<sup>1</sup>, Andrew L. Jackson<sup>2</sup>, Eric J. Ward<sup>3</sup>, Andrew C. Parnell<sup>4</sup>, Donald L. Phillips<sup>5</sup> and Brice X. Semmens<sup>1</sup>

## Supplement 1: advice on model setup

### MixSIAR Model Description

#### Contents

Preamble	1
Source data	1
Raw source data	2
Mean / SD / n source data	3
Source data by factor	3
Fixed / random / continuous effects	3
Fixed effects	4
Random effects	4
Choosing between fixed and random effects	4
Continuous effects	5
Nested vs. non-nested factors	5
Mixture mean	5
Mixture variance	6
Process x Residual error (with covariance)	6
Process x Residual error (no covariance)	7
Residual error only	7
Process error only	7
Mixture likelihood	7
References	8

# MixSIAR website + vignettes

<http://brianstock.github.io/MixSIAR/>

The screenshot shows the MixSIAR website interface. At the top, there are navigation links: "MixSIAR 3.1.12", "Vignettes", "Functions", "Source code", and "Issues". The "Vignettes" link is active, and a dropdown menu is open, listing: "Overview", "Ex 1: Wolves", "Ex 2: Geese", "Ex 3: Lake", "Ex 4: Killer whale", "Ex 5: Isopod", "Ex 6: Cladocera", and "Modifying output plots". The main content area is titled "MixSIAR" and describes it as an R package for running Bayesian mixing models. It lists features such as the number of biotracers, hierarchical data fitting, categorical covariates, continuous covariates, error structure options, concentration dependence, and model comparison. A list of vignettes is provided, including a full description of equations, advice on common issues, and a case study. The "Installation" section notes that the GUI has been removed from the CRAN version. On the right side, there are sections for "Links" (CRAN download, source code, bug report), "License" (GPL-3), "Citation" (Citing MixSIAR), "Developers" (Brian Stock, Brice Semmens), and "Dev status" (CRAN version 3.1.12, 48K downloads, DOI 10.5281/zenodo.1209993).

MixSIAR 3.1.12 Vignettes Functions Source code Issues

## MixSIAR

MixSIAR is an R package (i.e. stable isotopes, fatty acids) for running Bayesian mixing models to analyze biotracer data (e.g. sources by Region). MixSIAR represents a collaborative coding project between Eric Ward, Andrew Parnell, Brice Semmens, Brian Stock, and Andrew Jackson.

MixSIAR incorporates several features:

- Any number of biotracers (examples with 1 isotope, 2 isotopes, 8 fatty acids, 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)
- Continuous covariate (up to 1)
- Error structure options with covariance (Residual \* Process, Residual only)
- Concentration dependence
- Plot and include "uninformative"/generalist or informative priors
- Fit multiple models and compare relative support using LOO/WAIC weights

For details, please see the [MixSIAR paper](#):

- Full description of equations
- Advice/explanation on 4 common issues (error structures, priors, combining sources, covariates)
- Case study highlighting new functionality (model selection with LOO/WAIC weights)

Stock BC, Jackson AL, Ward EJ, Parnell AC, Phillips DL, Semmens BX. 2018. Analyzing mixing systems using a new generation of Bayesian tracer mixing models. PeerJ 6:e5096 <https://doi.org/10.7717/peerj.5096>

## Installation

The GUI has been removed from the CRAN version of MixSIAR (if desired, see [MixSIARgui](#) on GitHub). Running MixSIAR with scripts is easier to install and better for repeated analysis.

## Links

Download from CRAN at <https://cloud.r-project.org/package=MixSIAR>

Browse source code at <https://github.com/brianstock/MixSIAR/>

Report a bug at <https://github.com/brianstock/MixSIAR/issues>

## License

GPL-3

## Citation

[Citing MixSIAR](#)

## Developers

Brian Stock  
Maintainer, author

Brice Semmens  
Author

[All authors...](#)

## Dev status

CRAN 3.1.12

downloads 48K

DOI 10.5281/zenodo.1209993

## 5. GitHub site + resources

# GitHub Issues page

<https://github.com/brianstock/MixSIAR/issues/>

brianstock / MixSIAR Public

Unpin Unwatch 29 Fork 71 Star 63

<> Code Issues 99 Pull requests Discussions Actions Projects Wiki Security Insights Settings

Filters  Labels 5 Milestones 0 New issue

☐ 99 Open ✓ 231 Closed Author Label Projects Milestones Assignee Sort

- ☐ **Is it possible to define the order of categorical covariates?** 3  
#337 opened 2 weeks ago by chiating9283
- ☐ **include sampling precision error into MixSIAR model** 1  
#336 opened 2 weeks ago by jesmith5
- ☐ **Continuous effect output options** 1  
#334 opened last month by Elissavet182
- ☐ **Extract posterior estimates?** 1  
#333 opened on Feb 13 by bethanybethke2050
- ☐ **Warning message concerning deprecation of `..scaled..` to use `'after_stat(scaled)'` instead.**  
#332 opened on Jan 27 by trudgingon
- ☐ **Modify output posterior continuous effect**  
#330 opened on Jan 20 by josefernando876
- ☐ **How do you add more models to jags.mod without running every model over each time?** 1  
#329 opened on Jan 13 by naalipalo

## 5. GitHub site + resources

# GitHub Issues page

The screenshot shows a GitHub issue page for issue #80, titled "Accessing posterior chains with attach.jags function #80". The issue is marked as "Closed" with a red label. The issue was created by brianstock 18 days ago. The issue description is "Fixed now." and "Data files and script: #100". The issue is linked to commit 91cf880, which fixes issue #101. The issue is also linked to commit 91cf880, which fixes issue #101. The issue is also linked to commit 91cf880, which fixes issue #101.

brianstock added the **bug** label 19 days ago

brianstock added a commit that referenced this issue 18 days ago

fixes issue #101 bug with 2FE or 1FE + 1RE 91cf880

brianstock referenced this issue 18 days ago

**Accessing posterior chains with attach.jags function #80** Closed

brianstock commented 18 days ago Owner + 😊 ✎ ✕

Fixed now.

Data files and script: [#100](#)

With 1 fixed + 1 random effect, the proper proportions to use are created by combining the global intercept, the factor1 offset, and the factor2 offset:

```
1. ilr.both[,f1,f2,src] = ilr.global[,src] + ilr.fac1[,f1,src] + ilr.fac2[,f2,src]
```