Flexible Models and New Algorithms for Fitting Stable Isotope Mixing Models

A dissertation submitted for the degree of
Doctor of Philosophy

By:

Emma Govan

Under the supervision of:

Prof Andrew C. Parnell

Prof Andrew L. Jackson

Hamilton Institute, Insight Centre for Data Analytics
Maynooth University

October 2024

In memory of my Dad, John Govan.
You're here in my heart.

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education.

Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

The thesis work was conducted from August 2020 to October 2024 under the supervision of Prof. Andrew C. Parnell in the Hamilton Institute, Maynooth University and Prof Andrew L. Jackson in the Zoology Dept, Trinity College Dublin.

Emma Govan

Maynooth, Ireland,

October 2024.

Sponsor

This work was supported by an Insight Science Foundation Ireland grant number SFI/12/RC/2289_P2_Parnell

Insight

SFI RESEARCH CENTRE FOR DATA ANALYTICS

Collaborations

Andrew C. Parnell: As my supervisor, Prof. Parnell (Maynooth University) supervised and collaborated on the work of all chapters.

Andrew L. Jackson: As my external supervisor, Prof. Jackson (Trinity College Dublin) supervised and collaborated on the work of all chapters.

Stuart Bearhop: Prof Bearhop (University of Exeter) collaborated on the work in Chapters 3 and 4.

Richard Inger: Dr Inger (University of Exeter) collaborated on the work in Chapters 3 and 4.

Brian C Stock: Dr Stock (Institute of Marine Research, Norway) collaborated on the work in Chapter 4.

Brice X Semmens: Prof Semmens (Scripps Institution of Oceanography, University of California) collaborated on the work in Chapter 4.

Eric J Ward: Dr Ward (Northwest Fisheries Science Center, NOAA Fisheries) collaborated on the work in Chapter 4.

Alan Inglis: Dr Inglis (Hamilton Institute, Maynooth University) collaborated on the package that forms the basis of Chapter 4.

Ahmed Shalaby: Mr Shalaby (Hamilton Institute, Maynooth University) collaborated on the package that forms the basis of Chapter 4.

Publications

The chapters contained in this thesis have submitted to peer-reviewed journals. Chapter 3 is currently being prepared for submission to *Journal of Statistical Software*. Chapter 4 has been submitted to *Methods in Ecology and Evolution* and is currently under peer review. Chapter 5 is being prepared for submission.

Submitted articles (under review):

- Govan, E., Jackson, A. L., Bearhop, S., Inger, R., Stock, B. C., Semmens, B. X., Ward, E. J., and Parnell, A. C. (2024). cosimmr: an R package for fast fitting of Stable Isotope Mixing Models with covariates. arXiv preprint arXiv:2408.17230.

Articles in preparation:

- Govan, E., Jackson, A. L., Inger, R., Bearhop, S., and Parnell, A. C. (2023). simmr: A package for fitting Stable Isotope Mixing Models in R. arXiv preprint arXiv:2306.07817.
- Govan, E., and Parnell, A. C. (2024) cosimmrSTAN: an R package for running of Stable Isotope Mixing Models using STAN.

Contents

Abstract

Stable Isotope Mixing Models (SIMMs) are important for ecologists. They allow for the study of animal diets via measurement of biologically relevant stable isotopes. These measurements can be used to estimate the contribution of different food sources to an animals diet. Knowledge of an animals diet is important when we wish to conserve species, as we need to know what food they rely on. Knowledge of an animals diet can be used to quantify an animals niche and to assess competition between species. SIMMs are also widely used in studies on pollution and air quality, where they may be referred to as ‘source apportionment’, ‘end member analysis’, or ‘mass balance analysis’ models.

However, SIMMs are currently mainly run using Markov chain Monte Carlo (MCMC), which, while guaranteed to converge, can be prohibitively slow, requiring millions of iterations in order to reach convergence if the model is complex. In this thesis we have developed new tools for running SIMMs via Variational Bayes. This allows for a speed improvement ranging between two and one hundred times when compared to MCMC while still obtaining comparable results.

The work in this thesis is divided into 3 chapters, each focusing on a different R package. Separate R packages were implemented for ease-of-use for non-expert users as well as to allow past work using these packages to be replicable in future. The packages are all designed for ecologists and individuals without a robust statistical background, with detailed vignettes and examples included in the packages.

Chapter 3 in this thesis focuses on `simmr`, an R package for running SIMMs. `simmr` allows users to choose between running models through MCMC or through Fixed

Form Variational Bayes. `simmr` is designed for ease of use for non-expert users and has built-in plotting and summary functions.

Chapter 4 in this thesis focuses on `cosimmr`, an R package developed for running SIMMs with fixed covariates included. This package is developed using Variational Bayes and offers up to a one order of magnitude speed improvement over other packages. `cosimmr` has built-in predict and plotting functions and allows for users to easily visualise their results.

Chapter 5 focuses on `cosimmrSTAN`, an R package developed which utilises STANs Variational Bayes functionality in order to run complex SIMMs with fixed and/or random effects, as well as allowing the hierarchical fitting of food sources or the use of raw source data. `cosimmrSTAN` offers between 70-100 times speed improvement over other packages.

These speed improvements mean that ecologists can use SIMMs more easily, with accessible packages and quicker turnaround for results. This also means model comparison becomes more accessible, with users able to run multiple models quickly and compare results between them in order to make better informed decisions about covariate inclusion. Ultimately, use of these packages will allow for more comprehensive analyses of animal diets, and will allow users to gain insights into species' role in the ecosystem.

Acknowledgements

To my supervisor Andrew Parnell, thank you for your endless support, kindness, advice and enthusiasm. I feel very lucky to have gotten to work with you for these last few years. Your kindness and care when my dad was ill is something I'll never forget, and I will never be able to thank you enough for your compassion, as well as your guidance throughout this process. Thank you for everything.

To my supervisor Andrew Jackson, thank you for introducing me to theoretical ecology, for all of your support and advice, and for answering my many questions with patience and kindness. If it wasn't for your lectures I never would have become a zoologist. You've changed my life. Thank you.

To Shaun, thank you for listening, for making me laugh, for keeby, for cat photos, and for being the best person to watch football with. I cannot wait for many more adventures. Do do do do do do do do do. Love you so much and I don't know what I would do without you.

To Mammy and Daddy, for your endless support, for listening, for being there day and night, for instilling in me a love of science, and for always encouraging my curiosity (and answering my endless bank of questions!!), thank you. Love you both more than words can say. Daddy, we'll miss you forever.

To Nell, words cannot express how much you have done for all of us. You are an inspiration. Thank you for all the lit candles, for your encouragement, and for your endless support. I love you so much.

To Roisin, Mick, Mary, and Jim - thank you all for everything you've done for us all, for all the lit candles, for always being there. I love you all so much.

To Connor, Josie, and Gráinne - thank you for always being there. Love you all so much.

To my grandparents - thank you for always being there, for the many dinners, and for all your support. Love you.

To Jordyn - thank you for your endless patience and kindness, for listening to me no matter the topic, for providing the soundtrack to the last 4 years, I will be forever grateful you are in my life, I log6e you so much.

To Taari, thank you for always being there (despite many timezones separating us), for listening to me talk about every topic under the sun, and for photos every time you visit an airport. Love you lots.

To Dan, Micah and Mark - you all make my life weirder and I love you for it. SWC forever.

To Shona (and Flo!), Jody, James, Deirdre, Rigs, Dan C, Tamsin, Ash, Kat, Parrot, Kiki, Ed, Dani, Grundy, Emi, Leslie, Lyssa, Paul, Laura, Erin, Bri, Heather, Helen, Kathy, and the whole SUYS crew - I feel so lucky to have you all in my life, I love you all so very much. Shakespeare, a pair of pyjamas, and a mutton chop.

To Luke - for being the best lab partner, for your endless stories, and for always being there. Thank you.

To Sarah, Wiktoria, Rachel and Anne - here's to many more Caps visits. Love you all.

To everyone in the Hamilton - this has been the most lovely place to come into every day and it's thanks to the people who make it so wonderful. A special thank you to Maeve, Dáire, André, Alan, Ahmed, Nahia, Rachel, Cormac, Conor, YC, Marina, Eleni, Enola, Nathan, Shauna, Akash, Samara, Agustina, Anthony, Dara, Anna, Aoife, and Hannah, for help, advice, support, and endless laughs during lunch time.

To Kate and Rosemary - I would have been lost without you both, thank you for your endless patience and kindness.

And finally, to Jurgen Klopp, to Arne Slot, and to Liverpool FC - thank you for helping me stay sane, for always giving me something to look forward to, and for blessing me with James Milner. YNWA.

List of Figures

List of Tables

CHAPTER 1

Introduction

In this introduction, we discuss the motivations behind the work presented in this thesis and provide an overview of the material discussed in the following chapters.

1.1 Motivation

Knowledge of an animal's diet is essential if we wish to conserve a species, or if we wish to know how that species is interacting in its ecosystem. Climate change has been shown to have an impact on habitats and food web structure (Lurgi et al., 2012; Albouy et al., 2014; Thompson et al., 2023), and human activities are recognised as a main driver of climate change (Lee et al., 2023). Food webs and the interactions between species in these habitats are changing due to the impact of humans. The extinction of species can drive changes in species interaction and in the diets of species. While it is important to know about the food web and trophic interactions in a habitat in order to manage it and conserve species, these interactions can often be complex (Polis, 1991) and difficult to observe and quantify.

Instead, isotopic data may be used in order to estimate the proportional contribution that different foods make to an animal's diet, without needing prolonged direct observation or human interaction. Isotopic measurements can also be useful when looking at the niche of an animal. Hutchinson (1957) described the niche as an 'n-dimensional hypervolume', and so, by looking instead at the isotopic niche (which is shown to correlate with the niche; Newsome et al., 2007), we can compare species over time and see when invasive species are occupying a similar niche to (and thus competing with) native species.

Stable Isotope Mixing Models (SIMMs) are commonly used in ecology to study the proportional contribution that different sources make to an animal's diet. SIMMs are widely used and cited, and are a useful way to learn information about a system without having to observe the habitat for a long time, as this risks impacting the behaviour of the organisms present. Stable isotopes are useful for studying animal diets as they can provide information about trophic level (DeNiro and Epstein, 1978; Minagawa and Wada, 1984), or indicate marine or terrestrial origin of sources (Peterson and Fry, 1987). Some recent examples where SIMMs have been used include Pérez-Ramallo et al. (2024), where stable isotopes were used to examine the diets of members of an early religious order in 12th to 15th Century Spain, concluding that the order were likely members of the nobility or elite due to the food they had access to; Thibault et al. (2024), where `simmr` (Chapter 3) was used to look at the diet of Dugongs (*Dugong dugon*) and how diet varies between male and females; and Lipscombe et al. (2024), where a seasonal shift in the diet of immature white sharks (*Carcharodon carcharias*) was observed.

Markov chain Monte Carlo (MCMC) is a popular method for running SIMMs. MCMC is a sampling algorithm and while it is always guaranteed to converge, it can take millions of iterations to do so. This computational limitation means that SIMMs can be slow to run, especially if the model is more complex. This forms the motivation for this thesis: we aim to implement SIMMs using Variational Bayes. Variational Bayes is an optimisation-based technique and can therefore offer a speed improvement over sampling-based methods. In this thesis we use three different versions of Variational Bayes: Fixed Form Variational Bayes (Salimans and Knowles, 2013), Variational Bayes with Cholesky Decomposed Variance (Titsias and Lázaro-Gredilla, 2014; Tan and Nott, 2018), and Automatic Differentiation Variational Inference (Kucukelbir et al., 2015). Generally, Variational Bayes works by specifying a variational posterior, and then minimising the Kullback-Leibler

(Kullback and Leibler, 1951) divergence between the true posterior and the approximate variational posterior. Throughout this thesis we use several different types of Variational Bayes, which involve slightly different variational posteriors. The full description of these models is provided in sections 3.A and 4.A.

1.2 Thesis Outline

The primary aim of this thesis is to demonstrate the use of Variational Bayes (VB) in speeding up the fitting of SIMMs. We present VB as an alternative to Markov chain Monte Carlo (MCMC), which, while guaranteed to converge, can be prohibitively slow. In this thesis we show that Variational Bayes obtains comparable results to MCMC, but in a fraction of the time. This allows for users to speed up model fitting, as well as offering the option to fit multiple models with different covariates to investigate the best-fitting model for their system. Secondly, we aim to fit more complex SIMMs by allowing for users to include fixed or random covariates, or a combination of the two. This allows for users to gain a deeper understanding into the system that they are studying, by allowing for them to see how the proportion that sources contribute to mixtures change across different measured covariates. While software is available for fitting SIMMs with random or fixed covariates, it uses MCMC which can be impractically slow for users. We show between a 2- to 100-fold improvement in speed over current software when fitting these more complex models. We also offer users the options of selecting different error terms, allowing for users to customise their model if they wish. If covariates are relevant to a model then excluding them violates the assumption that our data are independent and means we do not get an accurate model for our data. Thirdly, throughout this thesis we focus on making software accessible for non-expert users. Built-in predict and plotting functions in the packages developed throughout this thesis means that insights about covariates and how the proportions that sources contribute to mixtures change across covariates are easily accessible for users. Vignettes designed to accompany the packages described in this thesis allows for users to easily run their own datasets through the packages developed.

This thesis is organised as follows: In Chapter 2 we discuss the statistical and biological background to SIMMs, describing the main statistical model used as the basis for the chapters that follow, as well as the biological assumptions made when running these models. We also outline some of the most common terminology used in SIMMs.

In Chapter 3 we implement a standard SIMM using Variational Bayes. This is implemented as an R package called `simmr`. The R package is designed for ease-of-use, so users do not need to know the intricacies of Variational Bayes in order to use it. `simmr` also allows for users to run SIMMs using MCMC, and this chapter uses the geese data from Inger et al. (2006) as a case study to illustrate the use of `simmr`. This chapter highlights the useful functions within the `simmr` package as well as the package's ease-of-use and the fact that FFVB produces equivalent results to MCMC in a shorter timeframe.

Chapter 4 discusses the R package `cosimmr`. Again this package is designed for ease-of-use and users do not need to know specific details about the VB algorithm in order to be able to run it. This package specifically uses Variational Bayes with Cholesky Decomposed Variance and the algorithm is discussed in the chapter. This package allows for users to include covariates as fixed effects and has a function developed to allow users to predict proportions for covariates not present in the original data set. We use a simulated case study to illustrate the functioning of the package as well as three case studies based on real data. These three studies are also carried out in MixSIAR, an R package which allows for the inclusion of covariates but is based around Markov chain Monte Carlo. Using these 3 studies we show that VB allows for models to run in a much faster timeframe (offering up to 1 order of magnitude speed increase over MCMC).

In Chapter 5 we present `cosimmrSTAN`, which makes use of STAN's (Carpenter et al., 2017) VB functionality in order to run SIMMs with either fixed or random effects. This package also allows users to fit sources hierarchically or to supply raw source data if they wish. This addition allows for users to run more complex models, while the use of VB ensures that the models can be run quickly. This chapter also uses three case studies run through both `cosimmrSTAN` and MixSIAR, to highlight the speed improvement conferred by the use of Variational Bayes, and we see between a 70-100 times speed improvement when using `cosimmrSTAN`, while obtaining comparable results.

Finally, in Chapter 6 we conclude this thesis and remark on limitations within these chapters as well as potential avenues for future expansion of this work. All proposed methods in this thesis were implemented using the R (R Core Team, 2021) software and are accessible on Github via three public repositories. The repositories https://github.com/emmagovan/simmr_paper_SIMM_package_scripts, https://github.com/emmagovan/cosimmr_paper, and https://github.com/emmagovan/cosimmrSTAN_paper are related to Chapters 3, 4, and 5, respectively. Within these repositories we have made available R scripts required to produce all analyses and plots presented in this thesis. Additionally, all datasets are publicly available via the above R packages. simmr and cosimmr are available on CRAN at the following urls: https://cran.r-project.org/web/packages/simmr/index.html and https://cran.r-project.org/web/packages/cosimmr/index.html and cosimmrSTAN is available on Github: https://github.com/emmagovan/cosimmrSTAN.

CHAPTER 2

A brief introduction to Stable Isotope Mixing Models

In this chapter, we provide a brief introduction to Stable Isotope Mixing Models, explaining the statistical and biological assumptions, and some of the common terminology that will feature in the chapters that follow.

2.1 Introduction: What is a SIMM?

Stable Isotope Mixing Models (SIMMs) are commonly used in the study of animal diets (Phillips, 2001) and utilise stable isotopes to calculate how different food sources contribute to the diet of an animal, the idea being that the tissues in an animal are a proportional combination of the foods that they eat, or equivalently "you are what you eat" (DeNiro and Epstein, 1978). It is important if we wish to conserve species that we know what foods they rely on. Knowledge of an animal's diet is important when studying invasive species (Vander Zanden et al., 1999). If we note that an invasive species has a very similar diet to a native species, then we can hypothesise that these species will compete (due to the Competitive Exclusion Principle; Hardin, 1960) and the invasive species may drive the native species to extinction (Webb et al., 2002) or they may co-exist, if one species changes its behaviour (Jackson and Britton, 2014). Changes in food-web structure can be indirectly observed through isotopes (Schmidt et al., 2007). A change in location in iso-space can indicate that a species' behaviour is changing (Jackson et al., 2012). Iso-space is a term used throughout this thesis. It is a way of thinking about isotopic data - where we think of an n-dimensional space with each axis representing an isotopic ratio, and consumers can be located at a point in this space, based on their isotopic values. Consumers close in iso-space have similar isotopic values and therefore likely have similar diets. Iso-space can be visualised by creating iso-space plots, which are shown throughout this thesis. In Figure 2.1 we can see a 2-isotope iso-space plot (also referred to as a Tracers plot) which also shows the mixing polygon drawn in red. If we wish to run a SIMM then our mixtures should all lie within the mixing polygon which is created by joining the outermost points of each source. If our mixtures do not all lie within this mixing polygon then it is an indicator that there is an issue with our data. The data in this example are explained and run through a SIMM in Section 2.8.

Mixing Models allow users to see how different sources contribute to a 'mixture'. Generally throughout this thesis we will be using animals as 'mixtures' as we wish to know how different foods contribute to the animal's diet. The tracers we use in order to see these contributions are usually stable isotopes of biologically relevant elements, hence the name Stable Isotope Mixing Models, but other tracers can be used, such as fatty acids. 'Mixtures' are also referred to as 'consumers'. The food eaten is more generally referred to as 'sources'. SIMMs can be used in the study of pollution and air quality, as well as in geological contexts. SIMMs can be referred to as 'source apportionment models' (Hopke, 1991), 'end member analysis' (Hooper et al., 1990), or 'mass balance analysis' (Miller et al., 1972) depending on the context and field in which they are being used. SIMMs are widely used across these different fields and software for running SIMMs are very popular and widely downloaded.

2.2 Stable Isotopes

Stable Isotopes are commonly used in mixing models when examining animal diets, or more generally, the composition of mixtures. Isotopes are atoms with a different

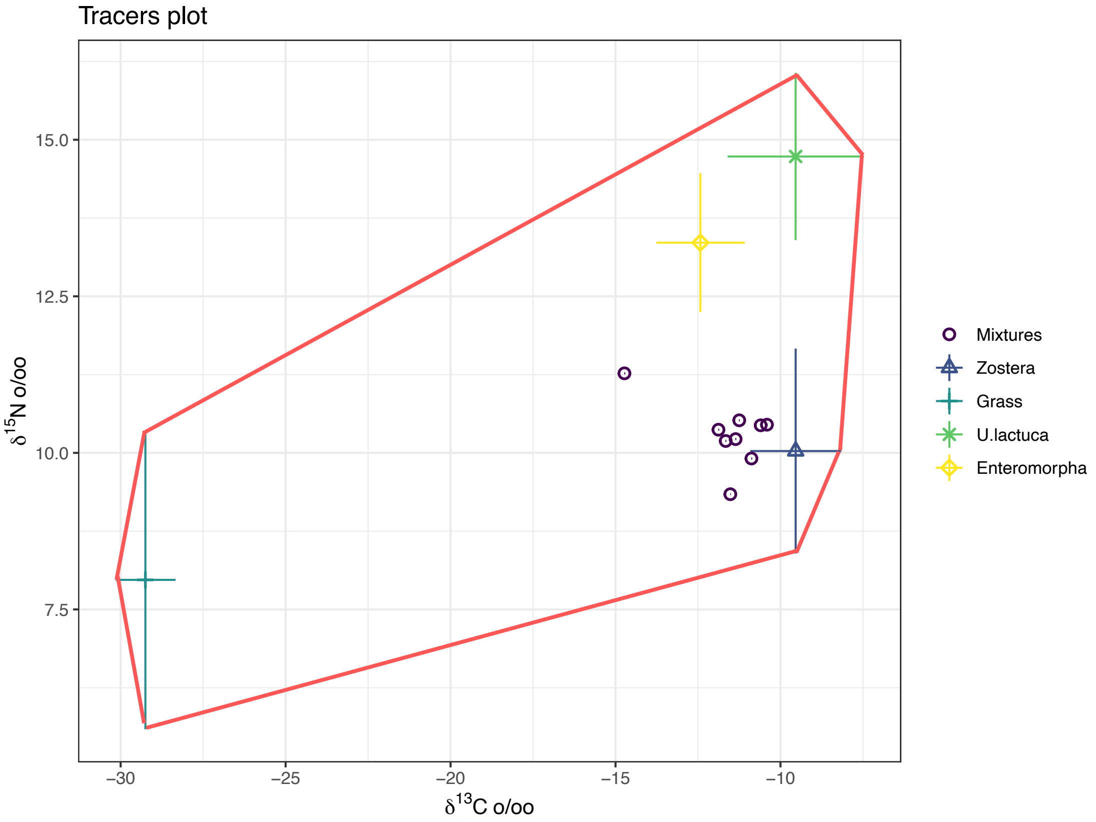


Figure 2.1: Iso-space plot showing measurements for Carbon-13 on the x-axis and Nitrogen-15 on the y-axis. The mixing polygon is drawn in red.

number of neutrons than the standard element. Stable isotopes are called stable as they do not radioactively decay. Isotopes can be incorporated into tissues in different ratios (relative to the 'standard' element) (Farquhar et al., 1989), and this is why they are useful for looking at biological systems. The most common isotopes used in SIMMs are: Carbon-13; and Nitrogen-15. These isotopes can indicate if something is of a marine or a terrestrial origin: marine plants get carbon/nitrogen from water, whereas terrestrial plants obtain their nitrogen from soil and carbon from air (Peterson and Fry, 1987). Carbon and Nitrogen can show if sources are from different trophic levels, because of enrichment up the food chain due to preferential retention of one isotope over another (DeNiro and Epstein, 1978; Minagawa and Wada, 1984). Other isotopes can be used, for example: hydrogen can indicate autochthonous (energy is coming from within the ecosystem) vs allochthonous (energy is coming from materials imported into the ecosystem) zones in freshwater systems; coupled hydrogen and oxygen measurements are often used (Vander Zanden et al., 2016) as the ratios of these elements can often be linked, and so it is less intensive to measure them together. Sulphur isotopes are used, for example in the study of the Caribbean spiny lobster, *Panulirus argus* (Higgs et al., 2016). In this case sulphur was used as the lobsters were known to feed on clams that host chemoautotrophic bacteria which oxidise sulphur (Caro et al., 2009), and so this food source had lower sulphur relative to the other samples. Strontium can be used in indicating the ages of rocks (Newsome et al., 2007).

Hutchinson (1957) refers to 'the niche' as an "n-dimensional hypervolume" where every axis refers to a measurable environmental variable, such as temperature, latitude, altitude, humidity and so on. However this definition, whilst very important in ecology, is impossible to quantify. Instead, isotopic niche can be used as a proxy (Newsome et al., 2007) and can be used for comparing niche width (Bearhop et al., 2004). It is shown that isotopic variation reflects variation in the diet of consumers (Arnoldi et al., 2023). Use of isotopes allows us to quantify the niche in a way that is not possible otherwise.

Isotopes are generally standardised and represented using δ notation as parts-per-mille (δ). The equation for $\delta^H X$ is as follows (Fry, 2006):

$$\delta^H X = \left(\frac{R_{sample}}{R_{standard}} - 1\right) \times 1000$$

where :

- X = the element in question
- H = the mass of the heavy isotope of that element
- R_{sample} = the ratio of the heavy to the light isotope for the element in the sample we are interested in
- $R_{standard}$ = the standard measurement of the ratio of the heavy to the light isotope for the element

Iso-space plots are a standard tool in SIMMs. Each axis represents a tracer (most commonly an isotope measured in parts-per-mille as stated above) and both the mixture and sources can be plotted. These plots are useful as tools before running SIMMs as a consumer must lie within the polygon created by joining the outermost points of each source in order for proportions to be able to be estimated. A 1-isotope iso-space plot can be seen in Figure 2.2. For this example our mixtures need to lie between the leftmost error bar of A and the rightmost error bar of C. The error bars represent the error in the source measurement. A two-isotope example can be seen in Section 2.8 in Figure 2.3.

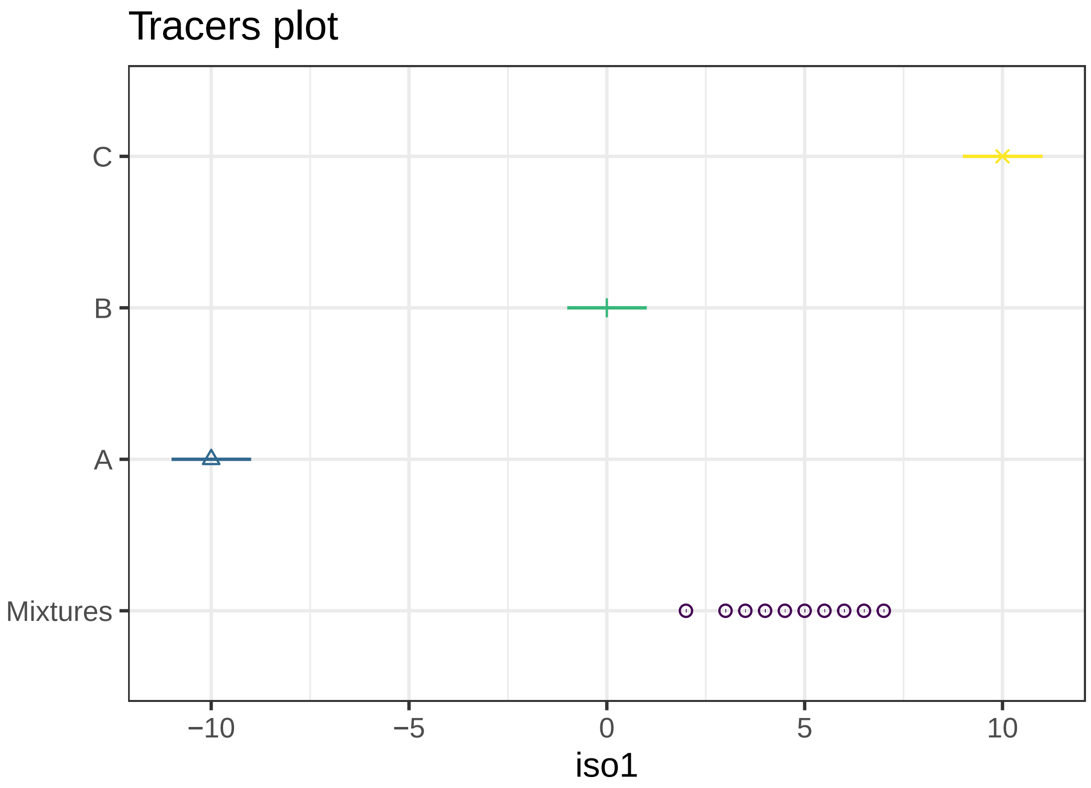


Figure 2.2: Iso-space plot showing measurements for 1 isotope. Food sources A, B, and C are plotted as well as the mixtures.

Individuals are represented by points on this plot. They represent a single measurement of an individual. These individuals are members of the same species and the aim is to find which sources are the main contributors to their diet. Sources are

	mean	sd
deviance	40.466	3.481
A	0.149	0.068
B	0.245	0.135
C	0.606	0.085
sd[iso1]	2.103	0.788

Table 2.1: Results obtained showing mean and standard deviation estimates for each food source as well as the error on iso1 for the 1-isotope example.

sampled multiple times and are therefore represented by bars to display the error. This example is run through a SIMM in Chapter 3 but the results are displayed below in Table 2.1:

We can see both from the results and from the iso-space plot that C makes up the majority of the diet in this case - our data points lie close to C in iso-space. We can also see that we are most unsure about how much of source B they are eating - the points in iso-space could be brought leftwards by consuming A, or by consuming B.

2.3 The Statistical Model behind SIMMs

2.3.1 The simplest model

Our aim when running SIMMs is to estimate the proportion of each source k (of K total sources) that are contributing to our mixtures. The data gathered is y_{ij} which is the isotope value of individual i on isotope j. Each sample might be collected via blood, or from other tissues, depending on the time frame we wish to look at. We assume that y_{ij}, conditional on the parameters, follows a normal distribution set out as follows:

$$y_{ij} \sim N\left(\sum_{k=1}^{K} p_k s_{ikj}, \sigma_j^2\right)$$

where:

- y_{ij} = tracer value of individual i on tracer j
- p_k = proportion of source k in the diet (of K total sources)

- s_{ikj} = source value of tracer j for individual i eating source k

Our aim is to estimate p_k and its uncertainty. We want to make this model more biologically accurate, as well as marginalising over s (Moore and Semmens, 2008) in order to make it computationally less intensive. It is common to assume $s_{ijk} \sim N(\mu_{s,jk}, \sigma^2_{s,jk})$, where s_{ijk} is an individual level random effect. The biological considerations we must make are in relation to Trophic Discrimination Factors and Concentration Dependence.

2.3.2 Trophic Discrimination Factors (TDFs)

Trophic Discrimination Factors account for the fact that when an animal consumes and assimilates the food into its own tissues, the processes involved in this will affect the isotopic ratio in the tissue of consumers. 'Heavy' isotopes (that is, isotopes with more neutrons) may be lost more or less than 'light' versions, depending on the metabolic process occurring (Inger and Bearhop, 2008). For example, Nitrogen-15 is preferentially retained over 'standard' nitrogen when proteinous wastes are excreted by animals (Minagawa and Wada, 1984), and similarly Carbon-13 is enriched due to respiration (DeNiro and Epstein, 1978). Thus a TDF correction factor is used to adjust the isotopic values up or down to correct for this change. TDFs can be applied to the mixture values themselves in the isospace plot, or to the food sources. In Figure 2.2 the former is equivalent to shifting the mixture values left or right, and the latter to shifting the food sources (A, B or C) left or right. Since the latter is more flexible, allowing for different food sources to have differing TDF values, we apply this approach to the models and software used throughout this thesis.

Finding suitable TDF values to use can be a challenging problem. A common but expensive method is to calculate them in the lab. The idea is to feed an animal a diet of known isotopic signature and allow their tissues to come to equilibrium. If the TDF was zero and they used all the food solely in tissue formation, then we would expect their tissue to have the exact same isotopic signature as the food. Instead, they will be offset from the isotopic signature of the food, and this offset gives us the TDF. However, this method is often not practical depending on the species you are dealing with. Instead, an examination of the literature is often carried out and TDF values of related species are used. SIDER (Healy et al., 2018) is an R package developed with an inbuilt dataset, allowing users to input their species and, based on the phylogenetic relatedness, obtain TDF values for their species. Because TDFs usually account for biological activity, they are less relevant when using SIMMs to study pollution or other non-biotic systems.

2.3.3 Concentration Dependence

Concentration dependence removes the assumption that a source contributes to each tracer equally (Phillips and Koch, 2002). A source can be particularly rich in one tracer and lacking in another, so concentration dependence assumes that a sources contribution is proportional to the product of the contributed mass and the elemental concentration of that source. It can also be the case that different tissues within an organism contain different levels of each tracer. As an example, Ben-David and Schell (2001) note that lean beef tissue and beef fat have similar levels of $\delta^{15}N$, but quite different $\delta^{13}C$ levels. In this case, depending on the tracer and tissue sample taken, we could obtain relatively different estimates of the animals diet. Dietary proteins or lipids might be preferentially routed to synthesis of bodily proteins or lipids respectively, and if bodily proteins are taken as the sample, this would result in dietary proteins being over-emphasised in the diet estimation (Phillips and Koch, 2002).

2.4 A more biologically accurate model

We obtain a more biologically accurate model by accounting for both TDFs and concentration dependence, as well as a model that is computationally less intensive, by marginalising over s (Moore and Semmens, 2008).

$$y_{ij} \sim N\left(\frac{\sum_{k=1}^{K} p_k q_{kj}(\mu_{skj} + \mu_{ckj})}{\sum_{k=1}^{K} p_k q_{kj}}, \frac{\sum_{k=1}^{K} p_k^2 q_{kj}^2(\sigma_{skj}^2 + \sigma_{ckj}^2)}{\sum_{k=1}^{K} p_k^2 q_{kj}^2} + \sigma_j^2\right) \tag{2.1}$$

where:

- y_{ij} = tracer value of individual i on tracer j

- p_k = proportion of source k
- q_{kj} = concentration dependence for source k on tracer j
- μ_{skj} = mean isotopic value of source k on tracer j
- μ_{ckj} = mean TDF value of source k on tracer j
- σ^2_{skj} = variance of source k on tracer j
- σ^2_{ckj} = variance of TDF of source k on tracer j

Again, in this model, our aim is to estimate p and σ_j. $q_{kj}, \mu_{skj}, \mu_{ckj}, \sigma^2_{skj}$ and σ^2_{ckj} are known.

p_k here represents the proportion of each source k eaten, and is constrained so $\sum_k p_{ik} = 1$. We use a Centralised Log-Ratio (CLR; Aitchison (1986)) link function on p so that:

$$[p_1, ..., p_K] = \left[\frac{\exp(f_1)}{\sum_S \exp(f_s)}, \dots, \frac{\exp(f_K)}{\sum_S \exp(f_s)}\right]$$

We perform this transformation as it is much easier to model f_{ik} because it is unconstrained and we can use a normal prior (so $f_k \sim MVN(\mu_f, \sigma_f)$), which is what is used in Chapter 3, or we can include fixed or random covariates (where $f_{ik} = \mathbf{X}_i\beta_{0k}$ or $f_{ik} = \mathbf{X}_i\beta_{0k} + \mathbf{Z}_i\beta_{1k}$) as we do in Chapters 4 and 5.

2.5 Bayesian Methods

We fit this model in a Bayesian framework. Bayes' Theorem is as follows:

$$p(\theta \mid x) = \frac{p(x \mid \theta) \times p(\theta)}{p(x)}$$

where θ represents parameters, and x represents data. Our aim is to get the posterior probability distribution, which is $p(\theta \mid x)$, or the probability of the parameters given the data. The likelihood is the probability of observing the data x given the parameters θ (or $p(x \mid \theta)$, and the prior $p(\theta)$ represents external knowledge about the parameters. In our case this could represent previous experiments, or information about the diet of the animal that is already known.

The full Bayesian model that we fit for the non-covariate model (Chapter 3) is therefore:

$$\begin{aligned}\pi(p_{ik}, \sigma_j^2 \mid y_{ij}, \mu_{sc,kj}, \sigma^2_{sc,kj}, q_{kj}, \mathbf{X_i}, \mathbf{Z_i}) \propto & \prod_{i=1}^{N}\prod_{j=1}^{J}\prod_{k=1}^{K} \pi(y_{ij} \mid p_{ik}, \mu_{sc,kj}, \sigma^2_{sc,kj}, q_{kj}, \sigma_j^2) \\ & \times \prod_{j=1}^{J} \pi(\sigma_j^2) \\ & \times \prod_{k=1}^{K} \pi(f_k)\end{aligned}$$

where $\mu_{sc,kj} = \mu_{s,kj} + \mu_{c,kj}, \sigma^2_{sc,kj} = \sigma^2_{s,kj} + \sigma^2_{c,kj}$, and q_{kj} are known. We place a prior distribution of $\frac{1}{\sigma_j^2} \sim Ga(a_0, b_0)$, where $a, b = 1$ as default, but this can be adjusted by the user. We set $p(f_k) = MVN(\mu_{f0}, \Sigma_{f0})$ as default. In later chapters we change this to allow for f to account for covariates (See sections 4.2 and 5.2.2). The Gamma prior is weakly informative which ensures the prior doesn't strongly affect the results. Inclusion of an informative prior will alter results and would only be recommended if there is strong prior knowledge.

Markov chain Monte Carlo (MCMC) (implemented via JAGS; Plummer, 2003) is commonly used in the fitting of SIMMs. MCMC is a sampling-based algorithm and thus can be slow, sometimes requiring millions of iterations in order to reach convergence. Instead, throughout this thesis we propose the use of Variational Bayes (VB; also referred to as Variational Inference or Variational Approximation), which is an optimisation-based technique. Our aim generally is to create a posterior distribution $p(\theta \mid y)$, which is proportional to the likelihood times the prior, or $p(y \mid \theta) \times p(\theta)$. With VB, we approximate our posterior distribution with a variational posterior, referred to as $q_\lambda(\theta)$. This usually means we can use a simpler distribution, one that is easier to work with. We then optimise $q_\lambda(\theta)$ by minimising the Kullback-Leibler (KL; Kullback and Leibler, 1951) divergence between $q_\lambda(\theta)$ and $p(\theta \mid y)$. Full details of the VB algorithm are provided in Sections 3.A and 4.A, but briefly, for `simmr` we use Fixed-Form Variational Bayes (Salimans and Knowles, 2013), and we set our variational posteriors as so: $q_\lambda(\theta) = q(f)q(\frac{1}{\sigma_j^2})$, where $q(f) \equiv MVN(\mu_f, \Sigma_f)$ and $q(\frac{1}{\sigma_j^2}) \equiv Ga(a, b)$, and so $\lambda = (\mu_f, \Sigma_f, a, b)$. We then iteratively update λ to minimise the KL divergence.

2.6 Current SIMM software

There have been many different software packages developed for the running of SIMMs. One of the earliest was IsoSource (Phillips and Gregg, 2003). This tested every combination of sources in 1% increments and presented feasible solutions. It did not offer posterior distributions, but instead offered the distribution of possible solutions. Isosource did not account for TDFs nor concentration dependence. For a 1-isotope system and 3 sources, IsoSource presents possible solutions for the equation: $\delta_M = p_A\delta_A + p_B\delta_B + p_C\delta_C$, where δ_M = the isotopic ratio of the mixture, $\delta_{A,\ldots,C}$ = the isotopic ratio for sources A, B, and C, and $p_{A,\ldots,C}$ = proportion of each source A, B, and C consumed.

MixSIR (Moore and Semmens, 2008) used a Bayesian framework to estimate proportions using importance resampling. It did provide users with a posterior distribution. It is implemented in MATLAB (Inc., 2022). Users gained the ability to incorporate prior information using MixSIR. MixSIR also allowed for the inclusion of TDF data. MixSIR did not include the residual error term σ_j included in Equation 2.1. MixSIR operates using a sampling-importance-resampling algorithm (DB, 1988).

SIAR (Parnell et al., 2010) was based around Markov chain Monte Carlo and included a residual error term as well as the ability to incorporate prior information. SIAR allowed for the inclusion of TDFs and concentration dependence. Users were also able to incorporate prior information. It was built as an R package, and used the same model described in Equation 2.1. SIAR is no longer updated and `simmr` (discussed in Chapter 3) was developed as a replacement.

FRUITS (Fernandes et al., 2014) allowed for the inclusion of concentration dependence and TDFs. Prior information can be included by the user. It also uses MCMC for model fitting. FRUITS is implemented using Visual Basic (Balena and Fawcette, 1999) and is available as a stand-alone computer programme.

MixSIAR (Stock et al., 2018) is one of the most recent developments in SIMM software. It is Bayesian, using custom JAGS (Just Another Gibbs Sampler, Plummer, 2003) files for each model run. MixSIAR allows for the inclusion of TDFs, concentration dependence, and fixed and random covariates. It introduced an extra multiplicative error term to account for consumer specialisation, and uses this in place of the residual error σ_j. Alternatively users can just use process error or just residual error. However, MixSIAR can be difficult for novice users and it is very slow when studying complex data sets, which may result in users preferring quicker, simpler models, over more complex, but more accurate models.

2.7 Practical recommendations when running SIMMs

There are some things that are important to consider when using SIMMs. Phillips et al. (2014) provides a guide to these recommendations and considerations that are important, and the most important considerations are summarised in this section. It is important to account for tissue turnover when sampling an mixture and their sources. Different tissues can take different times to turnover, for example the half-life of carbon was found to be 6.4 days in liver and 47.5 days in hair (Tieszen et al., 1983a). The otoliths of Atlantic cod, *Gadus morhua* may provide insights into the lifetime diet of the fish (Radtke et al., 1996). Blood has been shown to provide both short and longer-term information on diet when it is split into plasma and cellular components (Hobson and Clark, 1993). It is important that mixtures and their sources are sampled at the correct time relative to one another (Phillips et al., 2014).

There are some assumptions that are made when running SIMMs - for example it is assumed that the user knows all of the different food sources an animal is eating (Phillips et al., 2014), or more generally that the user knows all sources contributing to a mixture. If these are missing then it can alter the shape of the mixing polygon and could result in inaccurate results. It can happen that sources lie close together in iso-space - for example if they are closely related species of plants. It may be harder for the model to distinguish between these sources and therefore the model may provide a wider estimate of consumption for both sources. Users have the option of combining these sources if they wish. It is generally recommended that the SIMM is run with all sources and they can then be combined a posteriori if the user wishes. If sources are similar and related and so can be grouped as a ‘functional group’ then combining a priori can result in more interpretable results, if there is a sensible biological basis for doing so. Combining sources a posteriori is the general recommendation if there is no good biological reason for grouping (Phillips et al., 2005).

2.8 A Simple Example

To illustrate SIMMs I will use a simple example from Inger et al. (2006). These data are used for illustration in later chapters. This example is based around data for Brent Geese *Branta bernicla hrota*, collected at eight time periods over two winters at Strangford Lough, Northern Ireland. The isospace plot is seen in Figure 2.3. Here we see the isotope values for the geese (at timepoint 1), along with the isotope values (and error bars) for each of the food sources consumed by the geese. We can see that the geese are lying relatively close to *Zostera* spp. in iso-space, so we might speculate that a large proportion of their diet is comprised of *Zostera* spp. at this moment in time.

When we run a SIMM using these data (i.e. by fitting equation 2.1 and estimating p_k), we find out that their diet is approximately 60% *Zostera* spp. We can see an output from `simmr` in Figure 2.4. This shows the estimate on the diagonal for each food source consumed as well as the uncertainty. On the off-diagonals it shows the correlation between different food sources. Sources generally are negatively correlated as if the model estimates that an animal eating more of one food is consistent with the data it has to compensate by reducing the amount consumed of another. In this example we see the highest negative correlation between*Zostera* spp and *Enteromorpha*, as they are close in iso-space. The model cannot distinguish between *Enteromorpha* and *U. lactuca* which results in a lower correlation between the two. We see a low positive correlation between *Zostera* spp and Grass as if an individual is eating one, then they can also be eating the other in order to end up lying between the two sources in iso-space.

Inger et al. (2006) concludes by stating that these data can be used to ensure that when protecting Brent Geese, that grasslands, as well as intertidal areas, are

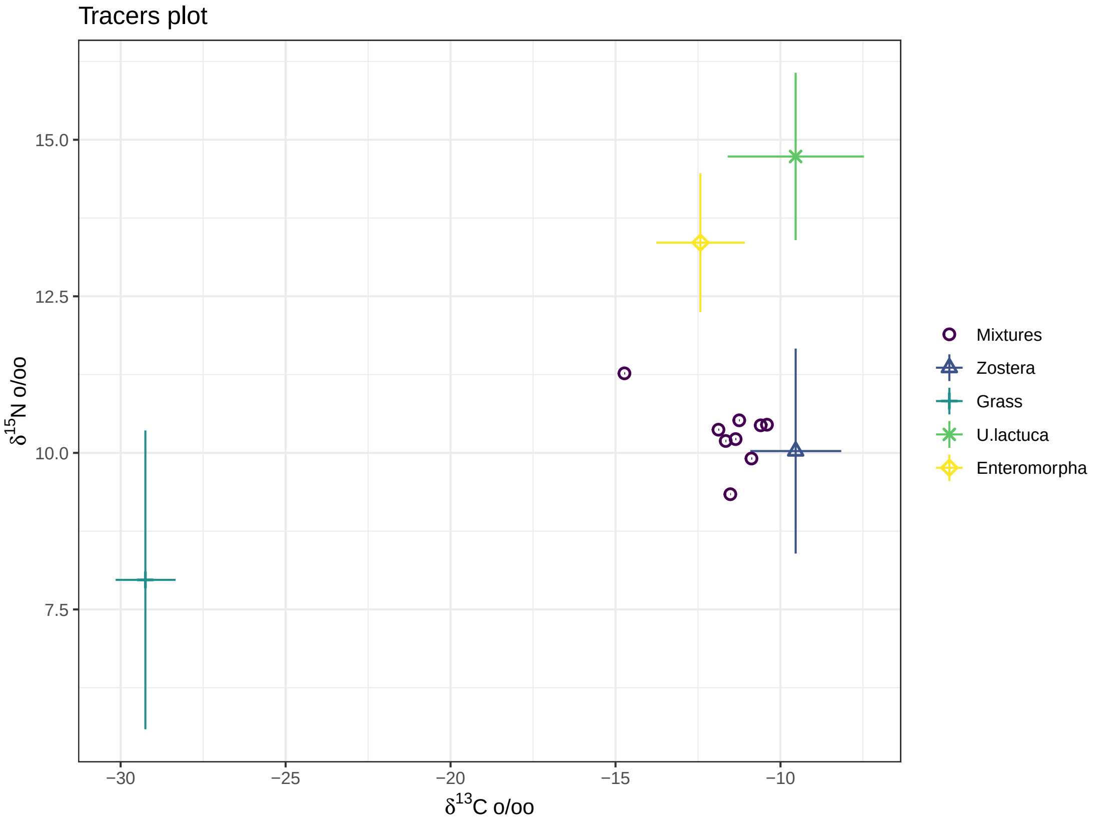


Figure 2.3: Iso-space plot showing Carbon ratio on the x-axis and Nitrogen on the y-axis. Food sources are plotted as well as mixtures.

conserved, as grasslands become more important to the species later in the winter. This highlights the use of SIMMs - knowing what food an animal depends on is essential when we wish to protect that species.

2.9 Summary

In this chapter, we discussed Stable Isotope Mixing Models, and examined the biological and chemical processes underlying these models. We looked at the basic statistical model and how this is expanded to become more biologically accurate, by incorporating TDFs and concentration dependence, as well as being made mathematically simpler. We looked at software that has been developed for running SIMMs, as well as factors that are important for users to consider when running SIMMs. Finally, we looked at the Geese example from Inger et al. (2006), which

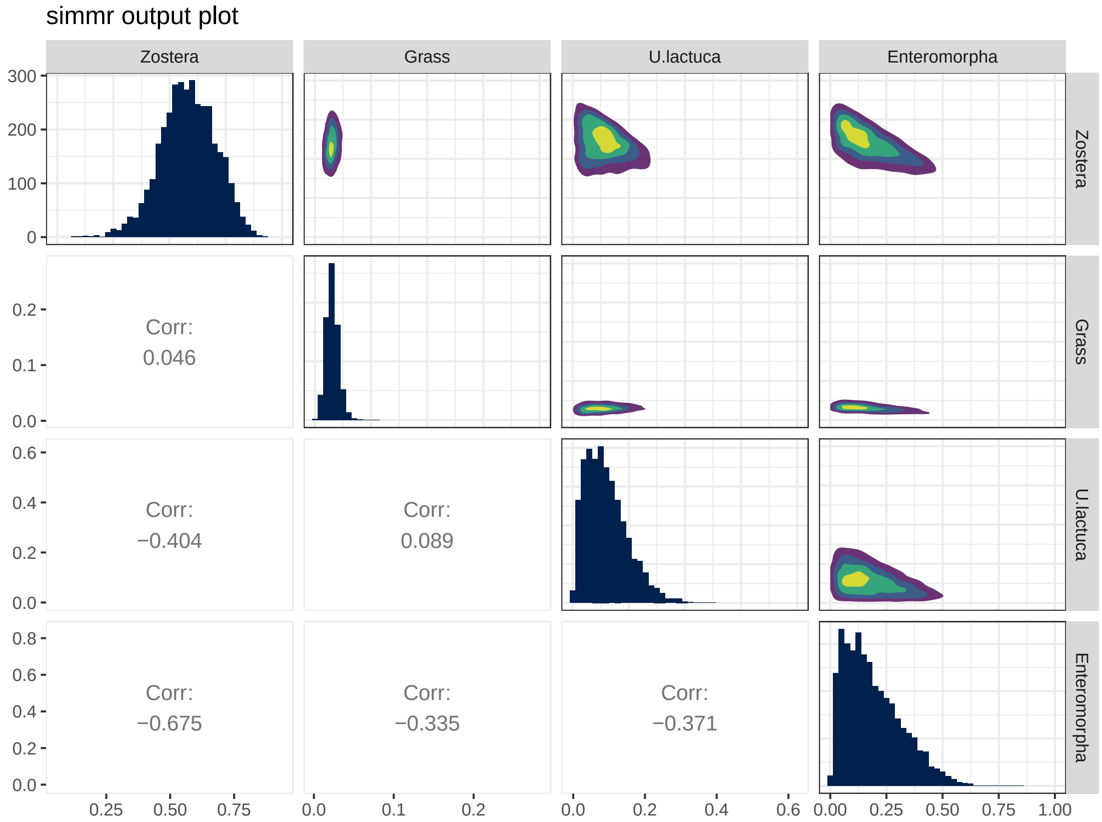


Figure 2.4: Plot created by `simmr` (Chapter 3) showing posterior distributions for p_k on the diagonal, contour plots of the posterior between p_k and p_j on the upper half, and correlation values between p_k and p_j for $k \neq j$ on the lower half.

is used in future chapters to illustrate how the packages within work.

CHAPTER 3

simmr: A package for fitting Stable Isotope Mixing Models in R

We introduce an R package for fitting Stable Isotope Mixing Models via both Markov chain Monte Carlo and Variational Bayes. The package is mainly used for estimating dietary contributions from food sources taken via measurements of stable isotope ratios from animals. It can also be used to estimate proportional contributions of a mixture from known sources, for example apportionment of river sediment, amongst many other use cases. The package contains a simple structure which allows non-expert users to interface with the package, with most of the computational complexity hidden behind the main fitting functions. In this paper we detail the background to these functions and provide case studies on how the package should be used. Further examples are available in the online package vignettes.

3.1 Introduction

Stable Isotope Mixing Models (SIMMs) are a useful tool for ecologists, especially in the reconstruction of animal diets (DeNiro and Epstein, 1978). Starting from stable isotope measurements of animal tissues and their food sources, mixing models allow for estimation of the proportional composition of their food sources in their diet (McKechnie, 2004). Stable isotope ratios represent the difference in relative abundance of non-radiogenic, stable isotopes expressed as the ratio of an isotope's "heavy" form of an element versus a "light" form relative to an internationally accepted standard. These stable isotope ratios can vary geo-spatially and across the different levels of food webs (Hobson, 1999). Isotopic data can be obtained relatively easily and allow for many different aspects of diet to be analysed, for example across timescales or locations, depending on the sampled tissues. Typical ecological applications include quantifying animal diets (Peterson and Fry, 1987), and estimating the origins of migratory animals (Hobson, 1999). Whilst isotope ratios were the original data used for these models, other data are used (see later discussion on end member analysis), so these data are referred to more generically as tracers in the `simmr` package. Similarly, the consumers (usually animals) are referred to more generically as mixtures.

SIMMs require data from both the mixtures being studied as well as all of their sources (for example, the foods they are consuming when we are looking at animal diets). When studying animals, the data can be obtained via tissue samples, such as blood or feathers, depending on the time-frame being studied. For example, isotopes from food are assimilated quickly into tissues with rapid turnover rates such as blood and so this provides a relatively recent estimate of their diet (Tieszen et al., 1983b); metabolically inactive tissues such as feathers or hair preserve an isotopic record of the diet at the time they were grown (Inger and Bearhop, 2008), and samples from otoliths may provide an overview of the diet of a fish over their lifetime (Radtke et al., 1996) with successive layers of the tissue being a record of diet through time. Typically, empiricists will make an assumption that a consumer (the mixture) is at equilibrium with its food sources in order to estimate the dietary proportions at a fixed time point. Similarly, users of mixing models are required to assume that all of the potential food sources have been sampled and included in the model (Phillips et al., 2014). A final parameter that needs to be known or estimated is the trophic discrimination factor (or trophic enrichment factor), which describes the change in isotope ratio between the diet and assimilation into consumer proteins. This can be estimated from captive studies of the same species, literature searches of closely related or functionally similar species or in some cases using the software package `SIDER` (Healy et al., 2018), which uses a Bayesian imputation approach to estimate TDFs of unknown species in relation to known TDF values from a built-in database.

Our paper covers the maths behind the models used for SIMM analysis in the R (R Core Team, 2021) package `simmr`. We demonstrate how to use the package with an example of Brent Geese data from Inger et al. (2006). The package aims to provide a set of powerful tools, but with a simple to use interface which allows beginners to run models sensibly whilst also allowing advanced users full access to all posterior quantities of the back-end Bayesian model.

The basic mathematical equation for a statistical SIMM is:

$$y = \sum_{k=1}^{K} p_k s_k + \epsilon$$

where y is the mixture value, p_k are the proportions associated with source k (of K total sources), s_k is the source tracer value for source k, and ϵ is a residual term. In this over-simplification of the model (see Section 3.4 for a more complete version), s_k is an individual level random effect with a given mean and standard deviation, $s_k \sim N(\mu_{sk}, \sigma^2_{sk})$, and the key task of the model is to estimate the p_k proportions. We call y the mixture value for each individual, but they can also be referred to as the consumer value or end member value in the literature.

There are now a number of software tools for fitting SIMMs, which are discussed in detail in Section 3.3. Whilst these models are mainly used to study the proportional contribution that different foods make up in an animal's diet, SIMMs can also be used in a wide range of different scenarios. These include the study of a Late Pleistocene bear (Mychajliw et al., 2020), which confirmed its trophic position is similar to other bears of the same species, and the study of crop usage in Iron Age settlements (Styring et al., 2022). We expand on this set of examples below.

Models mathematically identical to SIMMs are also used in many other areas. These are often known as 'end member analysis', 'mass balance analysis', or 'source apportionment'. Hopke (1991) is an early review of source apportionment models. It uses linear equations and a least-squares method for running these models. Henry (1997) explores different methods of running these models, such as a geometrical approach. Prior knowledge is then incorporated in Billheimer (2001), which is Bayesian based, and a non-additive error structure is also adopted. Park et al. (2001) incorporates temporal dependence and adopts a Markov chain Monte Carlo (MCMC) approach to estimate parameters. Lingwall et al. (2008) uses a Dirichlet prior distribution to allow flexible specification of prior information. The European Union has published several guides on use of source apportionment with receptor models in studies (e.g., Belis et al., 2019; Mircea et al., 2020).

End member analysis is generally employed by geologists and is used to estimate how different water sources contribute to a mixture. Examples include Soulsby et al. (2003), which employs MCMC methods to study runoff sources during storms in Scotland. Brewer et al. (2011) employs MCMC methods to study runoff sources. Their model allows for consideration of random effects, such as comparison across years. Palmer and Douglas (2008) operates using MCMC and estimates the proportion that different water sources have contributed to sediment samples. Liu et al. (2020) adopts a maximum likelihood method to estimate water sources. Their method employs a multivariate statistical approach to allow for uncertainty in the concentration of end-members, or sources. The end-members contributing to the mixture are first identified and then the proportion that each contributes is calculated. Tao et al. (2021) proposes a maximum distance analysis method that estimates both the number and spectral signatures of end-members, which means that it is not essential to know the number and identity of end-members in advance in order for a model to be run.

Another term commonly employed for SIMMs is that of mass balance modelling. The term is often used for apportioning sources of pollution. Christensen (2004) evaluates several of these methods, for example weighted least squares and the method of moments. In Campodonico et al. (2019) a log-ratio technique is used to analyse how elements move during chemical weathering. Cooper et al. (2014) uses SIMM-related modelling to study suspended particulate matter. They look at several different models and provide comparison and advice on choosing an appropriate model. Code is provided for one of their models.

Our package `simmr` implements mixing models via both Markov chain Monte Carlo (MCMC) algorithms (via JAGS (Just Another Gibbs Sampler, Plummer, 2003)) and faster Fixed Form Variational Bayes (FFVB). It is not designed to be more fully featured than other R packages that fit SIMMs, rather we aim for a simple unified data structure that enables both non-experts and advanced users to access the tools they need. FFVB is introduced as a foundation to enable much faster fitting of SIMMS, where MCMC can be prohibitively slow. The data structure needed for running SIMMs can be complex, using multiple different data frames of different dimensions, but `simmr` makes it easy to read in the data and subsequently create plots and model output. We use a Snake case naming convention for consistency and follow the 'tidyverse' (Wickham et al., 2019) style guide. We aim to keep the number of functions to a minimum and use S3 classes for access to summary and plot commands. Our visualisations are carefully selected for style and colour choices, and easily produced through built-in functions using `ggplot2` (Wickham, 2016). We aim to make all our functions easily extendable so that advanced users can create more complicated outputs.

SIMMs are widely used and applicable in many areas. Thus the R packages for running them are frequently downloaded and have received thousands of citations between them. Figure 3.1 shows the citation rates of the main papers used for SIMMs and the number of downloads of the associated R packages.

3.2 A short guide to fitting SIMMs using simmr

The first step in using `simmr` is to install and load the package:

```
R> install.packages("simmr")
R> library("simmr")
```

`simmr` requires the user to provide mixture data (y), source means (μ_{sk}), and source standard deviations (σ_{sk}). Other variables can be included as described in Section 3.4. The data is read into R using the `simmr_load` function. We will use an artificially generated dataset for illustration:

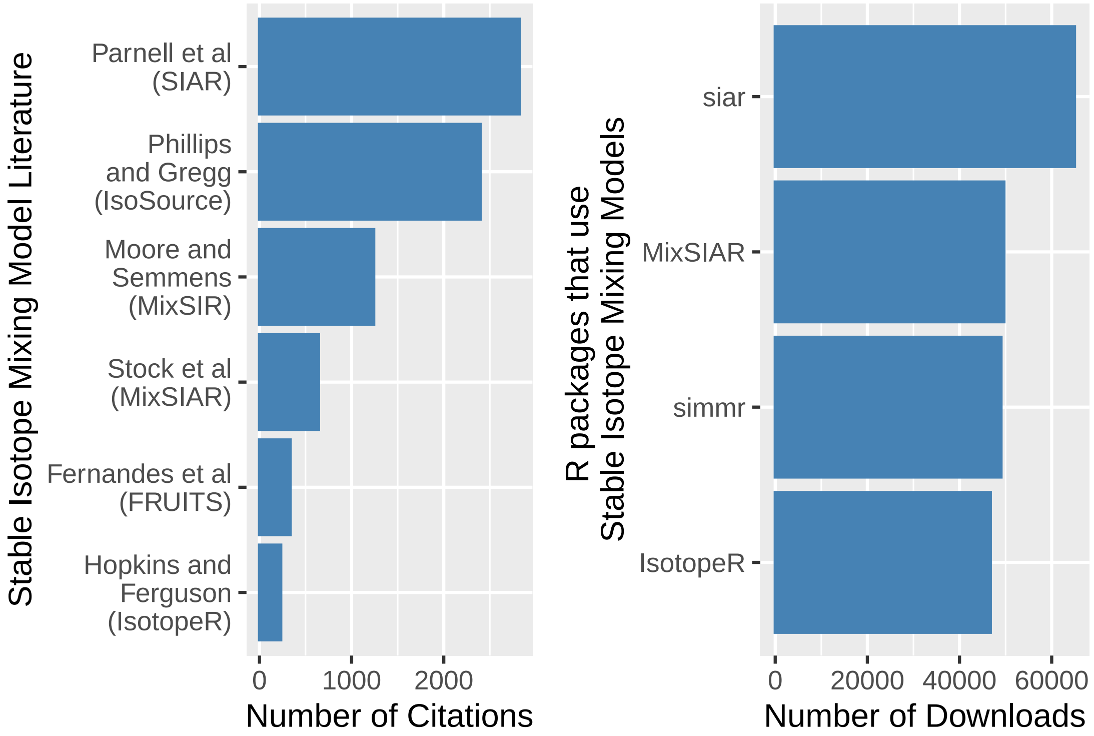


Figure 3.1: Barplot on the left showing the number of citations papers describing SIMMs software have received on Google Scholar and barplot on the right showing the number of downloads that packages using stable isotope mixing models have obtained (download numbers obtained via the cranlogs package (Csárdi, 2019)) (figures correct as of 9th June 2023).

```
R> y = data.frame(iso1 = c(4, 4.5, 5, 7, 6, 2, 3, 3.5, 5.5, 6.5))
R> mu_s = matrix(c(-10, 0, 10), ncol = 1, nrow = 3)
R> sigma_s = matrix(c(1, 1, 1), ncol = 1, nrow = 3)
R> s_names = c("A", "B", "C")
```

These artificially simple data have measurements of one isotope ratio, named ‘iso1‘ and three food sources, labelled A, B, and C. Loading the data into simmr creates an object of class simmr_in:

```
R> simmr_in_1 = simmr_load(
+       mixtures = y,
+       source_names = s_names,
+       source_means = mu_s,
+       source_sds = sigma_s)
```

We recommend that the user plot these data on an 'iso-space' plot before running any model. The iso-space plot shows the isotope(s) ratio on the x (and potentially y) axes. In this case, with one isotope, we need to check that the mixture values lie between the two most extreme values of the food sources on the iso-space plot for the mathematical model to give a reasonable fit to the data. With two isotopes its important to check that the mixture lies within the polygon that can be drawn by joining the food sources with straight lines. The shape created by joining the food sources is referred to as the mixing polygon. The iso-space plot can be generated by running the code:

```
R> plot(simmr_in_1)
```

Figure 3.2 shows the iso-space plot for this simple example, which shows the mixtures lie within the values of the most extreme food sources.The SIMM can then be fitted via MCMC using the simmr_mcmc function, which produces an object of class simmr_output and mcmc:

```
R> simmr_out_1 = simmr_mcmc(simmr_in_1)
```

When running simmr_mcmc the first step after running the model is to check convergence. This can be performed by running the following code:

```
R> summary(simmr_out_1, type = "diagnostics")

Summary for 1
Gelman diagnostics - these values should all be close to 1.
If not, try a longer run of simmr_mcmc.
deviance        A        B        C sd[iso1]
       1        1        1        1        1
```

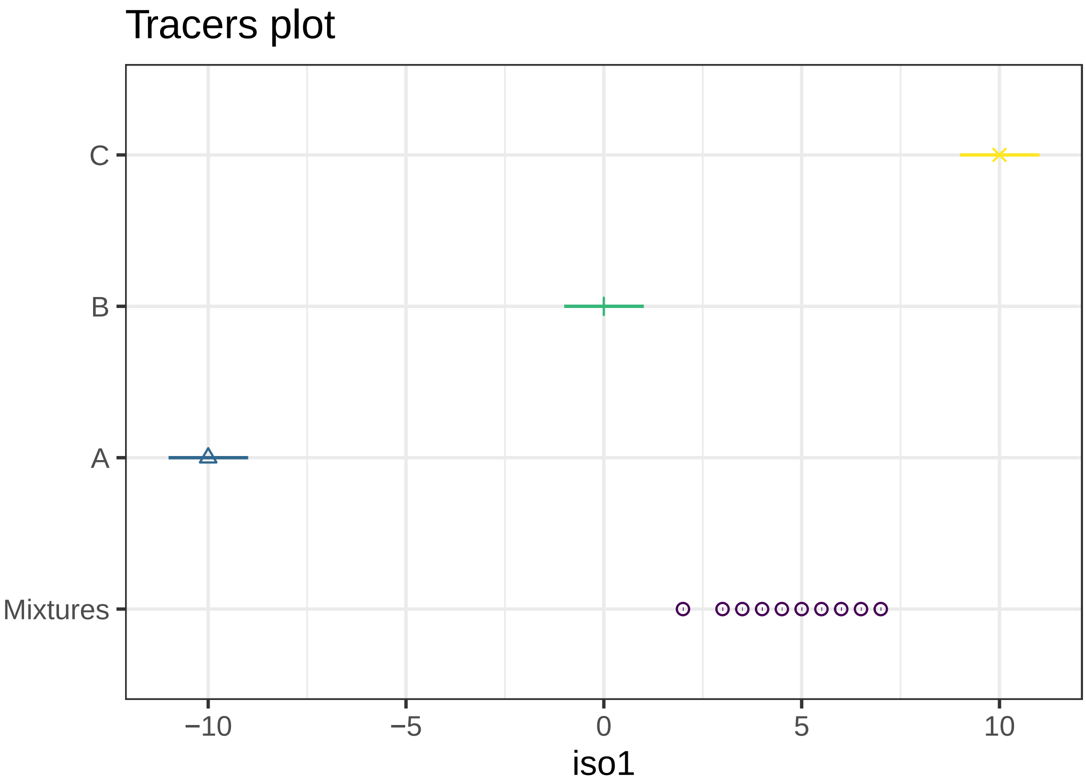


Figure 3.2: Simple iso-space plot produced by `simmr`. The isotope ratios are presented on the x-axis. A, B, and C represent different food sources (with error bars included) and the purple dots represent the mixture values.

The values in the diagnostics should all be close to 1.

`simmr` produces both textual and graphical summaries of the model run. Starting with the textual summaries, we can get tables of the means, standard deviations and credible intervals (the Bayesian equivalent of a confidence interval) with:

```
R> summary(simmr_out_1, type = "statistics")
```

```
Summary for 1
           mean    sd
deviance 40.466 3.481
A         0.149 0.068
B         0.245 0.135
```

```
C         0.606 0.085
sd[iso1]  2.103 0.788
```

This summary provides the mean and standard deviation estimates for the proportion of each food (A, B, and C) that these individuals are eating. It also provides an estimate of the marginal residual error of the isotope ratio (iso1 in this case). Here we can see food C is estimated to make up approximately 61% of these animals' diet. This finding matches the iso-space plot where the consumer isotope ratios are closest to the values for source C.

`simmr` has built-in functions to allow for visualisation of the results of these models once they have been run. There are multiple options for plotting the output, but perhaps the most useful is the matrix plot:

```
R> plot(simmr_out_1, type = "matrix")
```

Figure 3.3 shows histograms of the posterior distribution for the dietary proportions of each source on the diagonal, contour plots of the posterior relationship between the dietary proportions of each food source on the upper-right portion of the plot, and the posterior correlation between the sources on the lower-left portion of the plot. Large negative correlations indicate that the model cannot discern between the two sources; for example they may lie close together in iso-space. Large positive correlations are also possible when there are multiple competing sources. In general, high correlations (negative or positive) are indicative of the model being unable to determine which food sources are being consumed, though the marginal standard deviations can still be narrow. In this case the large negative correlations exist because there is only a single isotope and the model cannot discern, for example, which of sources A and B are pulling the mixture values to the left of source C.

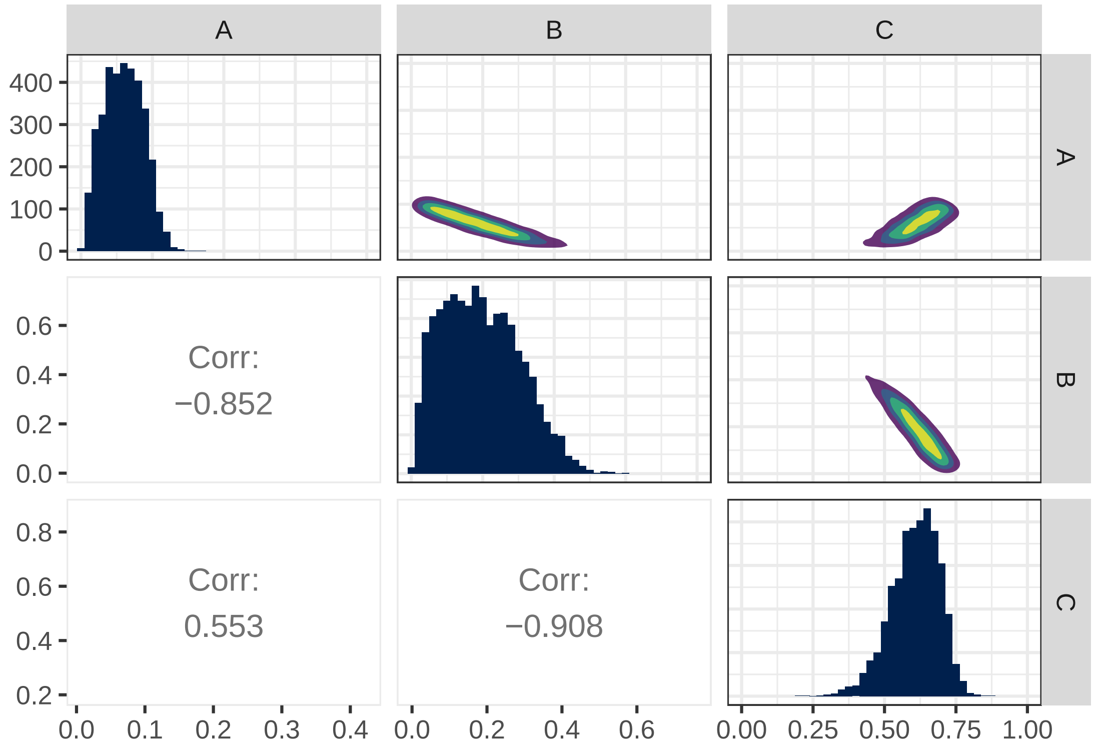


Figure 3.3: Matrix plot generated from Markov chain Monte Carlo run on Stable Isotope Mixing Model with 1 isotope. Histograms are presented along the diagonal showing the estimated proportion of each food source (A, B, and C) consumed by the individual. The upper-right section shows contour plots. The lower-left section gives correlation values.

3.3 Other software for fitting stable isotope mixing models

There have been many different software tools developed for the study of Stable Isotope Mixing Models (SIMMs). Table 3.1 gives a summary of these tools with a short description. The methods behind these tools includes both Frequentist and Bayesian approaches, and a variety of different fitting techniques. For reference, we provide code to fit our case study data (Section 3.6) using some of these packages at `https://github.com/emmagovan/simmr_paper_SIMM_package_scripts`.

The first widely available software for fitting SIMMs was `IsoSource` (Phillips and Gregg, 2003), which worked by generating all possible combinations of dietary

Package	Language	Reference	Description
Isosource	Visual Basic	Phillips and Gregg (2003)	Calculates all possible source combinations and returns feasible solutions
MixSIR	MATLAB	Moore and Semmens (2008)	Bayesian - Uses a sampling-importance-resampling algorithm
siar	R	Parnell et al. (2010)	Bayesian - uses Markov chain Monte Carlo (MCMC) for its fitting algorithm
FRUITS	Visual Basic	Fernandes et al. (2014)	Bayesian - allows for consideration of dietary routing. Operates based on Markov chain Monte Carlo simulations.
MixSIAR	R	Stock et al. (2018)	Bayesian - allows for consideration of fixed and random effects
simmr	R	This paper	Bayesian - can use either MCMC or fixed form variational Bayes (FFVB)

Table 3.1
Overview of different software for running Stable Isotope Mixing Models

proportions that would add to give the isotopic value of the individuals being studied, and presenting all possible combinations to the user. The recommendation was to report the distribution of solutions to avoid any misinterpretation of results. IsoSource was implemented in Visual Basic (Balena and Fawcette, 1999) via the IsoSource computer programme. Whilst IsoSource did not account for many of the intricacies of SIMMs, it was based on several previously developed ideas including concentration dependence, which provides the proportion of each element directly in the food source, (IsoConc; Phillips and Koch, 2002) and residual error (IsoError; Phillips and Gregg, 2001).

MixSIR (Moore and Semmens, 2008) was later developed and was the first to use a Bayesian framework, based on importance resampling, to estimate dietary proportions. MixSIR works by generating many vectors of possible proportional source contribution and calculating importance weights to determine the posterior distribution. MixSIR is implemented in MATLAB (Inc., 2022). It allows for several extensions over IsoSource including the ability to account for uncertainty, and incorporation of prior information. See Section 3.4 for a full description of these terms.

`SIAR` (Parnell et al., 2010) used a Bayesian framework but with Markov chain Monte Carlo (MCMC) as the fitting algorithm. It includes a residual error term and many of the extensions included in `MixSIR`. The R package is now defunct but maintained on GitHub for backwards compatibility. We have designed `simmr` to be a replacement for the `SIAR` software.

The `FRUITS` model (Fernandes et al., 2014) further extended the work by allowing the user to account for the concentration of different food fractions within each food source. `FRUITS` can also account for a diet-to-tissue signal offset which accounts for different tissues containing different ratios of isotopes, which is the same as concentration dependence discussed in Section 3.4. It also simplifies the incorporation of prior information. `FRUITS` is implemented in proglangVisual Basic via the FRUITS computer programme.

The most recent and perhaps most powerful R package, `MixSIAR` (Stock et al., 2018), is Bayesian, and allows for consideration of both fixed and random effect covariates on the dietary proportions amongst other extensions. `MixSIAR` works by creating a custom JAGS file for each model run. The package has a number of example data sets included and produces a wide array of output plots and summary statistics. However the model may not be appropriate for novice users and is very slow for complex data sets; which provides the motivation for development of `simmr` and the incorporation of FFVB.

3.4 Mathematical background of mixing models

The full model implemented in `simmr` for fitting a SIMM is:

$$y_{ij} \sim N\left(\frac{\sum_{k=1}^{K} p_k q_{jk}(\mu_{s,jk} + \mu_{c,jk})}{\sum_{k=1}^{K} p_k q_{jk}}, \frac{\sum_{k=1}^{K} p_k^2 q_{jk}^2 (\sigma_{s,jk}^2 + \sigma_{c,jk}^2)}{(\sum_{k=1}^{K} p_k q_{jk})^2} + \sigma_j^2\right)$$

where:

- y_{ij} are the mixture values for individual i on tracer j,
- $\mu_{s,jk}$ and $\sigma_{s,jk}$ are the mean and standard deviation of the source values for source k on tracer j,

- $\mu_{c,jk}$ and $\sigma_{c,jk}$ are the mean and standard deviation of the trophic discrimination factors(TFFs or "corrections") for source k on tracer j,
- q_{jk} represents concentration dependence for tracer j on source k,
- p_k are the proportions of each source k contributing to the mixture value
- σ_j is the residual standard deviation on tracer j.

The values y, $\mu_{s,jk}$, $\sigma_{s,jk}$, q_{jk}, $\mu_{c,jk}$, and $\sigma_{c,jk}$ are all given to the model as data.

The key extension of this model compared to the simple example given in Section 3.2 are that the model now includes multiple tracers, and corrects the proportions for Trophic Discrimination Factors (TDFs) and concentration dependence. As outlined above, TDFs account for the differential loss of one isotope over the other during the assimilation of diet into consumer proteins (Inger and Bearhop, 2008). Different tissues have different macronutrient compositions so TDFs can vary by tissue within a single consumer. Likewise different dietary items contain different elemental proportions (e.g., fats and carbohydrates contain little or no nitrogen when compared to proteins) and a concentration dependence correction can account for this (Phillips and Koch, 2002). The standard model assumes that a source contributes both elements (in the case of 2 isotopes) equally. Thus a concentration dependence value provides the proportion of that element directly in the food source (Phillips and Koch, 2002).

As before, the goal of the model fit is to estimate the posterior distribution of p given the data. As we fit the model using the Bayesian paradigm, prior distributions are required for the parameters. The prior for p follows a Centralised Log-Ratio (CLR) distribution (Aitchison, 1986):

$$[p_1, ...p_k] = \left[\frac{\exp(f_1)}{\sum_j \exp(f_j)}, \dots, \frac{\exp(f_K)}{\sum_j \exp(f_j)}\right]$$

f is then given a multivariate normal (MVN) distribution:

$$f \sim MVN(\mu_0, \Sigma_0)$$

The values of μ_0 and Σ_0 can then be set as vague (the defaults are $\mu_0 = 0$ and $\Sigma_0 = \mathbf{I}$) or tuned for informative prior situations (see later functions `prior_viz` and `simmr_elicit`). The prior distribution on σ is set as vague and gamma:

$$\sigma \sim Ga(a, b)$$

where a and b are small values. If correction values and concentration are to be used, they must also be provided by the user though they are not necessary to run the model (note: these should be applied in study of animal diets and migration, where TDFs in particular are needed in order to make appropriate inferences). Once the model is run it will then provide posterior samples for p, the proportion of each source in the mixture (for example the proportion of each food in the animals diet). Posterior distributions are also available for the parameters σ_j which, although are not of primary interest, can also provide some guidance as to the quality of the model fit since they quantify the size of the residual error.

3.5 Fitting SIMMs

3.5.1 Fitting using MCMC

The `simmr_mcmc` function allows the user to run their data through a mixing model coded using JAGS. The function has preset general priors for p, which can be altered by the user if they wish. The number of chains, iterations, burn-in period, and thinning can also be edited by the user, and are set to sensible default values otherwise.

The JAGS code for this model is provided as a model string inside the R function. The parameters saved when this model is run are p and σ. The output is assigned the class `simmr_output`. This allows for the package to use one plot function to plot inputs and outputs from both MCMC and FFVB. The function will pick out groups and run a separate MCMC algorithm for each one if needed. These groups can be represented by any categorical variable provided as part of the data. Grouping structures might include: demographic divisions such as age or sex, the same animals measured at different times of year; different packs within the same species; or populations of the same species living in different habitat types.

Often SIMMs need to be run on just a single consumer isotope observation, in which case the residual term becomes unnecessary. In cases where only a single observation is provided to simmr_load the model uses a prior for σ_j with high prior mass on zero. This is termed a ‘simmr solo‘ run. All the output plots and summaries work on this structure exactly as they do on a standard simmr run.

3.5.2 Fitting using VB

The simmr_ffvb function can be used if the user wishes to fit a SIMM using Fixed Form Variational Bayes (FFVB). FFVB works by approximating the full posterior using a simpler distribution (Pati et al., 2018). As it is an optimisation routine, it has the potential to run much faster than MCMC which relies on random sampling. FFVB aims to minimise Kullback-Leibler (KL; Kullback and Leibler, 1951) divergence between the posterior and the VB approximation. We provide a more detailed description of the FFVB fitting approach we use in Appendix 3.A.

Whilst the fitting method for FFVB is fundamentally different to the MCMC approach, the code still produces posterior samples of p and σ, and the output is assigned the class simmr_output as above. The user should not notice any difference in fitting using the two approaches, though fitting complicated models using FFVB should be faster than MCMC.

3.6 Case study: Brent Geese

This section provides code and explanation for running a two-isotope model in simmr with data in groups, in this case data on geese gathered at different times of year. The dataset is from Inger et al. (2006) and is provided as a sample data set within simmr. To begin we load in the package:

```
R> library("simmr")
```

In this example, our mixture is the geese, the sources are the food the geese eat and the tracers are $\delta^{13}C$ and $\delta^{15}N$. simmr requires the user to supply consumer data, the source means, and the source standard deviations. Trophic Discrimination Factors (TDFs) and concentration dependence are included in this example. The data is read into R using the simmr_load function. simmr has the ability to perform repeated runs on data sets if the data is separated into different groups. In simmr a separate model run will be performed for each group provided a grouping variable is given to simmr_load.

This data set can be called from the simmr package and an overview of the data can be seen via str:

```
R> str(geese_data)
```

```
List of 9
 $ mixtures           : num [1:251, 1:2] -11.4 -11.9 -10.6 -11.2 -11.7 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "d13C_Pl" "d15N_Pl"
 $ tracer_names       : chr [1:2] "d13C" "d15N"
 $ source_names       : chr [1:4] "Zostera" "Grass" "U.lactuca"
 "Enteromorpha"
 $ source_means       : num [1:4, 1:2] -11.17 -30.88 -11.17 -14.06 6.49
 ...
 $ source_sds         : num [1:4, 1:2] 1.21 0.64 1.96 1.17 1.46 2.27 1.11
 0.83
 $ correction_sds     : num [1:4, 1:2] 0.63 0.63 0.63 0.63 0.74 0.74 0.74
 0.74
 $ concentration_means: num [1:4, 1:2] 0.36 0.4 0.21 0.18 0.03 0.04 0.02
 0.01
 $ correction_means   : num [1:4, 1:2] 1.63 1.63 1.63 1.63 3.54 3.54 3.54
 3.54
 $ groups             : chr [1:251] "Period 1" "Period 1" "Period 1"
 "Period 1" ...
```

The data can then be used to create an object of class simmr_in:

```
R> simmr_groups = with(geese_data,
+       simmr_load(mixtures = mixtures,
+       source_names = source_names,
+       source_means = source_means,
+       source_sds = source_sds,
+       correction_means = correction_means,
+       correction_sds = correction_sds,
+       concentration_means = concentration_means,
+       group = groups))
```

We create the recommended iso-space plot to ensure all the mixtures lie inside the mixing polygon defined by the sources. group specifies which groups we want to plot. xlab and ylab allow for editing of the x and y axes labels. The following code creates an iso-space plot displaying groups 1 to 8. The axes labels are edited here to include the parts-per-mille sign.

```
R> plot(simmr_groups,
+       group = 1:8,
+       xlab = expression(paste(delta^13, "C (per mille)", sep = "")),
+       ylab = expression(paste(delta^15, "N (per mille)", sep = "")),
+       title = "Iso-space plot of Inger et al Geese data",
+       mix_name = "Geese")
```

The iso-space plot can be viewed in Figure 3.4 and it can be seen that all the data lies within the mixing polygon. The SIMM can be run either through JAGS or FFVB. The code is as follows:

```
R> simmr_groups_out = simmr_mcmc(simmr_groups)
R> simmr_groups_out_ffvb = simmr_ffvb(simmr_groups)
```

This runs each group independently. Future work could include the use of covariates, to allow for all 8 groups to be run together.

The first step after running simmr_mcmc is to check convergence, which can be performed by running the code:

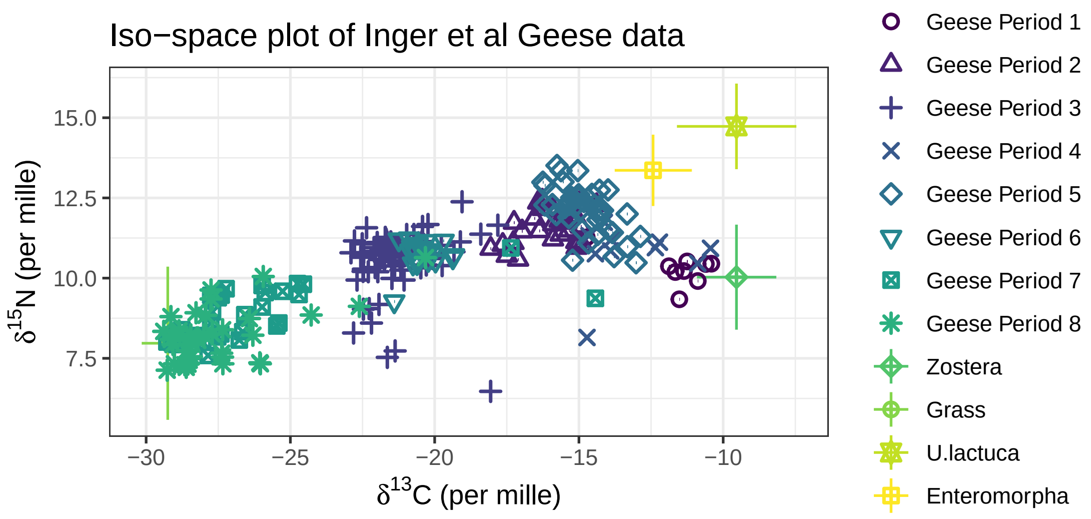


Figure 3.4: Iso-space plot of the eight groups of geese as well as their food sources.

```
R> summary(simmr_groups_out, type = "diagnostics")
```

It is important that all the diagnostic values are close to 1 - if not, a longer simmr_mcmc run is recommended. The diagnostic values are Gelman-Rubin statistics (Gelman and Rubin, 1992) and should tend towards 1 if convergence has occurred. simmr can produce textual summaries of the model run, an example of which can be seen below. Options within this function include quantiles, statistics, and correlations.

```
R> summary(simmr_groups_out,
+        type = "quantiles",
+        group = 1)

Summary for Period 1
               2.5%    25%    50%    75%  97.5%
deviance     52.775 56.292 59.347 63.174 72.727
Zostera       0.308  0.474  0.562  0.651  0.808
Grass         0.027  0.056  0.073  0.091  0.137
U.lactuca     0.022  0.076  0.134  0.208  0.376
Enteromorpha  0.024  0.099  0.183  0.298  0.534
```

```
sd[d13C_Pl]   0.551  1.135  1.543  2.056  3.849
sd[d15N_Pl]   0.272  0.646  0.935  1.347  2.563
```

```
R> summary(simmr_groups_out_ffvb,
+       type = "quantiles",
+       group = 1)

Summary for Period 1

              2.5%   25%   50%   75% 97.5%
Zostera      0.316 0.528 0.636 0.724 0.850
Grass        0.022 0.042 0.057 0.076 0.130
U.lactuca    0.023 0.069 0.118 0.188 0.402
Enteromorpha 0.022 0.075 0.138 0.238 0.540
d13C_Pl      0.463 0.682 0.887 1.193 2.510
d15N_Pl      0.447 0.628 0.777 1.020 1.896
```

The outputs are slightly different depending on whether the model has been run via MCMC or FFVB. For example, diagnostic values are only applicable for checking convergence of the MCMC run. For FFVB, we do not specify the number of iterations like we do with MCMC; instead we have implemented a stopping criterion in the underlying FFVB algorithm, which stops the run when the change in parameters between iterations falls below a specified threshold.

simmr has built-in functions to allow for visualisation of the results of these models once they have been run. Options for plots include matrix plots, boxplots, histograms, and density plots. The code for running a boxplot is as follows:

```
R> plot(simmr_groups_out,
+       type = "boxplot",
+       group = 2,
+       title = "simmr output group 2")
```

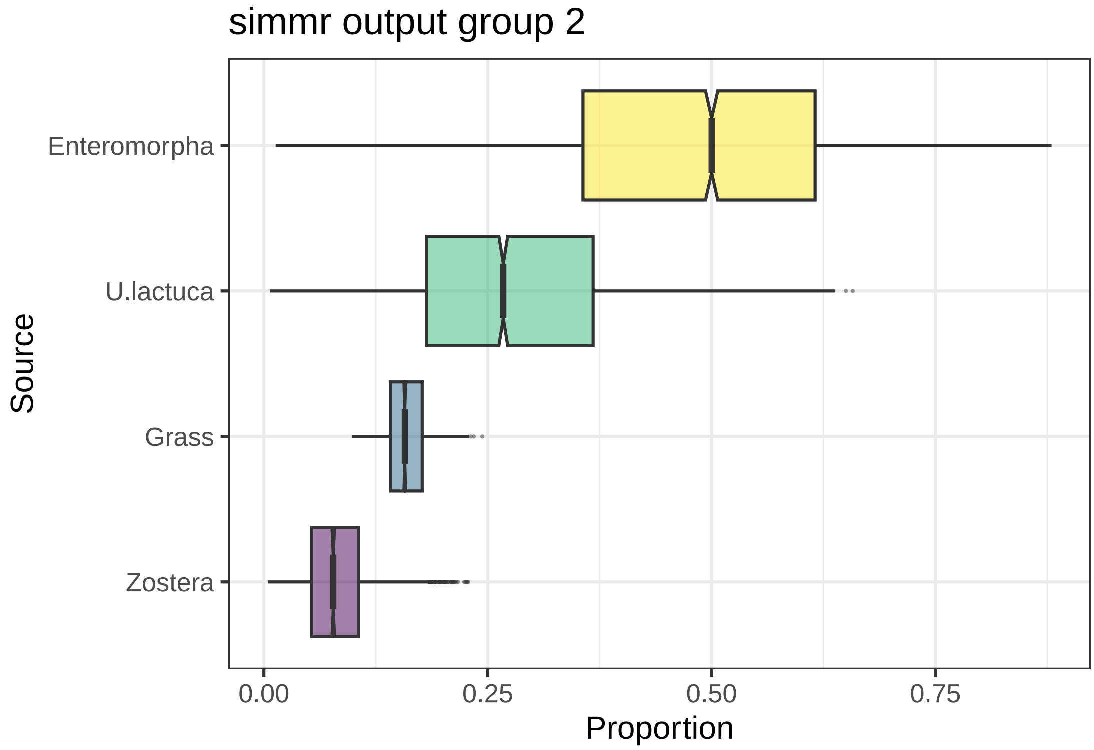


Figure 3.5: Boxplot of geese group 2 food proportion estimates generated from MCMC run.

```
R> plot(simmr_groups_out,
+       type = "matrix",
+       group = 6,
+       title = "simmr output group 6")
```

Figure 3.5 shows boxplots which show the proportion each food source is estimated to make up of the animals diet. The boxplot allows for easy visualisation of the proportions. Figure 3.6 shows the histograms of source proportions on the diagonal, contour plots of the relationship between the sources on the upper diagonal, and the correlation between the sources on the lower diagonal. In this case, we can see that the geese are consuming mostly *Enteromorpha spp*, some *Ulva lactuca* and Grass, and hardly any *Zostera* spp. The `compare_groups` and `compare_sources` functions in `simmr` allow for comparison of source consumption across different groups or sources. Below and in Figure 3.7 we show the output of comparing the

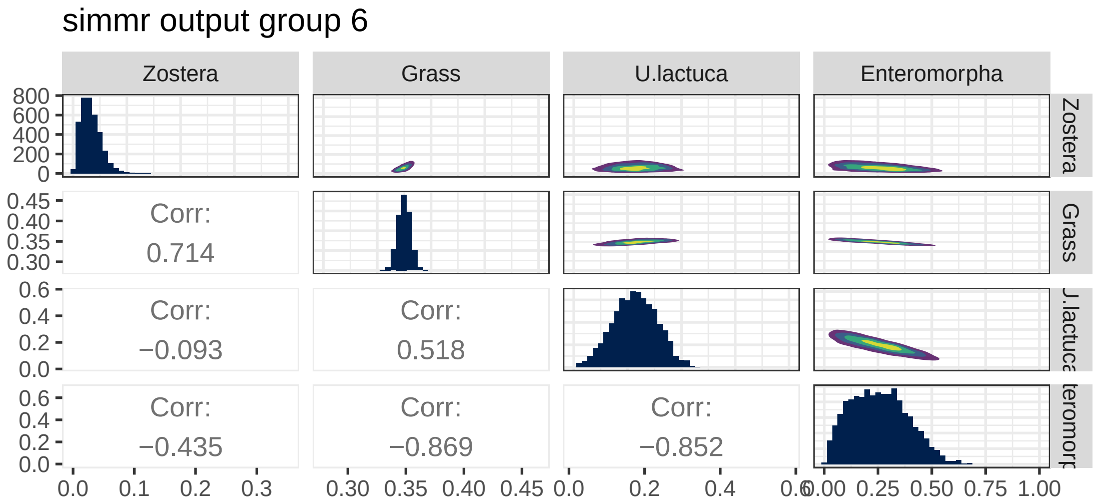


Figure 3.6: Matrix plot of geese group 6 generated from the MCMC run. The diagonal contains histograms showing the estimated proportion of each food. The upper section shows contour plots. The lower section gives correlation values.

proportions of *Zostera* spp between groups 1 and 2:

```
R> compare_groups(simmr_groups_out, source = "Zostera", groups = 1:2)
```

```
Prob (proportion of Zostera in group Period 1 > proportion
of Zostera in group Period 2) = 0.999
```

Beyond the main functions for plotting and summarising SIMMs, `simmr` contains other functions that may be useful for the user in interpreting output or guiding model development. `prior_viz` allows for visualisation of the priors set for the data versus the eventual posterior, and saves the data in a data frame if the user wishes to create their own plots. The function can be especially helpful when some food sources do not provide knowledge about diet and so the posterior dietary proportion reverts to the prior. Figure 3.8 shows the results of using the function on the data above with the following code:

```
R> prior <- prior_viz(simmr_groups_out, group = 1)
```

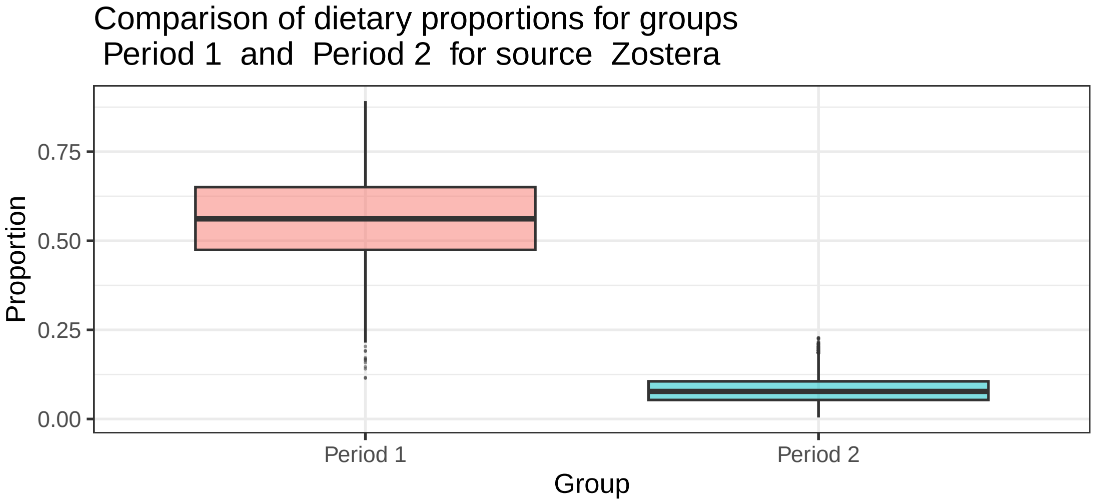


Figure 3.7: Boxplot comparing proportion of Zostera in diet in Period 1 versus Period 2.

In Figure 3.8 we can see that the algorithm is able to learn more about *Zostera* spp. and Grass as they are the most distinct sources. *Enteromorpha* and *U. lactuca* are closer in iso-space and less distinct and so they revert to the prior.

The `simmr_elicit` function allows the user to input informative prior information to the model and can be used with both the MCMC and FFVB functions. This function requires users to input a vector of proportion means and a vector of proportion standard deviations. Prior information about the dietary proportions might come from sources such as direct observation, faeces, stomach contents, or prey remains (Moore and Semmens (2008), Franco-Trecu et al. (2013), Hertz et al. (2017)). Finding appropriate prior values for the latent multivariate normal distribution parameters (μ_0 and Σ_0) can be hard since the prior information is usually available in the dietary proportion space. The function thus runs an optimisation routine to match provided dietary proportions to optimal values of μ_0 and Σ_0 and provides these to the user so that they can be added as arguments to e.g., `simmr_mcmc`. The `prior_viz` function can then be used to see the effect of these prior assumptions on the posterior.

When sources lie in similar locations on the iso-space plots it is sometimes desirable to combine sources together. The `simmr` package allows the user to choose

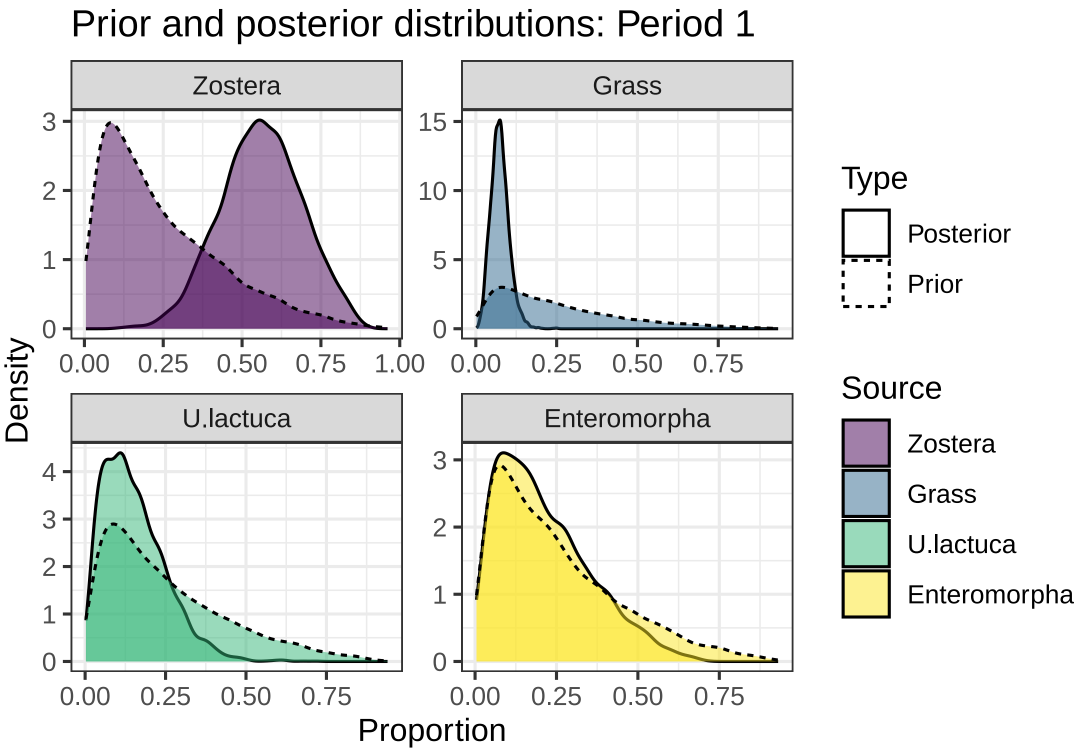


Figure 3.8: Four density plots showing the posterior and prior of each of the four food sources (Zostera, Grass, Ulva lactuca, and Enteromorpha).

which sources to combine *a posteriori* using the `combine_sources` function. The advantage of combining such sources is that the negative covariance between their estimated proportions will reduce the variance of the resulting summed source contribution. The output of `combine_sources` is also of class `simmr_output` and so can be used with all other plotting and summary functions.

For a final check of the model fit, the function `posterior_predictive` creates the posterior predictive distribution of the observations and plots this for each observation. For models that fit well we would expect, for example, 50% of observations to be within the 50% posterior predictive distribution. The function re-runs the JAGS code for the model but with an extra likelihood term inserted to extract the posterior predictive distribution. These values are returned from the function to enable more advanced use of the posterior predictive. The output is seen in Figure 3.9.

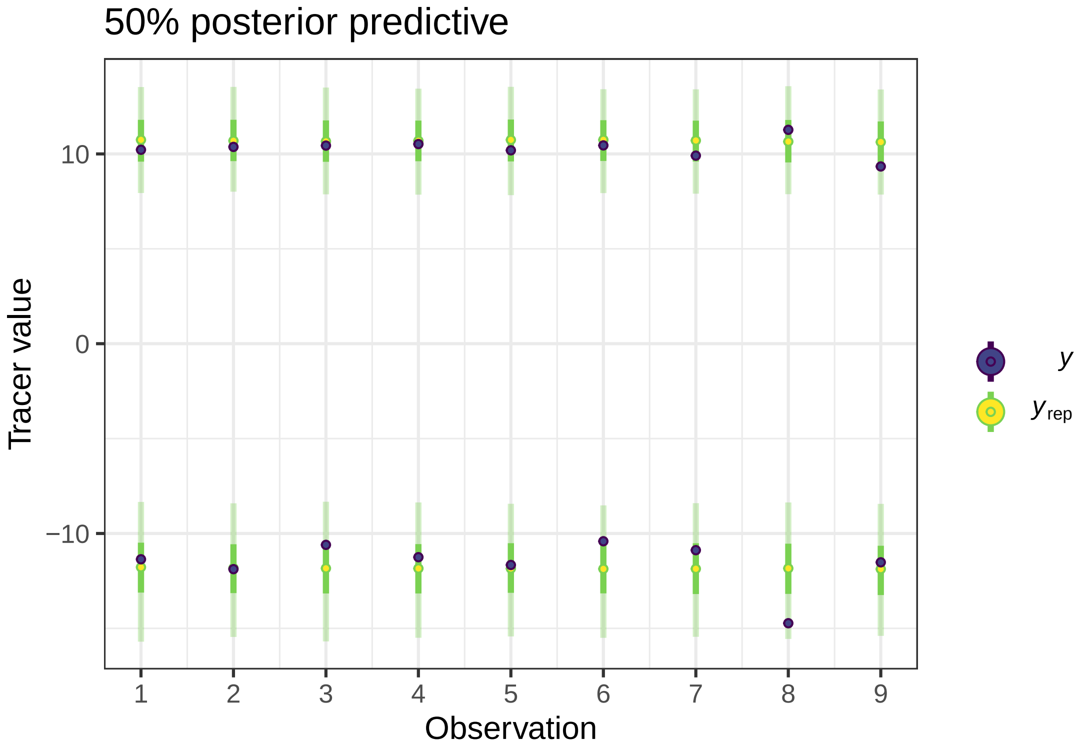


Figure 3.9: Posterior predictive distribution of the observations for geese data group one, probability interval = 0.5

3.7 Discussion

Our package `simmr` allows analysis of basic SIMMs using either MCMC or FFVB algorithms. It is a convenient tool with built-in class objects that allows analysis to be performed easily. Plots and summaries are simple to produce and the class system makes it easy to process data, even though the underlying data structure can be relatively complex for non-R users.

There are a number of assumptions made which are common to many of these models. Probably the most fundamental is that the consumer is at equilibrium with their food and that their diet is static. In a dynamic system, clearly this assumption is always violated, but careful interpretation of the results can still yield valid insights by shifting the considered time window over which the diet is quantified using SIMMs (Arnoldi et al., 2023). We assume that we know all the food sources that the animal is eating. If a food source is missing it can affect the shape of the mixing polygon and the resulting diet composition obtained by the model (Phillips et al., 2014). It is highly recommended that the user views the iso-space plot before running the model to ensure that the data points lie within the mixing polygon created by the food sources. If the food sources do not lie within the mixing polygon, it can indicate that a food source has been missed. However, all data lying within the mixing polygon does not guarantee every food source has been found. It is important to consider the biology of the organism, how robust sampling was, and to ensure samples have been obtained of every food the organism eats. Similar assumptions would apply if studying other systems.

There are a number of areas where the user will need to make decisions, for example, whether or not to exclude outliers, and whether food sources should be combined. Combining sources can be done through the `combine_sources` function provided by `simmr` but it is recommended that this is performed *a posteriori*, if it is to be performed, and that there is a sound biological basis for doing so.

Our new package has the advantage that the models are quick to run, easy to use, and have several built in checks. We have built-in functions which allow customisable high quality plots to be produced that allow for sources to be combined *a posteriori* and for sources and groups to be compared. Future plans include expansion of this package to allow it to run with more complex models, and include random and fixed effects.

Appendix

3.A Fixed Form Variational Bayes Algorithm

We use the FFVB algorithm of (Tran et al., 2021). If we define the joint set of parameters as $\theta = (f, \tau)$ where $\tau = \sigma^{-2}$ then we write our factorised variational posterior as:

$$q_\lambda(\theta) = q(f)q(\tau)$$

where $\lambda = (\mu_f, \Sigma_f, c, d)^T$ is the set of hyper-parameters associated with the variational posteriors:

$$\begin{aligned} q(f) &\equiv MVN(\mu_f, \Sigma_f) \\ q(\tau) &\equiv Ga(c, d) \end{aligned}$$

To start the algorithm, initial values are required for $\lambda^{(0)}$ (we use parenthetical super-scripts to denote iterations.), the sample size S, the adaptive learning weights (β_1, β_2), the fixed learning rate ϵ_0, the threshold α, the rolling window size t_W and the maximum patience P.

Define h to be the log of the joint distribution up to the constant of proportionality:

$$h(\theta) = \log\left(p(y|\theta)p(\theta)\right)$$

and h_λ to be the log of the ratio between the joint and the VB posterior:

$$h_\lambda(\theta) = \log\left(\frac{p(y|\theta)p(\theta)}{q_\lambda(\theta)}\right) = h(\theta) - \log q_\lambda(\theta)$$

The initialisation stage proceeds with:

1. Generate samples from $\theta_s \sim q_{\lambda^{(0)}(\theta)}$ for $s = 1, ...S$

2. Compute the unbiased estimate of the lower bound gradient:
$$\nabla_\lambda \widehat{LB(\lambda^{(0)})} = \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda [\log(q_\lambda(\theta_s))] \circ h_\lambda(\theta_s)\Big|_{\lambda=\lambda^{(0)}}$$
where $\circ$ indicates element-wise multiplication

3. Set
$$\begin{aligned}\bar{g}_0 &:= \nabla_\lambda LB(\lambda^{(0)}) \\ \bar{\nu}_0 &:= \bar{g}_0^2 \\ \bar{g} &= g_0 \\ \bar{\nu} &= \nu_0\end{aligned}$$

4. Estimate the control variate c_i for the ith element of λ as:
$$c_i = \frac{Cov\left(\nabla_{\lambda_i}[\log(q_\lambda(\theta))]h_\lambda(\theta), \nabla_{\lambda_i}[\log(q_\lambda(\theta))]\right)}{Var(\nabla_{\lambda_i}[\log(q_\lambda(\theta))])}$$
across the samples generated in step 1

5. Set $t = 1$, patience $= 0$, and 'stop $=$ FALSE'.

Now the algorithm runs with:

1. Generate samples from $\theta_s \sim q_{\lambda^{(t)}(\theta)}$ for $s = 1, ...S$

2. Compute the unbiased estimate of the lower bound gradient:
$$g_t := \nabla_\lambda \widehat{LB(\lambda^{(t)})} = \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda [\log(q_\lambda(\theta_s))] \circ (h_\lambda(\theta_s) - c)\Big|_{\lambda=\lambda^{(t)}}$$
where $\circ$ indicates element-wise multiplication.

3. Estimate the new control variate c_i for the ith element of λ as:

$$c_i = \frac{Cov\left(\nabla_{\lambda_i}[\log(q_\lambda(\theta))]h_\lambda(\theta), \nabla_{\lambda_i}[\log(q_\lambda(\theta))]\right)}{Var(\nabla_{\lambda_i}[\log(q_\lambda(\theta))])}$$

   across the samples generated in step 1

4. Compute:

$$\begin{aligned} v_t &= g_t^2 \\ \bar{g} &= \beta_1 \bar{g} + (1-\beta_1) g_t \\ \bar{v} &= \beta_2 \bar{v} + (1-\beta_2) v_t \end{aligned}$$

5. Update the learning rate:

$$l_t = min(\epsilon_0, \epsilon_0 \frac{\alpha}{t})$$

   and the variational hyper-parameters:

$$\lambda^{(t+1)} = \lambda^{(t)} + l_t \frac{\bar{g}}{\sqrt{\bar{v}}}$$

6. Compute the lower bound estimate:

$$\widehat{LB}(\lambda^{(t)}) := \frac{1}{S} \sum_{s=1}^{S} h_{\lambda^{(t)}}(\theta_s)$$

7. If $t \geq t_W$ compute the moving average LB

$$\overline{LB}_{t-t_W+1} := \frac{1}{t_W} \sum_{k=1}^{t_W} \widehat{LB}(\lambda^{(t-k+1)})$$

   If $\overline{LB}_{t-t_W+1} \geq \max(\bar{L}B)$ patience = 0, else patience = patience +1

8. If patience $\geq$ P, ‘stop = TRUE‘

9. Set $t := t + 1$

CHAPTER 4

cosimmr: an R package for fast fitting of Stable Isotope Mixing Models with covariates

The study of animal diets and the proportional contribution that different foods make to their diets is an important task in ecology. Stable Isotope Mixing Models (SIMMs) are an important tool for studying an animal's diet and understanding how the animal interacts with its environment. We present `cosimmr`, a new R package designed to include covariates when estimating diet proportions in SIMMs, with simple functions to produce plots and summary statistics. The inclusion of covariates allows for users to perform a more in-depth analysis of their system and to gain new insights into the diets of the organisms being studied. A common problem with the previous generation of SIMMs is that they are very slow to produce a posterior distribution of dietary estimates, especially for more complex model structures, such as when covariates are included. The widely-used Markov chain Monte Carlo (MCMC) algorithm used by many traditional SIMMs often requires a very large number of iterations to reach convergence. In contrast, `cosimmr` uses Fixed Form Variational Bayes (FFVB), which we demonstrate gives up to an order of magnitude speed improvement with no discernible loss of accuracy. We provide a full mathematical description of the model, which includes corrections for trophic discrimination and concentration dependence, and evaluate its performance *against the state of the art MixSIAR model. Whilst MCMC is guaranteed to converge to the posterior distribution in the long term, FFVB converges to an approximation of the posterior distribution, which may lead to sub-optimal performance. However we show that the package produces equivalent results in a fraction of the time for all the examples on which we test. The package is designed to be user-friendly and is based on the existing* `simmr` *framework.*

4.1 Introduction

Stable Isotope Mixing Models (SIMMs) are commonly used in ecology to study the proportional contribution that different foods make to an animal's diet (Phillips, 2012). This information can be important as it allows scientists to look at diet, which resources are important for different species (McDonald et al., 2020), and consequently niche overlap and competition (Teixeira et al., 2021; Aksu et al., 2023), as well as being useful in looking at trophic position and energy flow in an ecosystem (Manlick and Newsome, 2022). These models have been extensively used by ecologists over the past 20 years with recent papers revealing the foraging behaviour in Dugongs (Thibault et al., 2024), overlap in trophic niche between native and non-native species of carp (Aksu et al., 2023), and assessment of nursery areas used by the scalloped hammerhead shark (Paez-Rosas et al., 2024). SIMMs have been shown to produce results comparable to direct observation (Swan et al., 2020). The approach relies on the fact that the stable isotopes of several elements, but most usefully those of hydrogen($\delta^2 H$), carbon ($\delta^{13}C$), nitrogen ($\delta^{15}N$) and sulphur ($\delta^{34}S$), are incorporated into animal tissues from the diet in a predicable manner (Inger and Bearhop, 2008). Thus, if the isotope ratios of potential dietary items are known then animal diets can be reconstructed from the stable isotope ratios from proteinaceous tissues using SIMMs. `cosimmr` is a new R package (R Core Team, 2021) developed to allow for the fast running of SIMMs, especially but not limited to those that include covariates. It has been designed to be easy to use for non-expert R users, with S3 classes used throughout. SIMMs are widely used and cited in other fields, such as geology (Munoz et al., 2019) and pollution studies (Zaryab et al., 2022), amongst many others. In other scientific areas, SIMMs and similar models can be referred to in the literature as ‘source apportionment models’ (Hopke, 1991), ‘end member analysis’ (Hooper et al., 1990), or ‘mass balance analysis’ (Miller et al., 1972). Further discussion of these models can be found in Govan et al. (2023).

The basic mathematical equation for a statistical SIMM is:

$$y = \sum_{k=1}^{K} p_k s_k + \epsilon.$$

Here y is the mixture (consumer tissues) value (for example, the $\delta^{13}C$ or $\delta^{15}N$ values for the species we wish to study), p_k are the proportions contributed by each source (dietary item) k (of K total sources), s_k is the source tracer value for source k, and ϵ is a residual term. The parameters p_k are usually the main focus of scientific interest. These models are commonly expanded in diet analyses to allow for processes that cause the mixture and source tracer values to differ besides the source proportions, such as Trophic Discrimination Factors (TDFs; Inger and Bearhop, 2008) and concentration dependence (Phillips and Koch, 2002). Other expansions include process error on the dietary proportions (Moore and Semmens, 2008) as well as hierarchical source fitting (Ward et al., 2010). The models are further made richer by incorporating random effects on the source values (Semmens et al., 2009). Here, whilst we include many of these extensions, our focus is on the inclusion of covariate dependence on the proportions p_k. The restriction that these must sum to unity (i.e. a simplex) makes their estimation more complex, and specialist link functions are required to map their values on to covariates. We provide a more detailed explanation of the mathematical model behind `cosimmr` in Section 4.2.

Modern SIMMs are fitted using the standard tools of Bayesian inference. Most commonly this involves using Markov chain Monte Carlo (MCMC) to obtain samples from the posterior. For complex models with covariates this can be extremely slow, with models requiring millions of posterior samples and taking several days to converge, if they converge at all. By contrast in `cosimmr` we use Fixed Form Variational Bayes (FFVB), specifically Gaussian Variational Bayes with Cholesky Decomposed Variance (Titsias and Lázaro-Gredilla, 2014; Tan and Nott, 2018).

FFVB is an optimisation-based algorithm which works by first defining the form of the posterior distribution (here a multivariate normal distribution), and then minimising the Kullback-Leibler divergence between this VB approximation and the true posterior distribution. More details on the assumed distributions used in `cosimmr` can be found in Section 4.3. The main advantage of using FFVB, and thus of `cosimmr`, is that it works by optimisation rather than sampling, and therefore can be much faster to produce a posterior distribution. However, convergence issues can still occur (Yao et al., 2018). Our approach gives `cosimmr` a key advantage over other packages, which tend to use JAGS (Plummer, 2003) to implement the MCMC algorithm. Additionally, `cosimmr` runs using C++ code via Rcpp (Eddelbuettel and François, 2011) which allows for an additional speed boost.

SIMMs have been a popular method for studying animal diets, with thousands of citations across different papers, and many packages have been developed in order to make this easier. Some of the most popular packages are listed below. A summary is also provided in Table 4.1.1.

- Isosource (Phillips and Gregg, 2003) was one of the earliest developed packages for running SIMMs. It worked by simulating possible values of each proportion to produce many potential combinations of proportions. Importantly, it lacked an explicit statistical basis.

- MixSIR (Moore and Semmens, 2008) adopted a Bayesian framework and allowed for the inclusion of Trophic Discrimination Factors (TDFs). MixSIR utilised Importance Sampling, generating many samples of possible proportion combinations and calculating importance weights to produce the final posterior sample. MixSIR also allowed for the inclusion of prior information and allowed uncertainty to be incorporated into SIMMs.

- SIAR (Parnell et al., 2010) was developed as an R package. It utilised Markov chain Monte Carlo (MCMC) sampling. SIAR also included a residual component ϵ in the model. SIAR is no longer updated and simmr was developed to replace it.

- simmr (Govan et al., 2023) is an R package that follows a Bayesian framework, and provides the option of using either JAGS (Just Another Gibbs Sampler; Plummer, 2003) or Fixed Form Variational Bayes (FFVB; Tran et al., 2021) for running the models. simmr allows for the inclusion of concentration dependence and Trophic Discrimination Factors but does not allow for covariates on the dietary proportions.

- FRUITS (Fernandes et al., 2014) allowed for the inclusion of concentration dependence and prior information in a Bayesian framework. FRUITS is encoded in Visual Basic and runs via the FRUITS computer programme. This package simplifies the incorporation of prior information.

- IsotopeR (Hopkins III and Ferguson, 2012) adopted a Bayesian framework and allowed for inclusion of concentration dependence and TDFs, and also includes covariance. It uses MCMC for running the models.

- MixSIAR (Stock et al., 2018) is an R package that fits models in JAGS. MixSIAR allows for the inclusion of covariates as fixed, random, or continuous effects. It fits the source means hierarchically, either using raw data or sample statistics (means, variances, and sample sizes).

Software	Language	Algorithm	TDFs	Concentration Dependence	Covariates	Prior Info	Comments	Hierarchical/ Source Fitting	Reference
Isosource	Visual Basic	Trial and Error	N	N	N	N	Frequentist	N	Phillips and Gregg (2003)
simmr	R	MCMC and FFVB	Y	Y	N	Y	Ease-Of-Use Design	N	Govan et al. (2023)
FRUITS	Visual Basic	MCMC	Y	Y	N	Y	-	N	Fernandes et al. (2014)
IsotopeR	R	MCMC	Y	Y	N	Y	Hierarchical Model	Y	Hopkins III and Ferguson (2012)
MixSIAR	R	MCMC	Y	Y	Y	Y	Allows for Raw Data	Y	Stock et al. (2018)
cosimmr	R	FFVB	Y	Y	Y	Y	Aims for Speed	N	This Paper

Table 4.1.1: Table showing summary of current SIMM software and the features they offer

TDFs (Trophic Discrimination Factors) account for the fact that consumers may differentially lose 'light' versions of isotopes with respect to 'heavy' versions during the process of assimilation (Inger and Bearhop, 2008). Thus, while TDFs are important in ecological applications, they have less relevance to geological or pollution-based applications as these same processes do not occur (although similar processes such as isotopic fractionation may need to be accounted for). TDFs can be calculated in the lab, or calculated mathematically, such as in Greer et al.

(2015). Alternatively SIDER (Healy et al., 2018) is an R package that allows for estimates of TDFs based on phylogenetic relatedness of species. Estimates can also be obtained from the scientific literature.

Concentration dependence accounts for the fact that different food sources can contribute proportionally different amounts of each isotope (Phillips and Koch, 2002). The standard two-isotope model assumes all food sources contribute both isotopes equally. However, there are often occasions where a food source can be rich in one isotope and poor in another, thus not contributing equally to both. Instead, concentration dependence assumes a source's contribution to each isotope is proportional to the mass of the food source times the elemental concentration of the isotope within the food source. Inclusion of concentration depedence facilitates conversion between consideration of either the total mass of food sources assimilated and the mass of specific elements within them.

The inclusion of covariates in the SIMM allows users to avoid pseudo-replication (Hurlbert, 1984), because if a covariate is important to the diet proportions, its exclusion violates the assumption that all mixtures are independently and identically distributed. Including covariates in SIMMs allows users to determine the potentially causal relationships between covariates and diet proportions. Although the model returns these coefficients, they are only available in a transformed space (via the link function) and not directly interpretable. We have designed cosimmr to produce interpretable output in 'coefficient space' where users can determine the direction of the relationship and evaluate uncertainty, and also in 'proportion space' (p-space) which allows users to see the effect of the covariate directly on the dietary proportions. These tools are defined via a predict function that to allow the user to predict dietary proportions based on combinations of covariates that may not necessarily be present in the original dataset. We follow the 'tidyverse' (Wickham et al., 2019) style guide, with the Snake case naming convention and S3 classes used throughout. Being able to evaluate the model at new values of the covariates allows for a more detailed picture to be seen. Uncertainty intervals, in the form of Bayesian credible intervals, are provided for all estimated quantities. Advanced users have access to the full posterior distributions as created by the FFVB algorithm.

The data required by `cosimmr` can be illustrated by Figure 4.1.1. This example is discussed more fully in Section 4.6.3. This figure shows an 'iso-space' plot generated by `cosimmr`. It shows each individual plotted in iso-space, with the axes representing $\delta^{13}C$ and $\delta^{15}N$. The diet sources (Marine and Freshwater in this case) are also plotted on the graph. It is important that all individuals lie within the mixing polygon created by the source means. If they do not, it indicates that the mixing system does not follow the model assumptions. Possible reasons for observations lying outside the mixing polygon include: issues with data collection; inaccurate TDFs; or missing food sources, amongst others. However, note that the mixing polygon vertices are sample means subject to sampling error, so there is some uncertainty in their exact position. Hierarchical Bayesian models allow for the source means to deviate from the source sample means by maximizing the likelihood of the source and mixture tracer data together (Ward et al., 2010; Hopkins III and Ferguson, 2012; Stock et al., 2018).

4.2 Statistical approaches to stable isotope mixing models

The fundamental SIMM we fit can be written as:

$$y_{ij} = \sum_{k=1}^{K} p_k(\mathbf{x}_i) q_{jk}(s_{ijk} + c_{ijk}) + \epsilon_{ij}$$

Where:

- y_{ij} are the mixture/consumer tracer values of individual i for tracer j,
- $p_k(\mathbf{x}_i)$ are the proportions of each source k contributing to the mixture value at each covariate value $\mathbf{x}_i$ where $\mathbf{x}_i$ is an L-vector of covariate values for individual i,
- q_{jk} represents the concentration dependence for tracer j on source k,
- s_{ijk} is the consumed source value by individual i of the food source k on tracer j,

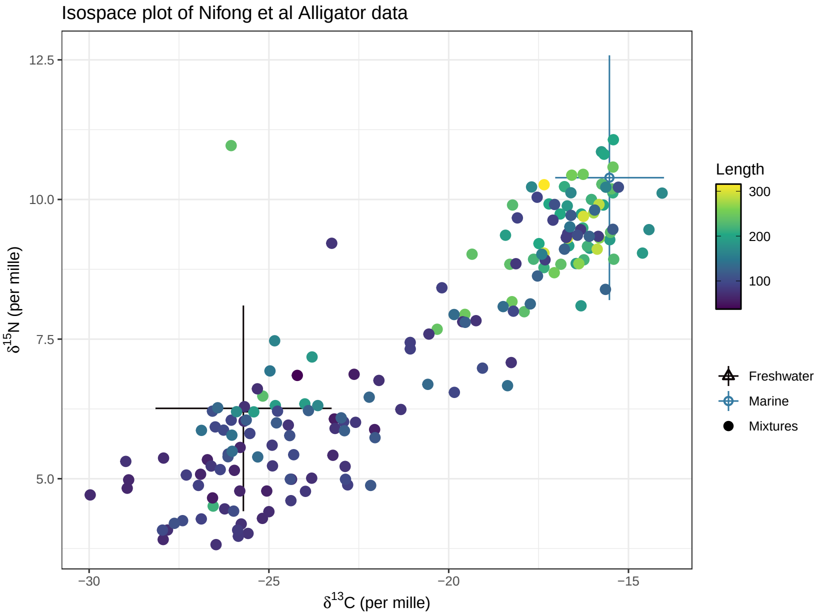


Figure 4.1.1: Iso-space plot for Alligator dataset (discussed in Section 4.6.3). Individuals are represented by circles coloured by covariate (length) and their isotope-ratio values are adjusted by TDFs. Two diet sources, Freshwater and Marine, are represented by a black triangle and blue circle, respectively, with 1 standard deviation also plotted. The axes on this plot are the Carbon-13 and Nitrogen-15 ratios of the isotope with respect to the 'standard' measurement.

- c_{ijk} is the trophic discrimination factor of individual i for source k on tracer j
- ϵ_{ij} is the residual error for individual i on tracer j

We index individuals as $i = 1, \dots, N$, tracers as $j = 1, \dots, J$, and sources as $k = 1, \dots, K$. We assume there are $l = 1, \dots, L$ covariates so that $\mathbf{x}_i = \{x_{i1}, \dots, x_{iL}\}$. For notational brevity we write $p_k(\mathbf{x}_i)$ as p_{ik}. It is common to make the prior assumptions that $\epsilon_{ij} \sim N(0, \sigma_j^2)$, $s_{ijk} \sim N(\mu_{s,jk}, \sigma_{s,jk}^2)$, and $c_{ijk} \sim N(\mu_{c,jk}, \sigma_{c,jk}^2)$. Here $\mu_{s,jk}, \mu_{c,jk}, \sigma_{s,jk}, \sigma_{c,jk}$ may be assumed fixed as they are commonly available from other data sources, or learnt as part of a Bayesian hierarchical model. The residual standard deviation σ_j is usually given a Uniform or (inverse) Gamma weakly informative prior. Other approaches have used multivariate normal distributions for these terms (e.g. Hopkins III and Ferguson, 2012; Parnell et al., 2013;

Stock et al., 2018) but these are not implemented in `cosimmr` as yet. The model of Stock et al. (2018) adds an additional multiplicative parameter to the first variance term to account for the assimilation of food items according to whether organisms are specialising in certain regions of the source probability distribution. Following their approach, we give the parameter ξ a $U(0, 20)$ prior when this additional process error is required. A table of prior values set can be seen at 4.2.1.

Term	Prior
ξ	U(0,20)
β_{kl}	N(0,1)
$\frac{1}{\sigma_j}$	Ga(1,1)

Table 4.2.1: Table showing default priors used in `cosimmr`

The source and TDF random effects add an additional burden of $2KJ$ parameters into the model which can cause a significant computational slowdown. Moore and Semmens (2008) proposed proposed marginalising across these parameters to produce a more complex, but more computationally tractable likelihood:

$$y_{ij} \sim N\left(\frac{\sum_{k=1}^{K} p_{ik}q_{kj}\mu_{sc,kj}}{\sum_{k=1}^{K} p_{ik}q_{kj}}, \frac{\sum_{k=1}^{K} p_{ik}^2 q_{kj}^2 \sigma_{sc,kj}^2}{\sum_{k=1}^{K} p_{ik}^2 q_{kj}^2} + \sigma_j^2\right)$$

where $\mu_{sc,kj} = \mu_{s,kj} + \mu_{c,kj}$ and $\sigma_{sc,kj}^2 = \sigma_{s,kj}^2 + \sigma_{c,kj}^2$.

The remaining prior distribution is that of the p_{ik} terms which must retain the constraint that $\sum_k p_{ik} = 1$, but also allow for the terms to be dependent on the covariates $\mathbf{x}_i$. We use a Centralised Log-Ratio (CLR; Aitchison (1986)) link function so that:

$$[p_{i1}, ...p_{iK}] = \left[\frac{\exp(f_{i1})}{\sum_j \exp(f_{ij})}, \ldots, \frac{\exp(f_{iK})}{\sum_j \exp(f_{ij})}\right]$$

This prior has the advantage that the resulting terms f_{ik} are unconstrained and made to depend directly on $\mathbf{x}_i$. We model this dependence linearly, but extensions that capture more nuanced dependence seem like a fruitful avenue for future research. We thus set: $f_{ik} = \mathbf{x}_i^T \beta_k$. In other words, we can write the proportion for individual i consuming food k as $p_{ik} = CLR(\mathbf{x}_i^T \beta_k)$. where the parameters β_k

model the dependence of the covariates across source k. We require a further prior distribution on β_k to ensure identifiability. By default we thus set $\beta_{kl} \sim N(0, 1)$. By default we also scale the covariates. Users can use un-scaled covariates if they wish but caution is advised. In certain circumstances where prior information is available on the β_{kl} values we may use an informative prior of the form $\beta_{kl} \sim N(\mu_{\beta,kl}, \sigma^2_{\beta,kl})$. The prior distribution for β can be changed by the user in `cosimmr` via the `cosimmr_ffvb` function.

4.3 The Fixed Form Variational Bayes Algorithm

Fixed Form Variational Bayes (FFVB) is an optimisation-based algorithm that aims to approximate the posterior distribution of a Bayesian model in a predefined form (Salimans and Knowles, 2013). It aims to finds the parameters of the 'closest' probability distribution to that of the true posterior. Unlike traditional MCMC sampling methods, the greedy nature of the optimisation can usually find this approximate posterior far faster. In `cosimmr` we use a sub-type of FFVB known as Gaussian Variational Bayes with Cholesky decomposed covariance (Titsias and Lázaro-Gredilla, 2014; Tan and Nott, 2018). To avoid becoming diverted in mathematical detail, we defer a full description of our approach to Appendix 4.A. However here we provide an intuitive guide to how the fitting process works.

Our model assumes that the joint posterior distribution of all the parameters is multivariate normal. The parameters for our model are β, and $\boldsymbol{\sigma^2}$, representing the regression parameters across sources and covariates, and the residual variances across tracers. Recall that p is a deterministic function of β so not included in the algorithm. Since the variances are all restricted to be positive, we model these on the log scale. We thus write $\theta = \{\beta, \log(\boldsymbol{\sigma^2})\} = \{\beta_{11}, \ldots, \beta_{KL}, \log(\sigma_1^2), \ldots, \log(\sigma_J^2)\}$ as the set of parameters for which we want a posterior.

For FFVB we need to define the form of the posterior distribution. We use:

$$\theta \sim MVN(\mu_\theta, \boldsymbol{\Sigma}_\theta)$$

where μ_θ and $\boldsymbol{\Sigma}_\theta$ are the mean and covariance matrix of the approximated posterior distribution. To avoid the positive semi-definite constraints on $\mathbf{\Sigma}_\theta$ we model the Cholesky decomposition of this matrix so that $\mathbf{\Sigma}_\theta = \mathbf{L}\mathbf{L}^T$. Together these terms are vectorised and written as: $\lambda = \{\text{vec}(\mu_\theta), \text{vec}(\mathbf{L})\}$. The goal of the algorithm is to provide a set of optimal values for λ which captures this posterior distribution. The MVN distributions allows us to capture correlations between parameters. The main steps of the algorithm are as follows:

- Get starting values and use these to get an estimate of the difference between the posterior and variational posterior.
- Calculate the gradient of this difference and use this to update λ. The gradient of the posterior is calculated using automatic differentation and the gradient of the variational posterior is calculated manually.
- Use new values of λ in place of starting values and repeat until stopping condition is met.

The algorithm requires several user-set hyper-parameters for fitting the model. The main hyper-parameters are: the patience P, which determines when the algorithm stops; and S which provides the number of parameter samples used at each stage of the algorithm. For other algorithm parameters, we have used the default values from Tran et al. (2021). These parameters include: the fixed (`beta_1` and `beta_2`) and adaptive (`eps_0`) learning rates; the size of the window to use when calculating stopping conditions (`t_W`); and the threshold for exploring the learning space before a the learning rate is decreased (`tau`). The stopping conditions work by calculating a moving average of the lower bound (the difference between the log of the posterior and the log of the variational posterior). When the moving average does not improve after P iterations then the algorithm stops. All hyper-parameters can be changed by the user if they wish when running the FFVB algorithm through `cosimmr`, though we have provided reasonable defaults which should work in most circumstances.

The use of FFVB in covariate-dependent SIMMs is novel, and confers an advantage over MCMC due to the increase in speed. This method is flexible and can be extended in the future, to include hierarchical source fitting, raw data, as well as random effects, features which are currently available in other SIMM software. We see up to an order of magnitude speed increase when comparing FFVB to MCMC, with comparable results produced.

4.4 Running the cosimmr package

The cosimmr package is available via CRAN. The first step to use it is to install and load the package:

```
R> install.packages("cosimmr")
R> library(cosimmr)
```

The user must provide mixture data (y), source means ($\mu_{s,kj}$), and source standard deviations ($\sigma_{s,kj}$) to run cosimmr. TDFs, Concentration Dependence, and any covariates are not necessary to run the model but they may need to be included in order for the model to be ecologically valid. The cosimmr_load function can be used to read the data into R. This ensures the data is loaded in the correct format. For illustration purposes we will use an artificially generated dataset, though see later sections for real case studies:

```
R> y = matrix(c(5, 5.1, 4.7, 3.6, 3.2, 0, -1, -2, -3, -7,
+               3.1, 5.6, 3.6, 4.7, 1.3, 1, -4, -3, -7, -9),
+               ncol = 2)
R> colnames(y) = c("iso1", "iso2")
R> mu_s = matrix(c(-10, 0, 10, -10, 10, 0), ncol = 2, nrow = 3)
R> sigma_s = matrix(c(1, 1, 1, 1, 1, 1), ncol = 2, nrow = 3)
R> s_names = c("A", "B", "C")
R> x = c(1.6, 1.7, 2.1, 2.5, 1.1, 3.7, 4.5, 6.8, 7.1, 7.7)
```

This dataset contains measurements of two isotopic ratios, ‘iso1’ and ‘iso2’, as well as three food sources named ‘A’, ‘B’, and ‘C’. There is one continuous covariate, named ‘x’. These data can then be loaded into cosimmr using cosimmr_load to create an object of class cosimmr_in:

```
R> cosimmr_in_1 = cosimmr_load(
+                               formula = y ~ x,
+                               source_names = s_names,
+                               source_means = mu_s,
+                               source_sds = sigma_s
)
```

As discussed in the introduction, it is recommended that these data are plotted on an 'iso-space' plot before modelling. The iso-space plot shows tracer values of the mixtures as well as sources, with each axis representing a tracer. These tracers are often isotope ratios. It is important that the mixture data lies within the polygon formed when the sources are joined with straight lines (this polygon is referred to as the mixing polygon). If the mixtures do not lie within this polygon it indicates that there is an issue - potential reasons are that TDFs are inappropriate or a food source has been omitted. The polygon vertices are subject to sampling error and this could be another potential source of error. The iso-space plot can be generated by `cosimmr` by running the following code:

```
R> plot(cosimmr_in_1, col_by_cov = TRUE, cov_name = "x")
```

The resulting plot can be seen in Figure 4.4.1. It shows the mixtures lie within the mixing polygon. These data can then be run through the `cosimmr_ffvb` function to produce an output of class `cosimmr_out`:

```
R> cosimmr_out_1 = cosimmr_ffvb(cosimmr_in_1)
```

`cosimmr` has built-in functions to produce summaries of the model run. Both graphical and textual summaries can be produced, as shown below.

```
R> summary(cosimmr_out_1, type = "statistics")

Summary for Observation 1
```

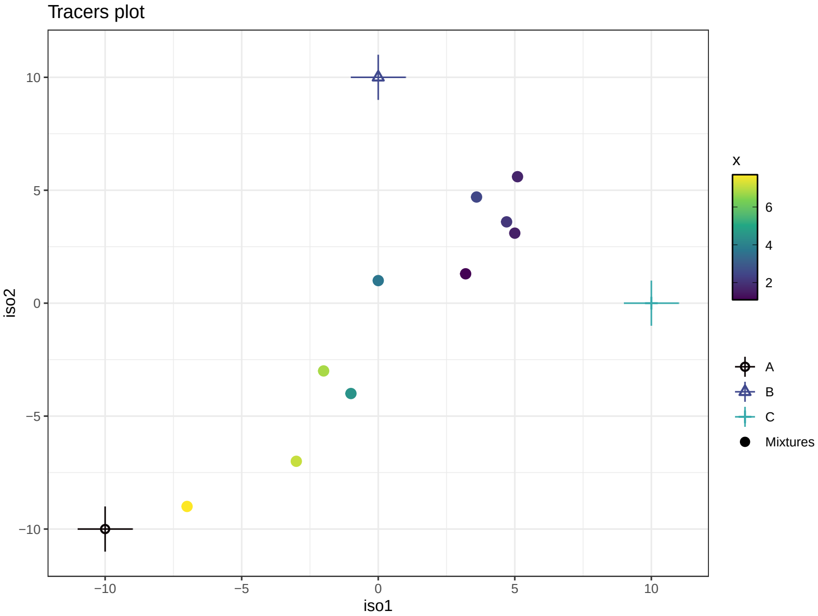


Figure 4.4.1: An iso-space plot generated for artificial dataset showing iso1 on the x-axis and iso2 on the y-axis. The food sources A, B, and C are shown. Individuals are shown by circles coloured by covariate x.

```
         mean    sd
P(A)    0.093 0.026
P(B)    0.424 0.052
P(C)    0.482 0.045
sd_iso1 1.449 1.133
sd_iso2 1.790 1.437
```

In this example, we have specified that we wish to produce ‘statistics’. As we have not specified an observation then the function defaults to returning statistics for observation 1. Any/multiple individuals can be selected and ‘statistics’, ‘quantiles’ or ‘correlations’ can be produced for each individual. The ‘statistics’ summary produces a table of the means and standard deviations for the estimates of the proportion of each food eaten by the individual. An estimate of the marginal residual error of the isotope ratios is produced. In this summary we can see that individual 1’s diet is mostly composed of foods B and C, with A contributing very little to their diet. This matches the observations we can make from the iso-space plot, where individual 1 lies at (5, 3.1), equidistant from food sources B and C and quite far away from food source A. The 'quantiles' summary produces the 2.5%, 25%, 50%, 75% and 97.5% quantiles for the same values as provided by the 'statistics' summary. The 'correlations' option produces the correlation values between each source and the residual error of the isotope ratios.

Graphical plots can be produced in `cosimmr`. For example, below we create a proportion plot for observation 1. We can see the plot in Figure 4.4.2. This shows the range of the proportion estimates for each food source for individual 1. We also create a 'covariates plot' which shows the change in the proportion of foods consumed as the covariate changes. This plot can be seen in Figure 4.4.3.

```
R> plot(cosimmr_out_1, type = c("prop_histogram", "covariates_plot"),
+                            obs = 1, cov_name = "x")
```

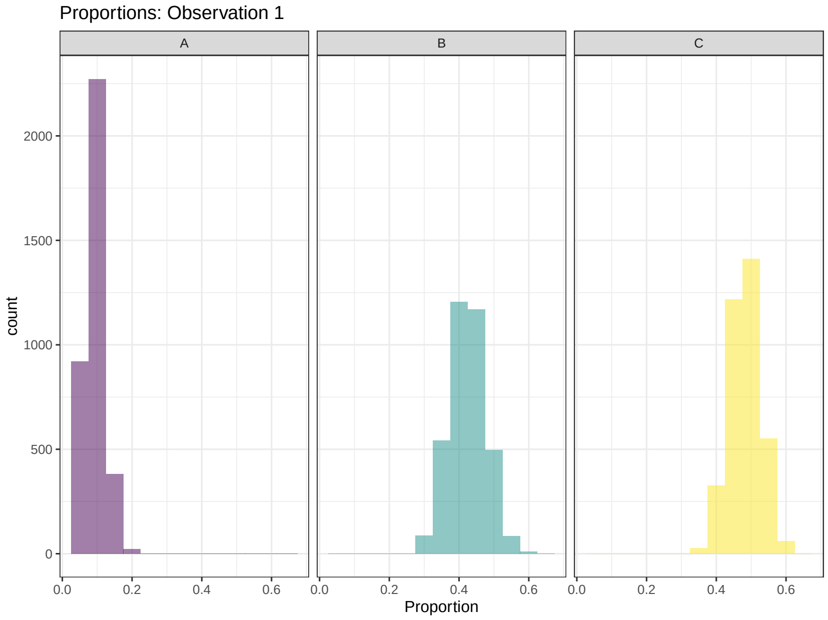


Figure 4.4.2: `cosimmr` proportion plot showing consumption of different food sources for observation 1 for the simple example

Another important function within `cosimmr` is the ability to predict values based on covariate values, using the `predict` function, as illustrated below:

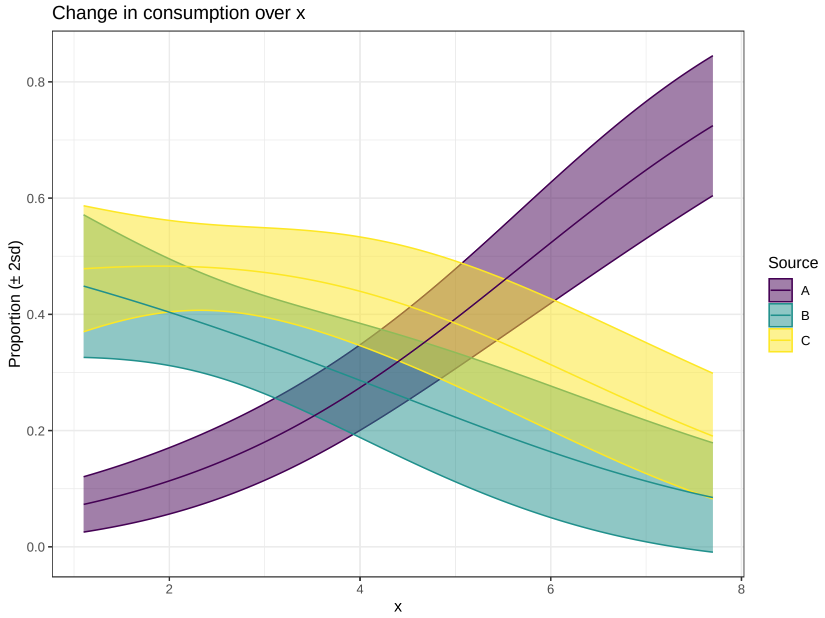


Figure 4.4.3: Covariates plot showing the change in the consumption of three food sources A, B, and C, as the covariate ‘x’ increases for the simple example. Shaded interval shows mean ± 2 standard deviations.

```
R> x_pred = data.frame(x = c(3, 5))
R> pred_out = predict(cosimmr_out_1, x_pred)
R> summary(pred_out, type = "statistics", obs = c(1,2))

Summary for Observation 1

      mean    sd
P(A) 0.181 0.033
P(B) 0.345 0.041
P(C) 0.474 0.039

Summary for Observation 2

      mean    sd
P(A) 0.393 0.043
P(B) 0.221 0.054
```

```
P(C) 0.386 0.054
```

The `predict` function allows us to provide a vector of covariate values, and returns a `cosimmr_out` object, which can be used as normal in the `summary` and `plot` functions to return graphs and textual summaries. This allows users to return predictions for covariate values not observed in the sample population, which can enrich understanding of the system. It is also important, however, for users to note that caution is advised for predictions of values which lie outside of the data.

We can use the `prior_viz` function to visualise how the posterior has changed from the prior. This plot overlays the posterior distribution and the prior distribution to see how it has changed or if the posterior has not changed much from the prior. This plot can be seen in Figure 4.4.4. We can see that our posterior estimates have changed from the prior estimate. One figure is shown here but it is recommended that users create plots for multiple individuals when running a model.

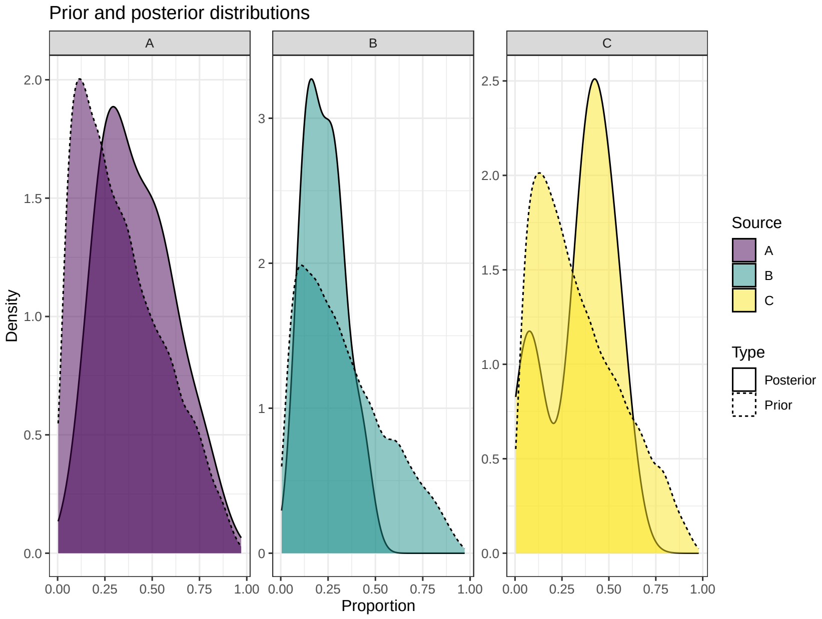


Figure 4.4.4: Three density plots showing the prior and posterior for each of the 3 food sources A, B, and C, in the simple example. Posterior estimates are shown for individual 1.

For convenience a summary of the main `cosimmr` functions is presented in Table 4.4.1.

`cosimmr_load`	Load in data in correct order/format
`cosimmr_ffvb`	Run SIMM using Fixed Form Variational Bayes algorithm
`summary`	Produce summary of proportion values. Options include statistics, quantiles, and correlations
`plot`	Create plots, options include histogram or boxplot of beta values, iso-space plot, histogram or density plot of estimated proportions, plot of covariates vs proportions
`predict`	Predict proportion values for covariates not present in original dataset
`posterior_predictive`	Create posterior predictive distribution of observations and plots for each observation
`prior_viz`	Create plots showing prior values set vs posterior obtained

Table 4.4.1: Main functions available in `cosimmr`

4.5 Simulation Checks

In this section we use simulated data to verify that `cosimmr` returns valid estimates when a run is performed. We simulate data from the model using a variety of different data set sizes, varying N, J, K and L, and changing the main parameters in the model. We evaluate the performance of the model by looking at how often the posterior distribution obtained by `cosimmr` matches the true values. The code for performing the runs in this section can be found at https://github.com/emmagovan/cosimmr_paper/.

The values selected for several different simulations are presented in Table 4.5.1. We run low ($N = 50$, $J = 2$, $K = 3$, $L = 2$), medium ($N = 200$, $J = 3$, $K = 4$, $L = 5$) and high ($N = 500$, $J = 4$, $K = 5$, $L = 10$) versions to capture a range of scenarios, where N = no. of individuals, J = no. of tracers, K = no. of food sources, and L = no. of covariates. For each of these we simulate data using the default prior distribution of $\beta_{kl} \sim N(0, 1)$ and with $\sigma \sim Ga(1, 1)$. We set $\mu_s \sim U(-10, 10)$, and $\sigma_s \sim U(0, 2)$. TDFs and Concentration dependence are ignored for this example.

	Low	Medium	High
N	50	200	500
J	2	3	4
K	3	4	5
L	2	5	10

Table 4.5.1: Values of parameters for different runs of our simulation checks

After running each model, we produce posterior uncertainty intervals at the 50% level and calculate the proportion of posterior samples inside these values. As an example, the posterior predictive plot for tracer 1 in the Medium run is shown in Figure 4.5.1. Posterior predictive plots for other simulations and tracers are found in the Appendix 4.B (Figures 4.B.1, 4.B.2 and 4.B.3).

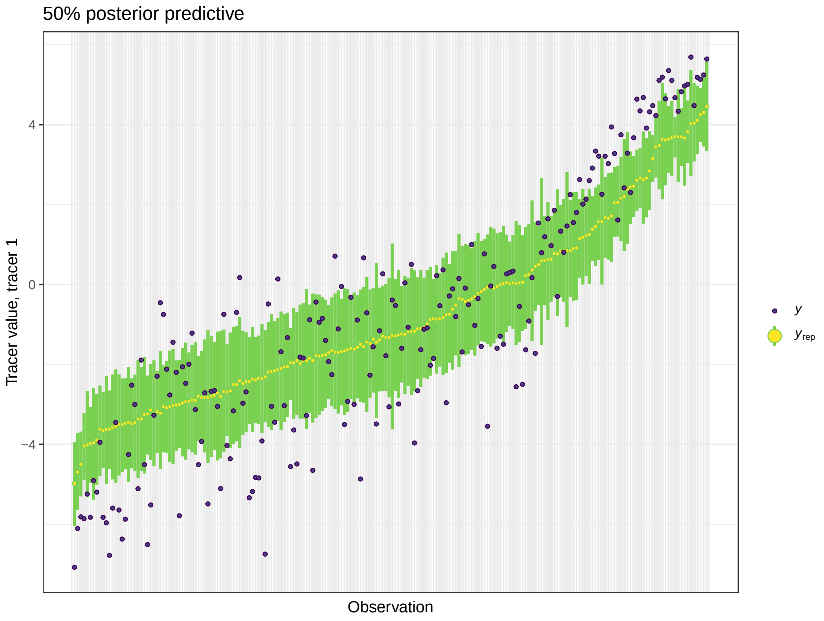


Figure 4.5.1: Plot showing posterior uncertainty intervals at the 50% level for the 'Medium' simulated model run for tracer 1. The proportion of posterior values inside these values was 51%.

The results show that `cosimmr` produces accurate estimates for the posterior values, even with more complex data and increased numbers of tracers and covariates.

We show an example posterior distribution of β in Figure 4.5.2 for the 'Low' run. From Figure 4.5.2 we can see that `cosimmr` is performing well and producing β

values similar to the original values used to generate data for this example, as illustrated by the red lines visible on the plots. This holds true for the Medium and High examples. Plots are omitted to avoid repetition.

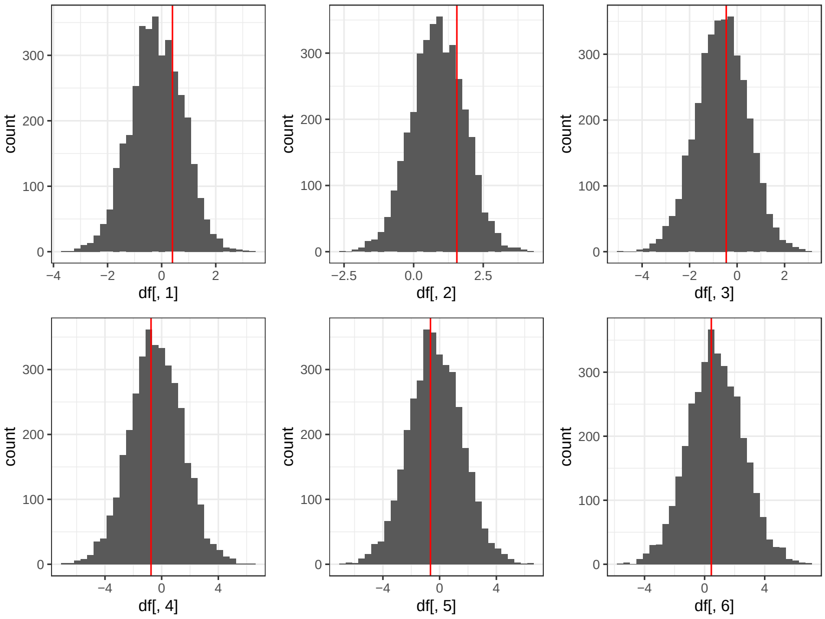


Figure 4.5.2: Histograms showing posterior samples for beta values generated via `cosimmr` for the 'Low' example, and red line showing 'true' value of β used to generate the mixture data.

4.6 Case Studies

We now perform a direct comparison of `cosimmr` and MixSIAR to evaluate both the accuracy of the FFVB posterior and check the computational gains. MixSIAR is very popular and widely cited, with nearly 70,000 downloads, as of August 11th 2025 (Csárdi, 2019). We provide three case studies: the first using the Geese data of Inger et al. (2006), for which we include a single categorical covariate (Group number). The second uses the Isopod data of Galloway et al. (2014) which contains 8 tracers and a single covariate (Site). The third uses the Alligator data of Nifong et al. (2015), for which we provide a detailed model comparison across 8 different potential covariate panels. In each case we compare the posterior distributions of the parameters, the posterior predictive performance, and the computational speed differences. All of the data for our model fits is available in the `cosimmr` package, available at https://github.com/emmagovan/cosimmr and on CRAN, and the code for running the models is available at https://github.com/emmagovan/cosimmr_paper.

4.6.1 Geese data (Inger et al., 2006)

Our first example looks at the Brent Geese (*Branta bernicla hrota*) dataset originally from Inger et al. (2006). The covariate in this example is 'Group' which is discrete. There are eight different groups which represent different time points at which individuals were sampled. $\delta^{13}C$ and $\delta^{15}N$ are the two isotopes used in this study. The iso-space plot for this data is seen in Figure 4.6.1. TDFs and concentration dependence are accounted for in this model.

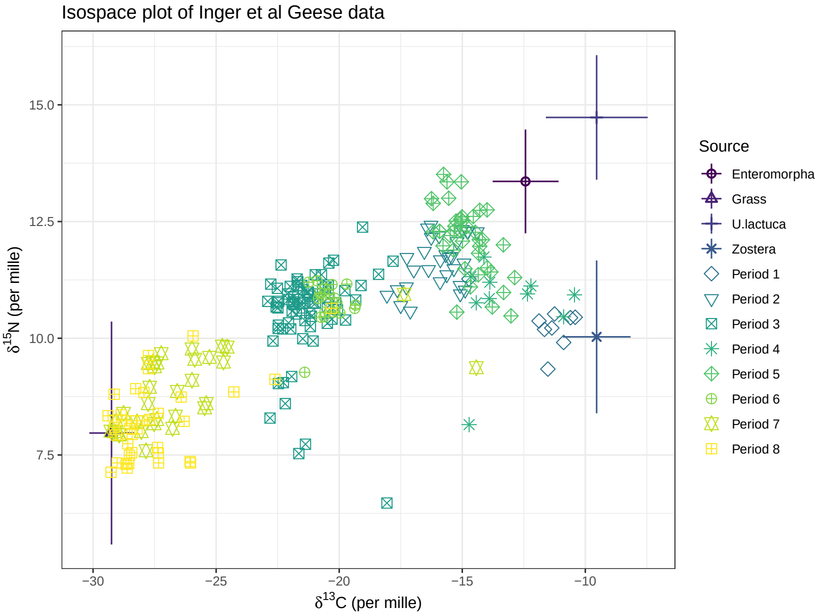


Figure 4.6.1: Iso-space plot for geese dataset, coloured by covariate to highlight the differences between groups.

By looking at the Group covariate and how the proportion of each food in the diet of the geese differs between groups, we can see how their diet changes over time. This example highlights the usefulness and importance of covariates in SIMMs. The time of year influences the diet of these geese. They are known to consume *Zostera* spp. between October and December at Strangford Lough (where these data were collected), and as time passes the geese remaining on the lough consume more *Enteromorpha spp.* and *Ulva lactuca* (Mathers and Montgomery, 1997). To exclude time as a covariate in this example would violate the assumption that the data are IID, as their diets are influenced by the time of year, and consequently, by what food is easily available to the geese.

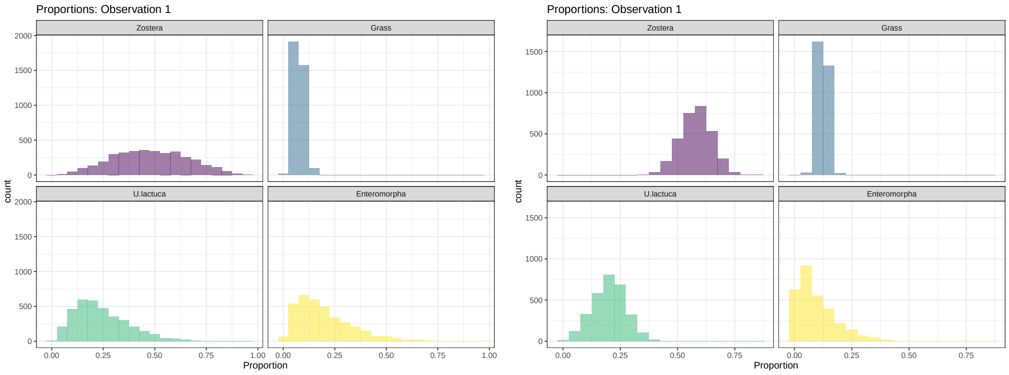
For this example, the Geese data was run through both `cosimmr` and MixSIAR. For MixSIAR a 'long' run was needed for convergence. The first thing to note from these model runs is that `cosimmr` produces these results in a much quicker timeframe than MixSIAR, as we can see in Table 4.6.1. `cosimmr` is over three times faster than MixSIAR for this example. The 'Group' covariate is discrete and therefore in `cosimmr` it is treated as eight covariates when transformed into numeric covariates. Therefore this example is slower in `cosimmr` than other examples with only one numeric covariate. The code for using a categorical covariate in `cosimmr` is demonstrated below:

```
R> data(geese_data)
R> cosimmr_geese_in = cosimmr_load(
    formula = geese_data$mixtures ~ as.factor(geese_data$groups),
    source_names = geese_data$source_names,
    source_means = geese_data$source_means,
    source_sds = geese_data$source_sds,
    correction_means = geese_data$correction_means,
    correction_sds = geese_data$correction_sds,
    concentration_means = geese_data$concentration_means)
```

	min	lower quartile	mean	median	upper quartile	max	neval
cosimmr	29.5	30.4	38.0	33.7	45.0	61.2	10
MixSIAR	113.6	121.5	130.1	124.5	139.3	155.3	10

Table 4.6.1: Table showing computation time (minutes) of `cosimmr` and MixSIAR ('long' run needed for convergence) for Geese example, showing the minimum (min), lower quartile, mean, median, upper quartile, maximum (max) time, and number of evaluations (neval).

As well as comparing computational time, it is important that `cosimmr` produces results that are comparable to other SIMM software. From looking at Figure 4.6.2 we can see that `cosimmr` and MixSIAR produce comparable results in terms of proportional estimates. These figures show the estimated dietary proportions for group 1. We can see that both `cosimmr` and MixSIAR produce similar estimates for the percentage that each food makes up in the diet of the first group. This result holds for the other seven groups in this dataset. Differences in results may be due to a slight difference in error structure between `cosimmr` and MixSIAR. For this example, MixSIAR recommends only accounting for residual error. In `cosimmr` we instead recommend process and residual error, to account for errors in sampling as well as specialisation by individuals. We chose to compare the recommended models from each package for this comparison as well as comparisons in the next two sections.



(a) `cosimmr`

(b) MixSIAR

Figure 4.6.2: Proportion plot showing consumption of different food sources for observation 1 for Geese example for `cosimmr` and MixSIAR

We can generate more complex plots using the `plot` function in `cosimmr`, to see how the consumption of a specific food changes between groups. In Figure 4.6.3 we see the difference in consumption of *Zostera* spp. across different groups. This highlights the usefulness of including covariates, as this detail would otherwise be lost. A convergence check can be performed using the function `convergence_check`. This function returns the mean lower bound values produced and shows that this value converges. The result of this can be seen in Figure 4.6.4.

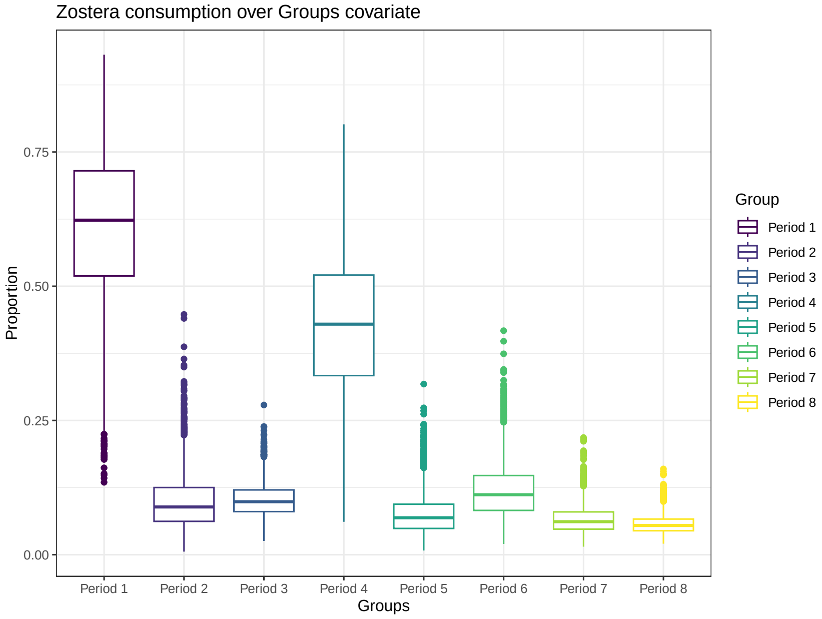


Figure 4.6.3: Boxplots showing change in consumption of *Zostera* for Geese in different periods of the year.

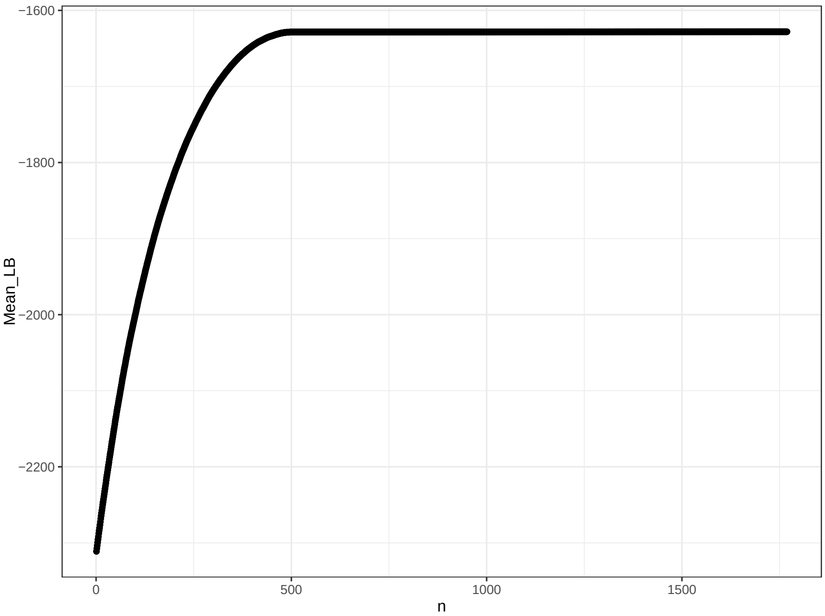


Figure 4.6.4: Lineplot showing convergence of mean lower bound values for geese example, produced using `convergence_check` function in `cosimmr`.

The posterior_predictive function can be used to produce a plot showing the posterior uncertainty intervals for this dataset. The produced plot is seen in Figure 4.6.5. We can see that 77% of the data lies within the 50% uncertainty intervals, showing that cosimmr is fitting the data adequately, although this value is above what we would expect. The posterior predictive values are a useful check of model fit and these are not available in other packages, so easy comparison is not available, but can be accessed using the posterior_predictive function in cosimmr. Groups 7 and 8 contain the outliers seen in Figure 4.6.5 and this may be a potential reason for these results. For 75% uncertainty intervals 87% of observations are inside these intervals and 93% are inside for 95% confidence intervals. From this example we can see the importance and usefulness of including covariates, as it allows for us to look at the diet of the geese over time to see how the proportions of different foods in their diets change as the season progresses. It also highlights observations that may require further scrutiny.

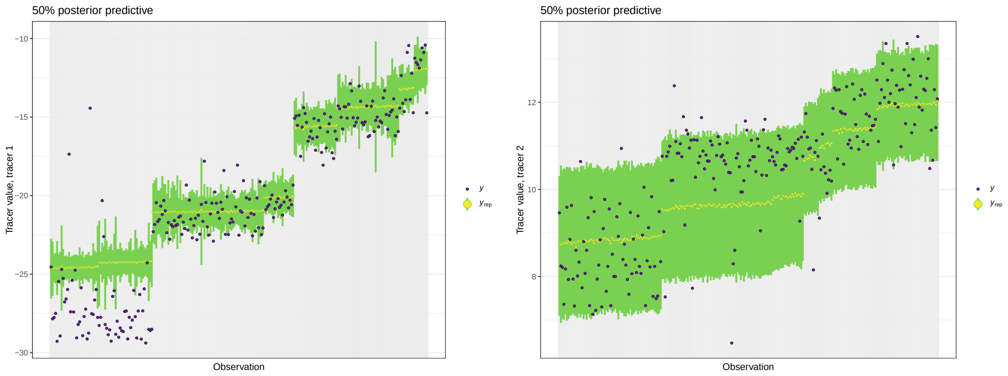


Figure 4.6.5: Plot showing posterior uncertainty intervals at the 50% level for the Geese data. The first plot shows values for tracer 1 and the second shows values for tracer 2, with plots showing the observations (points) and the posterior uncertainty intervals in green. The proportion of posterior values inside these values was 77%. Groups 7 and 8 in this example contain outliers which may be a reason for the proportion being higher than we would expect.

4.6.2 Isopod data (Galloway et al., 2014)

The second case study is the isopod dataset (*Pentidotea wosnesenskii*) from Galloway et al. (2014). Six sites were used, which varied in algal cover, and this is included as the sole covariate. Three food sources are included in this example:

Green (phylum Chlorophyta), Brown (phylum Ochrophyta), and Red (phylum Rhodophyta) algae. There are eight tracers - fatty acids instead of stable isotopes, as fatty acid signatures are shown to differ significantly between algal phyla (Galloway et al., 2012). An iso-space plot for these data is seen in Figure 4.6.6, which is 2-dimensional and can therefore only show two of the eight tracers. `cosimmr` allows for users to specify which tracers they would like to plot in the iso-space plot. More than two tracers can make it difficult to check visually that all individuals lie within the multidimensional mixing polytope so caution is needed to ensure accurate TDFs are included and all relevant food sources are included. The posterior predictive plots can be particularly helpful when using >2 tracers to discover problem observations (or tracers themselves) because these are available per tracer as opposed to per pair of tracers in the iso-plot.

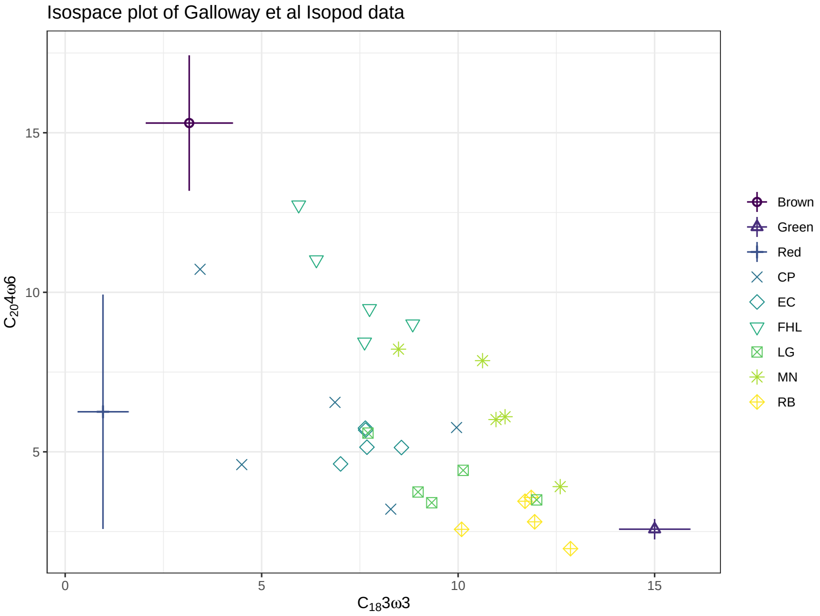


Figure 4.6.6: Iso-space plot for isopod dataset showing 6 different Sites across 2 of 8 possible tracers

As in the previous example, comparing the proportion estimates for `cosimmr` and MixSIAR (Figure 4.6.7) across different covariates levels, we can see that both are returning similar estimates, with `cosimmr` returning those estimates in a much shorter time (Table 4.6.3). A ‘normal’ run in MixSIAR (100,000 chain length, 50,000 burn-in) was enough to ensure convergence with this model. Numerical results are presented for both `cosimmr` and MixSIAR in Table 4.6.2. Slight differences in results may be due to the fact MixSIAR treats Site as a random effect vs `cosimmr` treating it as a fixed effect.

Programme	Source	25%	50%	75%
`cosimmr`	Green	0.322	0.345	0.370
MixSIAR	Green	0.362	0.401	0.437
`cosimmr`	Brown	0.214	0.248	0.287
MixSIAR	Brown	0.095	0.144	0.196
`cosimmr`	Red	0.372	0.401	0.431
MixSIAR	Red	0.411	0.454	0.495

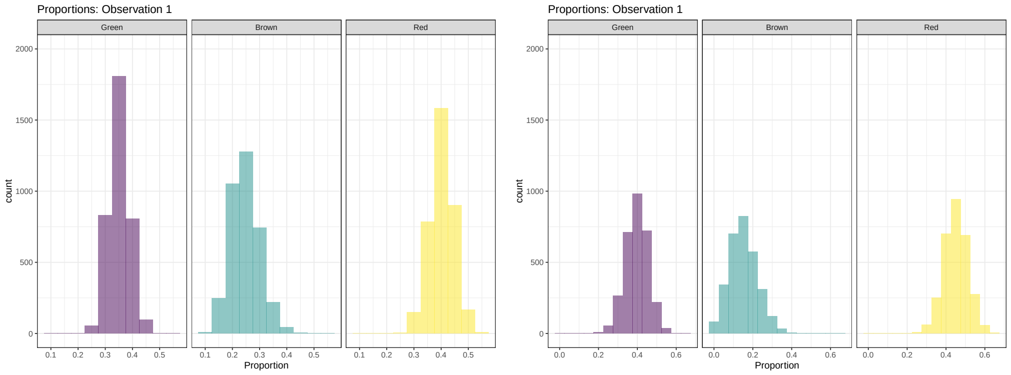
Table 4.6.2: Table showing estimates of food consumed for Observation 1 for both `cosimmr` and MixSIAR for Isopod example.

	min	lower quartile	mean	median	upper quartile	max	neval
cosimmr	418	455	587	582	748	791	10
MixSIAR	1182	1246	1268	1283	1292	1298	10

Table 4.6.3: Table showing computation time (seconds) of `cosimmr` and MixSIAR for Isopod example.

The importance of the covariate in this example is seen in Figure 4.6.8. This plot shows the difference in average consumption of Green algae across different sites. This allows us to see the importance of the included covariate and how it affects the dietary proportions of individuals at that site.

The posterior predictive plot can be produced using the `posterior_predictive` function. The resulting plot for tracer 1 can be seen in Figure 4.6.9. 59% of values are inside the 50% interval for this overall run. The posterior predictive for the other tracers can be viewed in the Appendix (Figure 4.B.5).



(a) `cosimmr`

(b) MixSIAR

Figure 4.6.7: Proportion plot showing consumption of different food sources for observation 1 for Isopod example for `cosimmr` and MixSIAR

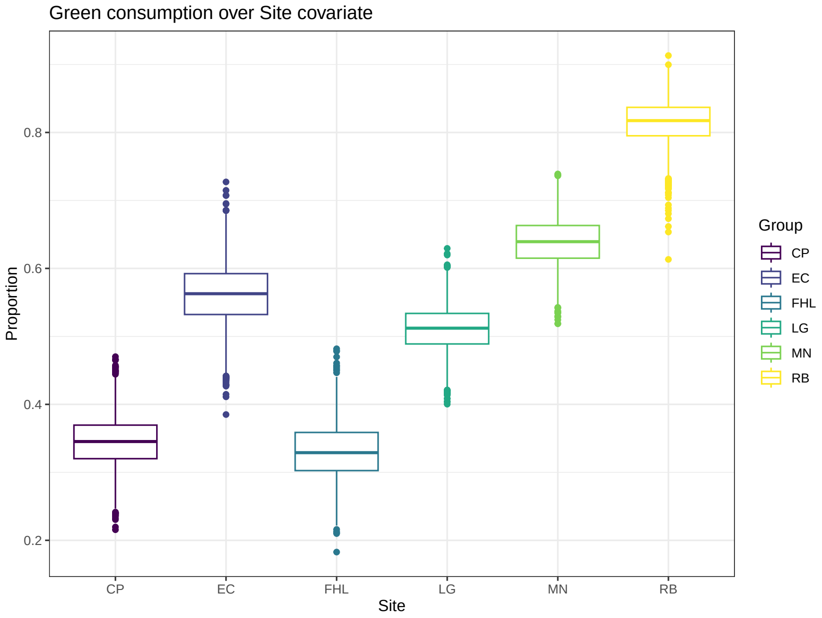


Figure 4.6.8: Boxplots showing change in Algae consumption across sites for Green Algae for Isopod example

This example highlights the computational efficiency of `cosimmr` over MixSIAR and other SIMM software. `cosimmr` is producing similar results to other SIMMs for this example, but is much quicker thanks to the use of FFVB.

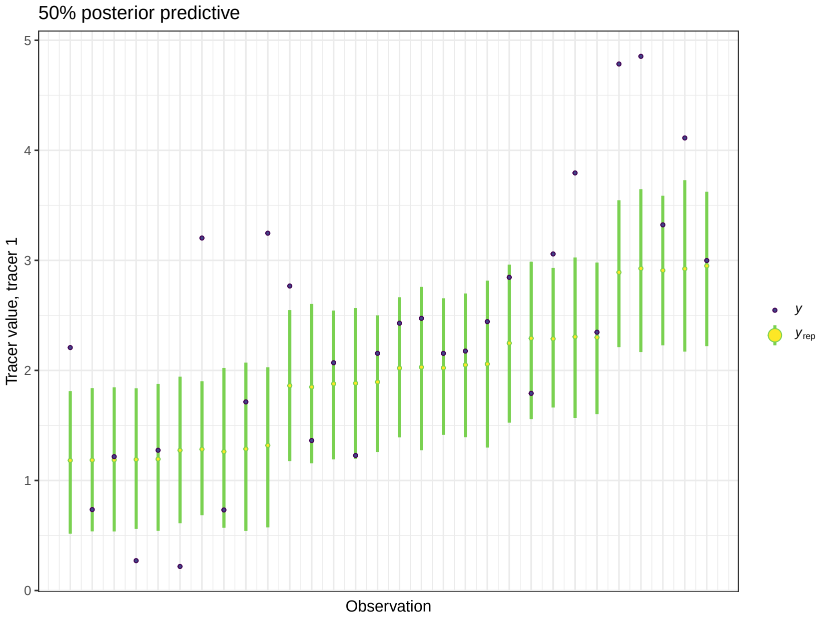


Figure 4.6.9: Plot showing posterior uncertainty intervals at the 50% level for the Isopod data for tracer 1. The proportion of posterior values inside these values was 59%.

4.6.3 Alligator data (Nifong et al., 2015)

The final example utilises alligator (*Alligator mississippiensis*) data from Nifong et al. (2015). In this example we run 8 alternative models with both `cosimmr` and MixSIAR, where each model utilises a different combination of covariates, and determine the best model fit, with the aim being that both algorithms present the same model as the best fit. The eight models are described in Table 4.6.4.

The iso-space data for this example can be seen in Figure 4.1.1. There are only two food sources in this set, Marine and Terrestrial. All food sources in this example were grouped into one of these two categories. The iso-space plot is coloured by covariate 'Length' (this covariate is used in Model 5 and Model 7).

All eight models were fitted in both `cosimmr` and MixSIAR. MixSIAR uses 'LOO' (leave-one-out cross validation) for choosing the best fitting model (Vehtari et al., 2024). We used the same package and method on results from `cosimmr` to choose the best fitting model. For both, model 5 (Length) is selected as the best model.

Model	Covariate(s)
1	NULL
2	Habitat (Fresh, Intermediate, Marine)
3	Sex (Male, Female)
4	Sclass (Small Juvenile, Large Juvenile, Sub-Adult, Adult)
5	Length (continuous effect)
6	Sex + Sclass
7	Sex + Length
8	Sex * Sclass

Table 4.6.4: Table showing different model options for Alligator example

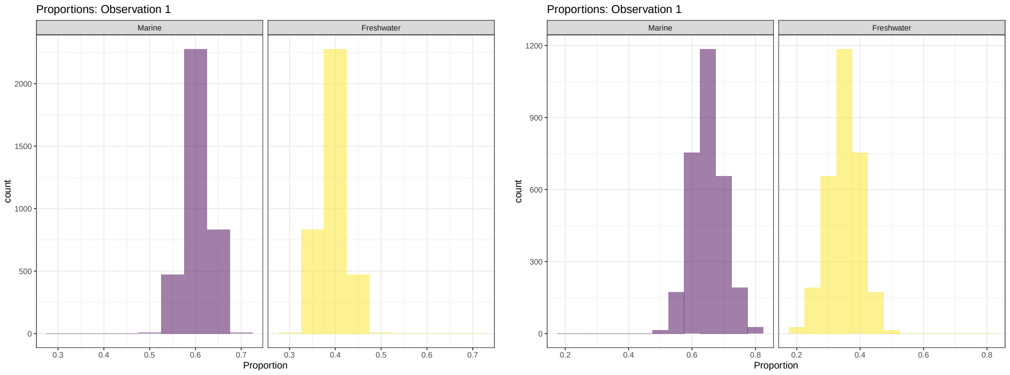
The output of this model comparison can be seen in Table 4.6.5. The error structure of `cosimmr` and MixSIAR is slightly different. MixSIAR also utilises hierarchical source fitting which is not implemented in `cosimmr`. This may explain the slight differences in results obtained.

	`cosimmr`		MixSIAR	
Model	looic	SE	looic	SE
Model 1	1990.9	31.2	1834.6	16.7
Model 2	1943.1	61.0	1747.9	28.8
Model 3	2072.8	45.6	1831.3	17.6
Model 4	1833.3	60.0	1687.5	31.8
Model 5	**1754.1**	**41.8**	**1678.3**	**31.3**
Model 6	1844.7	57.5	1689.2	31.5
Model 7	1822.5	54.6	1681.2	31.4
Model 8	1770.8	37.8	1690.4	29.8

Table 4.6.5: Table showing LOO output for `cosimmr` and MixSIAR alligator models, where looic is the LOO information criterion ($-2 \times elpd\_loo$) and SE is the standard error of looic

We can compare the output of both `cosimmr` and MixSIAR and see that they are returning comparable results. In Figure 4.6.10 we plot the estimates for Model 5 for observation 1, an individual alligator of length 186 cm. Comparing the time for both runs shows that `cosimmr` is approximately 10 times faster (See Table

4.6.6). The 'short' version of MixSIAR is a long enough run for convergence for this example.



(a) `cosimmr` (b) MixSIAR

Figure 4.6.10: Proportion plot showing consumption of different food sources for observation 1 for Alligator example for `cosimmr` and MixSIAR

	min	lower quartile	mean	median	upper quartile	max	neval
cosimmr	111	121	126	127	132	140	10
MixSIAR	1330	1349	1383	1379	1410	1449	10

Table 4.6.6: Table showing computation time (seconds) of `cosimmr` and MixSIAR for Alligator example for Model 5 run.

Figure 4.6.11 shows the predicted consumption of each food source varying with Length. The variable Length was provided to the `predict` function as a regular grid. This figure highlights why covariates can be a useful tool in SIMMs, as without covariates we get an average diet across all individuals. By including Length as a covariate, we get a much deeper insight into the animals diet. We can see that as an individual increases in length, it increases its consumption of Marine sources and consequently its consumption of Freshwater sources drops. Marine consumption increases from below 10% to above 90%. These findings agree with stomach content analysis performed by Nifong et al. (2015).

For this example we generate a posterior predictive plot, seen in Figure 4.6.12. Here for a 50% interval 42% of individuals lie inside. For a 75% interval 87% of individuals lie inside and this climbs to 93% for the 95% interval. This indicates a good level of fit for this model. Like previous examples, the posterior predictive

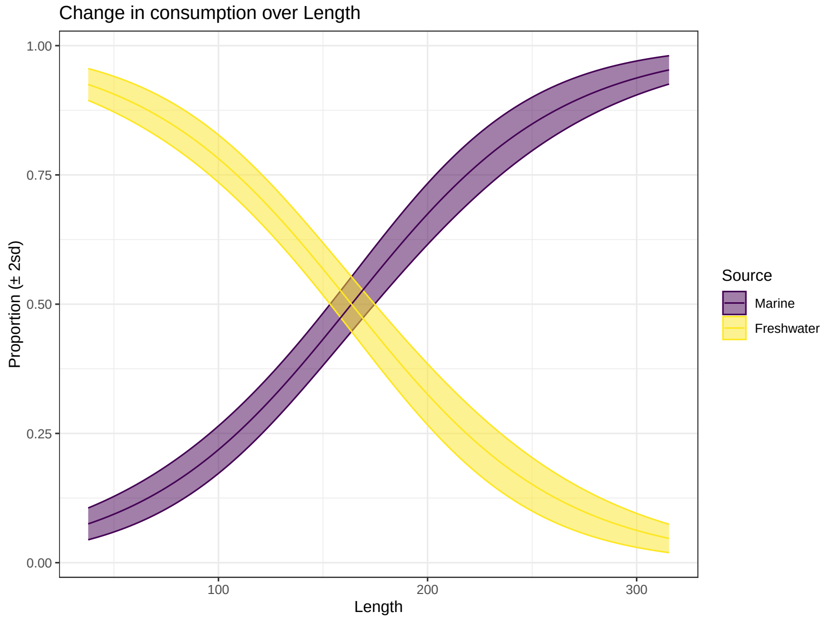


Figure 4.6.11: Plot showing the change in Freshwater and Marine consumption vs change in Length for Alligator example, where proportion is plus or minus two standard deviations.

plot highlights outliers and points that lie far outside the 50% interval. This is an indicator that these observations may require further scrutiny.

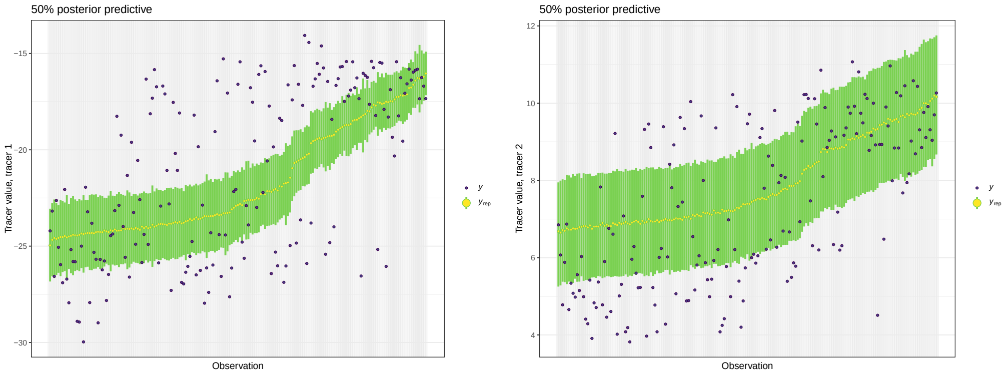


Figure 4.6.12: Plot showing posterior uncertainty intervals at the 50% level for the Alligator example for tracer 1 and 2. The proportion of posterior values inside these values was 42%.

This example highlights the computational efficiency of `cosimmr`. The ten-fold increase in performance speed of `cosimmr` over MixSIAR highlights the high value of this package for those fitting multiple models. We see that the package produces comparable results to MixSIAR, both in terms of proportional output as well as when multiple models are compared against each other. The addition of the `posterior_predictive` function provides strong guidance (not available in other packages) as to how well the model is fitting, as well as highlighting observations for further inspection.

4.7 Conclusions

Fixed Form Variational Bayes is a novel technique within the field of Stable Isotope Mixing Models. It is an optimization-based algorithm which contrasts with the sampling-based approaches traditionally used in SIMMs, such as MCMC. FFVB works by estimating a variational posterior which approximates the true posterior. Through the examples presented in this paper we have demonstrated that it performs as well as MCMC in terms of results produced, while also offering a significant speed improvement of up to one order of magnitude.

The use of FFVB over MCMC allows users to run more complex models in a shorter time. Alternatively it may allow users to compare more models across differing covariates with a view to finding one that matches the data best. We believe that this speed advantage (without loss of accuracy) is an important development for SIMMs. It is important to remember that the FFVB method only ever produces an approximation of the posterior so model checking through, e.g. posterior predictive distributions, is especially important.

We have introduced the package `cosimmr` to implement these new methods. The inclusion of covariates allows users to avoid violating the assumption of IID data. The package contains functions that allow users to make predictions for combinations of covariates not found or recorded during data collection, allowing for a deeper understanding of the system being studied. We have shown that `cosimmr` is demonstrably faster than previous packages (due to FFVB), while returning comparable results. We have designed it to be user-friendly for non-expert users, with built-in summary, plotting and predict functions. It is important that users are aware plots are generally created for an individual observation, with the option to specify the individual(s) for which to create plots built into the package functions. Other functions help users to see which of their covariates are having an impact on the diet of the animal being studied, and how well the model fits the data.

Future work on `cosimmr` could include allowing for random effects, hierarchical modelling, or source fitting - these are all options that are currently available in MixSIAR but could also be implemented using FFVB in order to speed up model fitting. Currently we don't account for uncertainty with our source samples. If the number of samples is very large then this is not an issue, but if using smaller sample sizes then it is important to account for this potential source of error. The challenge here is ensuring the optimization still converges satisfactorily despite the additional parameters. We plan to implement these options in a future evolution of the package.

Appendix

4.A Gaussian Variational Bayes with Cholesky Decomposed Covariance

We use the Gaussian Variational Bayes with Cholesky decomposed covariance algorithm of Tran et al. (2021). If we define the joint set of parameters as $\theta = (\beta, \log(\sigma^2))$ then we write our factorised variational posterior as:

$$q_\lambda(\theta) = q(\beta, \log(\sigma)))$$

where $\lambda = (\mu_\beta, \mu_\sigma, vech(L))^T$ is the set of hyper-parameters associated with the variational posteriors:

$$\begin{aligned} q(\theta) &\equiv MVN(\mu, \mathbf{\Sigma}) \\ \mu &\equiv (\mu_\beta, \mu_\sigma) \\ \mathbf{\Sigma} &\equiv (\mathbf{\Sigma}_\beta, \mathbf{\Sigma}_\sigma) \end{aligned}$$

To avoid the positive semi-definite constraints on $\mathbf{\Sigma}_\theta$ we model the Cholesky decomposition of this matrix so that $\mathbf{\Sigma}_\theta = \mathbf{L}\mathbf{L}^T$.

To start the algorithm, initial values are required for $\lambda^{(\mathbf{0})}$ (we use parenthetical super-scripts to denote iterations), the sample size S, the adaptive learning weights (β_1, β_2), the fixed learning rate ϵ_0, the threshold α, the rolling window size t_W, the maximum patience P.

Define h to be the log of the joint distribution up to the constant of proportionality:

$$h(\theta) = \log\left(p(y|\theta)p(\theta)\right)$$

and h_λ to be the log of the ratio between the joint and the VB posterior:

$$h_\lambda(\theta) = \log\left(\frac{p(\mathbf{y}|\theta)p(\theta)}{q_\lambda(\theta)}\right) = h(\theta) - \log q_\lambda(\theta)$$

The initialisation stage proceeds with:

1. Generate samples from $\kappa_\mathbf{s} \sim N_d(0, I)$ for $s = 1, ...S$
2. Compute the unbiased estimate of the lower bound gradient:

$$\begin{aligned}\widehat{\nabla}_\lambda LB(\lambda^{(0)}) = \left(\widehat{\nabla_\mu LB}(\lambda^{(0)})^T, \widehat{\nabla_{\mathbf{vech(L)}} LB}(\lambda^{(0)})^T\right)^T \\ \widehat{\nabla}_\mu LB(\lambda^{(0)}) = \frac{1}{S}\sum_{s=1}^{S} \nabla_\theta h_\lambda(\theta_s) \\ \widehat{\nabla}_{\mathbf{vech(L)}} LB(\lambda^{(0)}) = \frac{1}{S}\sum_{s=1}^{S} vech\left(\nabla_\theta h_\lambda(\theta_s)\kappa_s^T\right)\end{aligned}$$

Create estimates of θ_s

$$\theta_s = \mu^{(0)} + \mathbf{L}^{(0)}\kappa_\mathbf{s}$$

3. Set

$$\begin{aligned}\bar{g}_0 := \nabla_\lambda LB(\lambda^{(0)}) \\ \bar{\nu}_0 := \bar{g}_0^2 \\ \bar{g} = g_0 \\ \bar{\nu} = \nu_0\end{aligned}$$

4. Set $t = 1$, patience $= 0$, and 'stop $=$ FALSE'.

Now the algorithm runs with:

1. Generate $\kappa_s \sim q_{\lambda^{(t)}(\theta)}$ for $s = 1, ...S$. Recalculate $\mu^{(\mathbf{t})}$ and $\mathbf{L}^{(\mathbf{t})}$ from $\lambda^{(\mathbf{t})}$
2. Compute the unbiased estimate of the lower bound gradient:

$$g_t := \widehat{\nabla}_{\lambda\mathbf{LB}}(\lambda^{(t)}) = \left(\widehat{\nabla}_{\mu} LB(\lambda^{(t)})^T, \widehat{\nabla}_{\mathbf{vech(L)}} LB(\lambda^{(t)})^T\right)^T$$

where

$$\widehat{\nabla}_{\mu} LB(\lambda^{(t)}) = \frac{1}{S}\sum_{s=1}^{S} \nabla_\theta h_\lambda(\theta_s)$$

$$\widehat{\nabla}_{\mathbf{vech(L)}} LB(\lambda^{(t)}) = \frac{1}{S}\sum_{s=1}^{S} vech\left(\nabla_\theta h_\lambda(\theta_s)\kappa_s^T\right)$$

with $\theta_s = \mu^{(t)} + \mathbf{L}^{(t)}\kappa_{\mathbf{s}}$

3. Compute:

$$\begin{aligned} v_t &= g_t^2 \\ \bar{g} &= \beta_1 \bar{g} + (1-\beta_1) g_t \\ \bar{v} &= \beta_2 \bar{v} + (1-\beta_2) v_t \end{aligned}$$

4. Update the learning rate:

$$l_t = min(\epsilon_0, \epsilon_0 \frac{\alpha}{t})$$

and the variational hyper-parameters:

$$\lambda^{(t+1)} = \lambda^{(t)} + l_t \frac{\bar{g}}{\sqrt{\bar{v}}}$$

5. Compute the lower bound estimate:

$$\widehat{LB}(\lambda^{(t)}) := \frac{1}{S}\sum_{s=1}^{S} h_{\lambda^{(t)}}(\theta_s)$$

6. If $t \geq t_W$ compute the moving average LB

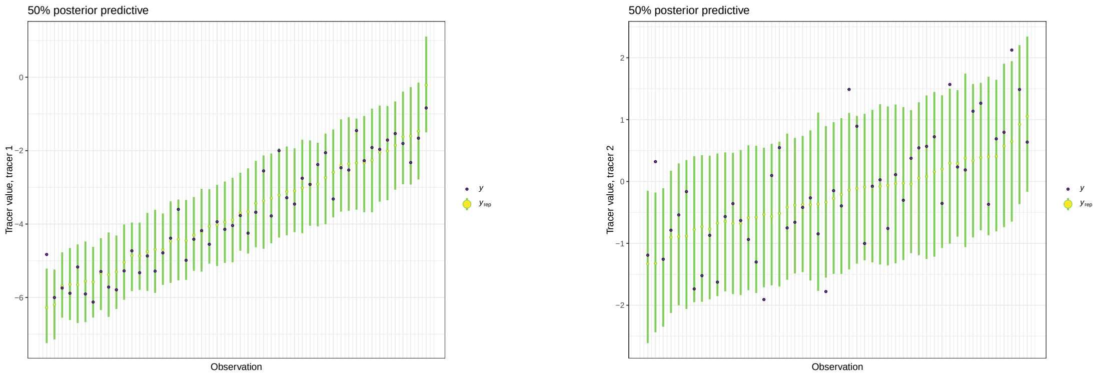
$$\overline{LB}_{t-t_W+1} := \frac{1}{t_W}\sum_{k=1}^{t_W} \widehat{LB}(\lambda^{(t-k+1)})$$

   If $\overline{LB}_{t-t_W+1} \geq \max(\overline{LB})$ patience $= 0$, else patience $=$ patience $+1$

7. If patience $\geq$ P, ‘stop $=$ TRUE‘

8. Set $t := t + 1$

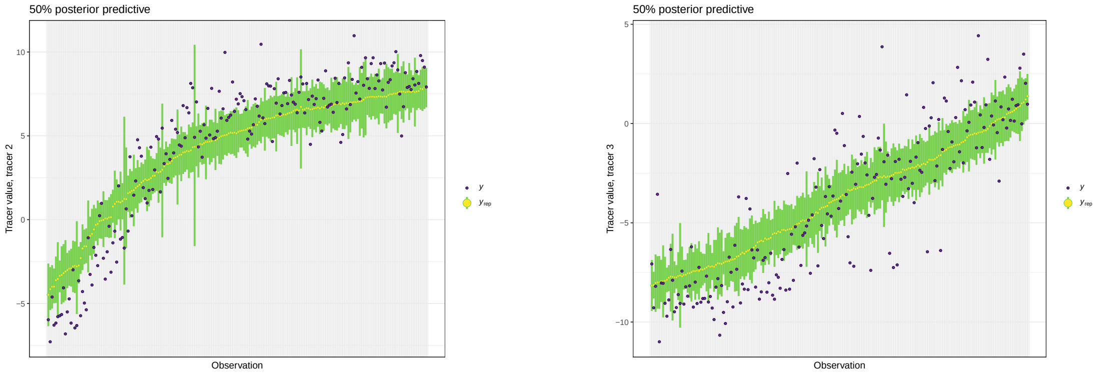
4.B Further plots



(a) Tracer 2

(b) Tracer 3

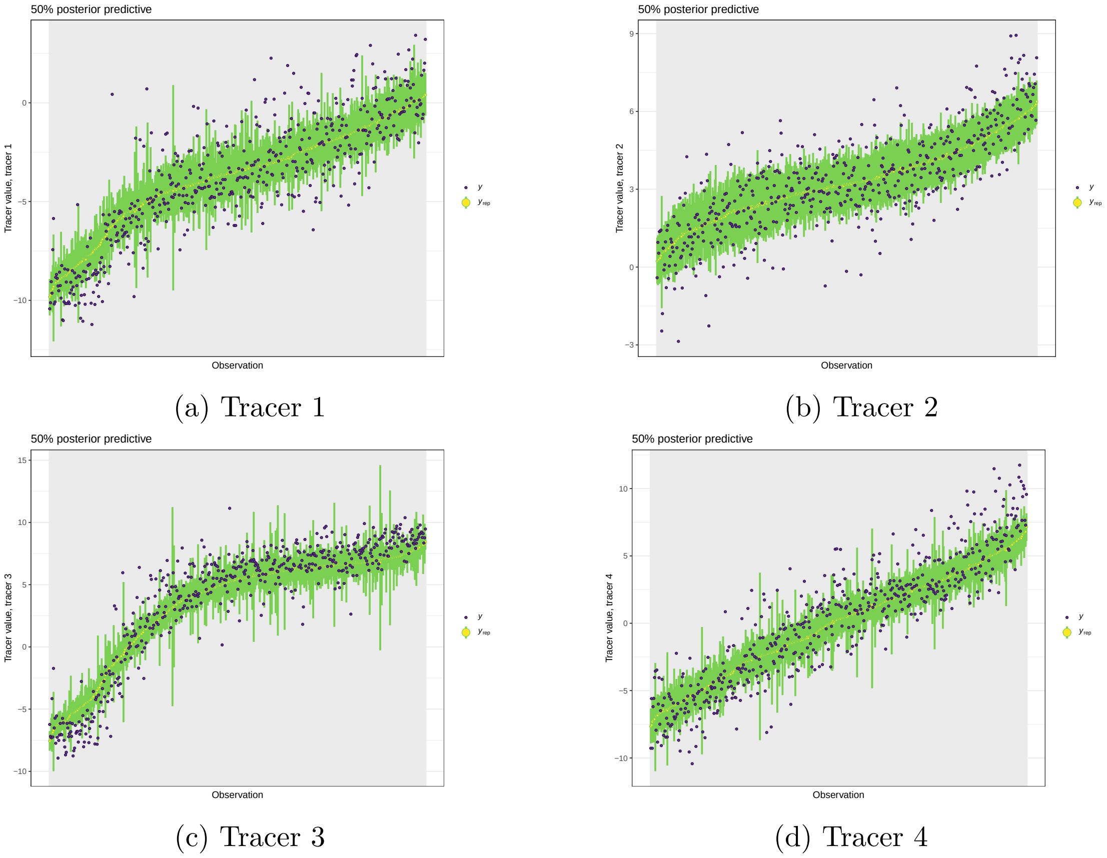
Figure 4.B.1: Plot showing posterior uncertainty intervals at the 50% level for the ‘Low’ simulated model run for tracer 1 and tracer 2. The proportion of posterior values inside these values was 93%.



(a) Tracer 2

(b) Tracer 3

Figure 4.B.2: Plot showing posterior uncertainty intervals at the 50% level for the 'Medium' simulated model run for tracers 2 and 3. The proportion of posterior values inside these values was 51%.



(a) Tracer 1

(b) Tracer 2

(c) Tracer 3

(d) Tracer 4

Figure 4.B.3: Plot showing posterior uncertainty intervals at the 50% level for the ‘High’ simulated model run for tracers 1-4. The proportion of posterior values inside these values was 60%.

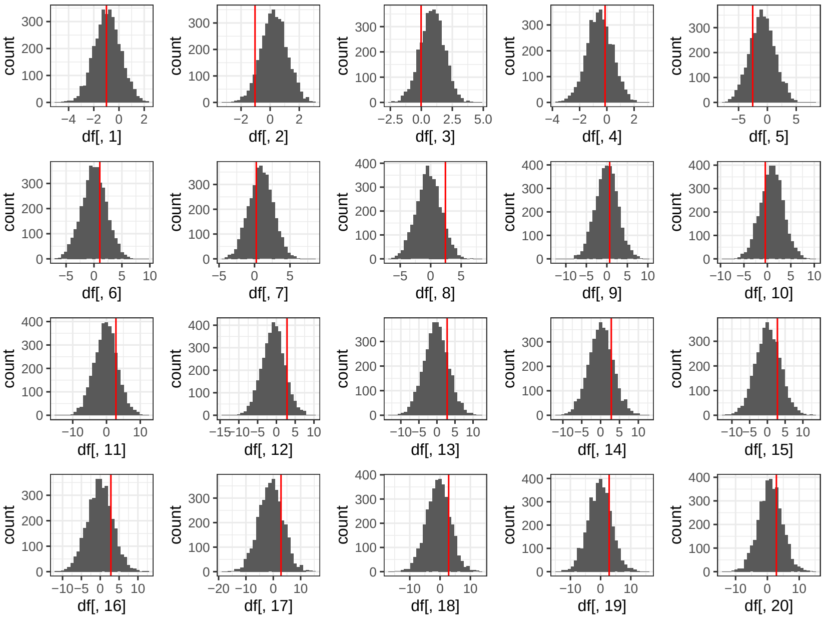
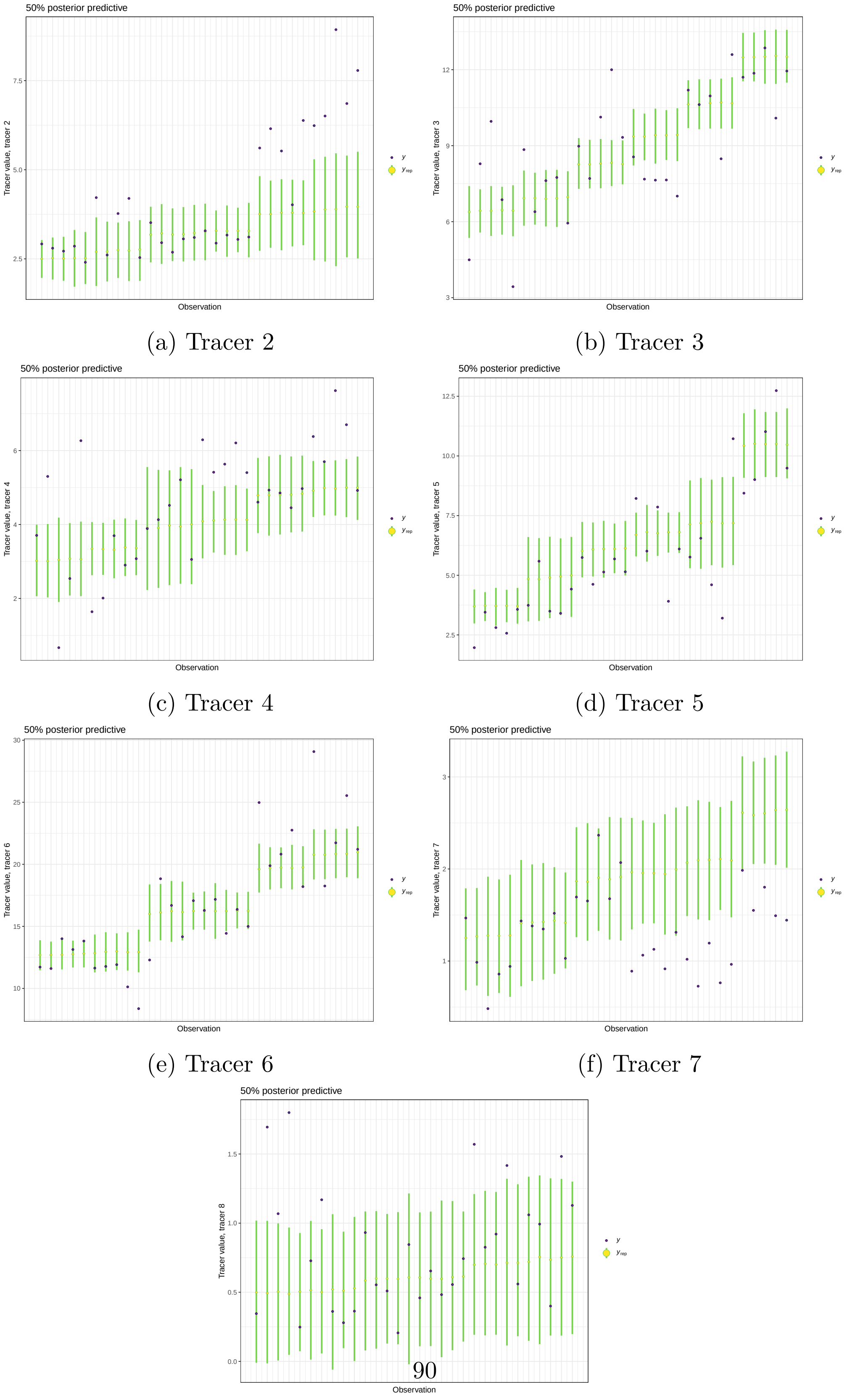


Figure 4.B.4: Histograms showing posterior samples for beta values generated via `cosimmr` for the ‘Medium’ example, and red line showing ‘true’ value of β used to generate the mixture data.



(a) Tracer 2

(b) Tracer 3

(c) Tracer 4

(d) Tracer 5

(e) Tracer 6

(f) Tracer 7

(g) Tracer 8

Figure 4.B.5: Plot showing posterior uncertainty intervals at the 50% level for the Isopod data for tracers 2-8. The proportion of posterior values inside these values was 59%.

CHAPTER 5

cosimmrSTAN: an R package for fitting Stable Isotope Mixing Models using STAN

In this chapter, we discuss `cosimmrSTAN`, an R package for fitting Stable Isotope Mixing Models using Hamiltonian Monte Carlo via the external software package STAN. The package has been designed to allow inclusion of richer covariate structures including random effects and related hierarchical modelling techniques.

5.1 Introduction

Stable Isotope Mixing Models (SIMMs) are a very common tool which allow users to study the proportional contribution that each food makes to an animal's diet. SIMMs are widely used in ecology; recent examples include an examination of the diet (and seasonal shift in the diet) of White sharks (*Carcharodon carcharias* Lipscombe et al., 2024), identifying the location Camelids originated from in the Late Pleistocene/Early Holocene Atacama desert (Ugalde et al., 2024), and arsenic detection in fish (Lescord et al., 2022). These models are also popular in geology and pollution studies, where they may be referred to as 'end-member analysis' (Hooper et al., 1990), ‘mass balance analysis’ (Miller et al., 1972), or ‘source apportionment models’ (Hopke, 1991) depending on the field in question.

The simplest mathematical equation for a SIMM is as follows:

$$y = \sum_{k=1}^{K} p_k s_k + \epsilon$$

where y refers to the mixture value (for example, the $\delta^{13}C$ value for the consumer we wish to study), p_k are the proportions contributed by each food source k (of K total sources), s_k is the source tracer value for each source k, and ϵ is a residual term. p_k is the parameter we are most interested in, and it is p_k that we wish to estimate. The equation is made more complex when including multiple observations and isotopes, Trophic Discrimination Factors (TDFs; Inger and Bearhop, 2008) and concentration dependence (Phillips and Koch, 2002) in order to make it more biologically accurate. The full equation of the model with these terms included is presented in Section 5.2.1.

TDFs account for the fact that consumers may selectively assimilate lighter or heavier isotopes into their tissues. This is important to account for in a biological system; they may have less relevance in other applications. TDFs can be calculated in the lab, however this can be difficult depending on the species being studied. Values from the literature for related species are often used, and `SIDER` (Healy et al., 2018) is an R package developed to allow for the estimate of TDFs based on species’ relatedness. TDFs are usually added on to the source values s_k to shift them in iso-space.

Concentration dependence accounts for the fact that sources are composed of different proportional amounts of each isotope (Phillips and Koch, 2002). If this is not considered then, for example in a two-isotope model, the assumption is that every source contributes the same amount for each element. However, often a food can be rich in one isotope and poor in another, and so instead concentration dependence is calculated by accounting for the elemental concentration within that food source. Concentration dependence is included as a multiplicative term on p.

The addition of covariates in SIMMs can help users to understand the relationships between individual organisms when they vary by one or more measured factors.

However, adding covariates can result in the model taking a long time to run. Markov chain Monte Carlo (MCMC) has traditionally been used in software that has been developed for the running of SIMMs, such as MixSIAR (Stock et al., 2018) and SIAR (Parnell et al., 2010). While MixSIAR allows for the running of complex models, containing both fixed and random effects, as well as hierarchically fitted sources, it can be quite slow, with MCMC sometimes requiring millions of samples in order to reach convergence. We have instead implemented these models in STAN (Carpenter et al., 2017) in order to take advantage of STANs VB function, which allows for models to run much more quickly than they would using MCMC. Specifically these algorithms have been implemented in an R (R Core Team, 2021) package, which can be found at `https://www.github.com/emmagovan/cosimmrSTAN`, utilising the rstan (Stan Development Team, 2024) functionality. The use of STAN allows allows for more complex models to be run using VB. We have been able to build on the `cosimmr` package to include random effects, as well as hierarchical source fitting, using STANs VB functionality. STAN simplifies the process of incorporating these complex terms into the model, as we only specify the model structure without managing the underlying algorithm, which would have been challenging to encode directly.

5.2 Statistical Approaches to fitting Stable Isotope Mixing Models

5.2.1 Statistical Model behind SIMMs

The more complex model for SIMMs, with TDFs and concentration dependence accounted for, is as follows:

$$y_{ij} = \sum_{k=1}^{K} p_k(\mathbf{x}_i) q_{jk}(s_{ijk} + c_{ijk}) + \epsilon_{ij}$$

Where:

- y_{ij} are the mixture/consumer tracer values of individual i for tracer (commonly isotope) j,

- $p_k(\mathbf{x}_i)$ are the proportions of each source k contributing to the mixture value at each covariate value $\mathbf{x}_i$ where $\mathbf{x}_i$ is an L-vector of covariate values for individual i. We commonly shorten these to p_{ik}.
- q_{jk} represents the concentration dependence for tracer j on source k,
- s_{ijk} is the consumed source value by individual i of the food source k on tracer j,
- c_{ijk} is the trophic discrimination factor of individual i for source k on tracer j
- ϵ_{ij} is the residual error for individual i on tracer j

We index individuals as $i = 1, \ldots, N$, tracers as $j = 1, \ldots, J$, and sources as $k = 1, \ldots, K$. We assume there are $l = 1, \ldots, L$ covariates so that $\mathbf{x}_i = \{x_{i1}, \ldots, x_{iL}\}$. We use the term 'Mixture' to refers to an individual's set of tracer/isotope values for which we wish to know the proportional compositions. The examples we describe in Section 5.4 all use stable isotopes as tracers, but fatty acids can be used instead depending on the system being studied. We assume that $\epsilon_{ij} \sim N(0, \sigma_j^2)$, $s_{ijk} \sim N(\mu_{s,jk}, \sigma^2_{s,jk})$, and $c_{ijk} \sim N(\mu_{c,jk}, \sigma^2_{c,jk})$. $\mu_{s,jk}, \sigma_{s,jk}$ are assumed fixed, but we provide an option whereby they can be learnt via the `cosimmrSTAN_load` function. $\mu_{c,jk}, \sigma_{c,jk}$ are assumed to be fixed. σ_j is given a weakly informative gamma prior.

We marginalise across the source parameters, as proposed in Moore and Semmens (2008) to produce a likelihood that is more complex but less computationally intensive:

$$y_{ij} \sim N\left(\frac{\sum_{k=1}^{K} p_{ik} q_{kj} \mu_{sc,kj}}{\sum_{k=1}^{K} p_{ik} q_{kj}}, \frac{\sum_{k=1}^{K} p_{ik}^2 q_{kj}^2 \sigma^2_{sc,kj}}{\sum_{k=1}^{K} p_{ik}^2 q_{kj}^2} + \sigma_j^2\right)$$

where $\mu_{sc,kj} = \mu_{s,kj} + \mu_{c,kj}$ and $\sigma^2_{sc,kj} = \sigma^2_{s,kj} + \sigma^2_{c,kj}$. Occasionally an additional multiplicative term can be included on the first variance term above, as proposed by Stock et al. (2018), to account for a consumer's dietary specialisation. We restrict this term to lie between 0 and 1. The term is useful, or even necessary, when $\sigma^2_{s,jk}$ and $\sigma^2_{c,jk}$ are too large compared to the variability of the mixtures, and often improves the quality of the fit. The ecological motivation for such a term is clear: the individuals are sub-sampling or averaging over the sources to assimilate their food. However the mathematics of this extra term are harder to justify, since its inclusion no longer corresponds to the unmarginalised model presented at the start of this section. We model this parameter on the logit scale to enforce the range constraint. We call the parameter ξ and apply a $N(0, 2.5^2)$ prior when it is required that centers the value around 0.5 with a standard deviation that covers the majority of the (0,1) range. The new equation when including ξ is:

$$y_{ij} \sim N\left(\frac{\sum_{k=1}^K p_{ik}q_{kj}\mu_{sc,kj}}{\sum_{k=1}^K p_{ik}q_{kj}}, \frac{\sum_{k=1}^K p^2_{ik}q^2_{kj}\sigma^2_{sc,kj}}{\sum_{k=1}^K p^2_{ik}q^2_{kj}} \frac{1}{1+\exp(-\xi_j)} + \sigma^2_j\right).$$

As stated above the dietary proportions p_{ik} terms are of key interest. They must be constrained so that $\sum_k p_{ik} = 1$. Thus we use a Centralised Log-Ratio (CLR; Aitchison, 1986) link function so that:

$$[p_{i1}, ...p_{iK}] = \left[\frac{\exp(f_{i1})}{\sum_S \exp(f_{is})}, \dots, \frac{\exp(f_{iK})}{\sum_S \exp(f_{is})}\right]$$

We can then place a prior distribution or other structure through the f_{ik} terms. We cover some of the structures we implement below.

5.2.2 Fitting Random or Fixed Covariates

Covariates can be set as fixed or random in `cosimmrSTAN`. The way this is modelled is as follows:

$$f_{ik} = \mathbf{X}_i\beta_{0,l_1k} + \mathbf{Z}_i\beta_{1,l_2k}$$

where $\mathbf{X}$ is a matrix of l_1 fixed effects and $\mathbf{Z}$ is a matrix of l_2 random effects, for a total of $l_1 + l_2 = L$ covariates. The priors on β are: $\beta_{0,r,k} \sim N(0, 1)$, for $r = 1, \dots, l_1$ fixed effects (we use this prior based on Stock et al. (2018), where the same prior on fixed effects is used) and $\beta_{1,t,k} \sim N(0, \omega^2_k)$, for $t = 1, \dots, l_2$ random effects, and $\omega_k \sim t^+_1(0, 1)$.

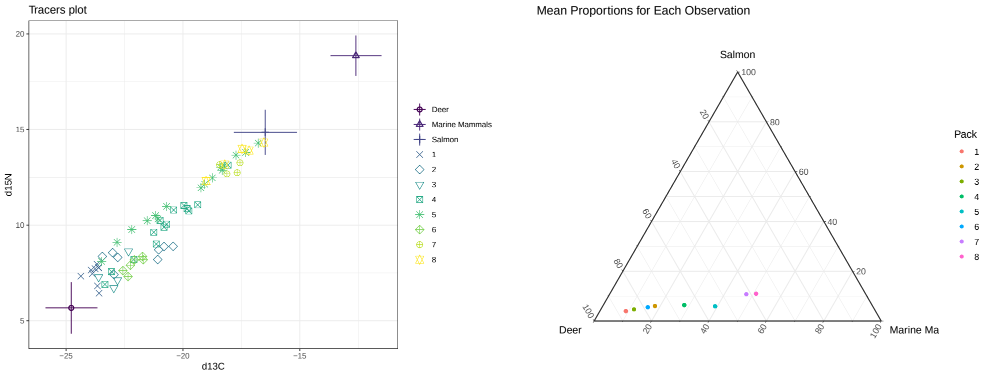
The full Bayesian model that we fit is therefore:

$$\begin{aligned}\pi(p_{ik}, \sigma^2_j, \xi_j, \beta_0, \beta_1, \omega_k \mid y_{ij}, \mu_{sc,kj}, \sigma^2_{sc,kj}, q_{kj}, \mathbf{X_i}, \mathbf{Z_i}) \propto & \prod_{i=1}^{N}\prod_{j=1}^{J}\prod_{k=1}^{K} \pi(y_{ij} \mid p_{ik}, \mu_{sc,kj}, \sigma^2_{sc,kj}, q_{kj}, \xi_j, \sigma^2_j) \\ & \times \prod_{j=1}^{J} \pi(\sigma^2_j) \\ & \times \prod_{j=1}^{J} \pi(\xi_j) \\ & \times \prod_{s=1}^{l_1}\prod_{k=1}^{K} \pi(\beta_{0sk}) \\ & \times \prod_{t=1}^{l_2}\prod_{k=1}^{K} \pi(\beta_{1sk} \mid \omega_k) \\ & \times \prod_{k=1}^{K} \pi(\omega_k)\end{aligned}$$

where all notation is defined as above.

When fitting SIMMs, a distinction is often made between looking at the data in 'iso-space' vs 'p-space'. Iso-space is the plot of the raw data showing the tracer values on their original scale, as well as the sources and the mixtures. Once the model is fitted, users can plot and look at the posterior dietary proportions (i.e. $p_{i,k}$) and this is known as p-space. In Figure 5.1 we show examples of the wolves data (discussed in Section 5.4.2) in both iso-space (left plot) and p-space (right plot).

Adding in covariates allows the user a much more detailed look at the way these proportions change alongside the covariate values. In `cosimmrSTAN` users can plot the posterior β values obtained by the model using the `plot` function. Non-zero values indicate that the covariate related to the β value in question significantly impacts the consumption level of the food source being looked at. Positive values for β indicate increased consumption of the food source with that covariate, but the size of the effect is not directly interpretable due to the CLR transform. To ameliorate the problem the `predict` function can used to create predictions in



(a) `iso-space plot`

(b) p-space plot

Figure 5.1: Plot showing wolves data plotted in iso-space and average results from `cosimmrSTAN` for each pack in p-space

'p-space' which are easier to interpret. However we do caution users with these plots since there is an implicit assumption that other covariates are fixed which may given a non-biologically accurate interpretation. `summary` can also be used to provide outputs in 'p-space' for individual observations. Providing simple interpretations of the effect of covariates from complex SIMMs remains an ongoing research challenge.

Whether to use fixed or random effects in a model can be a difficult choice. Gelman (2007) notes that there are several different definitions on what is a fixed or random effect. Solving the problem of choosing fixed versus random effects is outside the scope of this paper, but there are several useful considerations which can help to indicate the choice for a specific model. Harrison et al. (2018) suggests that if we assume that the groups we have sampled are a subset of all possible groups, then we should use a random effect. This gives us the power to predict outside the groups we have sampled. If instead, we are comparing two possible treatments, for example the growth of plants at 10°C versus 20°C, then we are only considering these two temperatures, we have sampled all possible groups, and so we should use fixed effects. However, Gelman (2007) finds this unhelpful, and instead recommends to always use random effects, in order to borrow strength between groups.

In the field of social sciences, Clark and Linzer (2015) proposes the general guide-line that if variation is primarily within groups i.e. the groups are similar to one another, then choosing either fixed or random effects will not greatly impact results - unless there is high correlation between the independent variable and the group effects. However if there is less variation within groups, then random effects are recommended if there are few groups, or few observations within groups, and when correlation is low. Otherwise fixed effects are preferred.

Much of the discussion in the literature has focused on the frequentist treatment of fixed and random effects. From a Bayesian viewpoint, the only difference in our model between fixed and random effects is whether the parameter ω is given as data or estimated with a prior distribution as part of the model fit. Fixing the ω parameter represents a very strong prior belief in the variability of the regression coefficients, which may over-ride the information in the data. Clearly this may be preferable in situations when such information exists. However more generally it seems sensible to estimate this parameter with a prior distribution that places probability mass on areas which are reasonable given our knowledge about the data. This is why we tend to prefer the random effects structure and would advise users to default to this case when including covariates. The situation is made more complicated in our model as the parameter space on f will mostly be contained in the range (-3, 3) due to the CLR transformation, and so β is likely to lie in a smaller range. Thus our default prior of $N(0, 1)$ is likely to be sensible for many situations and the distinction between fixed and random effects less important as in standard modelling situations. In any case, we follow the advice in Gelman (2006) and use the half-Cauchy distribution for the prior on ω.

In summary, much depends on the goals of the user and the differences between them are unlikely to make a strong difference to the resulting posterior estimates of p. In `cosimmrSTAN` a user can have both fixed and random effects in the same model if they wish, or choose one or the other as the goals of their research requires. Furthermore, thanks to the speed improvements offered by `cosimmrSTAN`, it is possible to run covariates as both fixed and as random in separate models, and to then compare results to see if it has an impact on a particular system. The choice often remains, however, one of personal modelling preference.

5.2.3 Fitting Sources

There are three options for users when supplying source data in `cosimmrSTAN`. The first option is to supply a mean and standard deviation for each food source, and these are used as μ_s and σ_s as in the equation in Section 5.2.1. This treats the estimates as fixed and is the method used in `simmr` (Govan et al., 2023) and `cosimmr` (Govan et al., 2024). This assumes that users know the true mean and standard deviation. Alternatively, users can supply a sample mean and sample standard deviation, and number of samples for each food source, and the source data can be fitted as follows (Ward et al., 2010):

$$\mu^s_{k,j} \sim N\left(m^s_{k,j}, \frac{s^s_{k,j}}{\sqrt{n_k}}\right)$$

$$\sigma^s_{k,j} = \frac{1}{\sqrt{\frac{T_{k,j}}{s^2 s_{k,j}(n_k-1)}}}$$

$$T_{k,j} \sim \chi^2(n_k)$$

Where $m^s_{k,j}$ is the sample mean for food source k and tracer j, $s^s_{k,j}$ is the sample standard deviation for food source k and tracer j, and n_k is the number of samples of food source k taken. Note that if n_k is large enough (> 10000) then this method is equivalent to fixing μ^s and σ^s at the sample values provided by the user and is essentially the same as the first option (Stock et al., 2018).

Users can alternatively supply raw source data which is fitted as follows (Ward et al., 2010):

$$y^s_{k,j} \sim N(\mu^s_{k,j}, \Sigma^s_k)$$

$$\mu^s_{k,j} \sim N(0, 1000)$$

$$\Sigma_k = \text{diag}(\tau) \cdot \Omega \cdot \text{diag}(\tau)$$

$$\Omega \sim LKJCorr(\eta)$$

$$\tau \sim t^+_1(0, 2.5)$$

Where $y^s_{k,j}$ is the food source measurement for food source k on tracer j. The prior on $\mu_{k,j}$ is set to match MixSIAR. An LKJ prior (Lewandowski et al., 2009) is used for the correlation matrix Ω. With $\eta = 1$ set as the default prior, this is an uninformative prior that makes only weak assumptions about the correlation between tracers in a source. Users can change the prior value on η if they wish. The LKJ prior is often used for correlation matrices and is available as a function in STAN. It is important to note that it is not recommended to fit raw source data when using compositional tracer data such as fatty acids, as the assumption of normality does not hold (Stock et al., 2018).

However the source data are provided, the sources are estimated as part of the `cosimmrSTAN_load` function, following Ward et al. (2010) and thus similar to how they are fitted in MixSIAR. Fitting sources (either as raw source data or by supplying a sample mean and standard deviation) removes the assumption that we know the true value of the source values, and allows us to account for uncertainty in measurement (Ward et al., 2010). However one disadvantage of this method is that it does result in an increased computational load as it means there are more parameters to be estimated.

5.3 STAN Algorithms

STAN allows for the fitting of statistics models via Hamiltonian Monte Carlo (HMC) or Variational Bayes. Hamiltonian Monte Carlo (Betancourt and Girolami, 2015), like more standard Markov chain Monte Carlo (MCMC), is a sampling algorithm which aims to sample from the joint posterior distribution of the parameters given the data. HMC differs from MCMC in that it introduces auxiliary 'momentum' variables, which direct the exploration of the state space. This means that instead of randomly exploring the space like in standard MCMC, these 'momentum' variables ensure that a certain section of the state space is explored, which means there is a higher probability of accepting the new state, and slow random explorations are avoided (Brooks et al., 2011).

In the package `cosimmrSTAN` there are several parameters in the `cosimmr_stan` function that users can use to change their HMC run. These are referred to as `mcmc_control` and the options within are `iterations`, `chains`, and `cores`. `iterations` refers to the number of iterations in each chain, set at a default value of 10000, `chains` refers to the number of Markov chains, set at a default value of 4, and `cores` refers to the number of cores used to run MCMC in parallel. This is set as a default of 1 but if multiple cores are available it will result in the model running more quickly. Users may wish to increase the number of iterations if a model is not converging. Convergence can be checked using the `convergence_check` function. For MCMC this returns a table with a count of the number of r-hat (Gelman and Rubin, 1992) values between 1.0 and 1.1 and the number greater than 1.1. If there are many values are far away from 1 then it indicates that the model is not converging. It may be helpful to increase the number of iterations if this is the case.

Variational Bayes (VB) is another option in STAN, and the one we have set as our default in the package `cosimmrSTAN`. STAN uses automatic differentiation variational inference (ADVI; Kucukelbir et al., 2015). VB generally works by approximating the true posterior with a variational posterior. For ADVI a Gaussian variational posterior is used. The aim in VB to minimise the Kullback-Leibler (KL) divergence between the true posterior and the variational posterior. This is done by maximising the evidence lower bound (ELBO). VB can offer a significant speed advantage over MCMC due to the fact it is optimisation-based versus MCMC's sampling-based techniques. We have previously found that VB offers up to a one order of magnitude increase in speed (Govan et al., 2024). This was specifically for Variational Bayes with Cholesky Decomposed Variance (Titsias and Lázaro-Gredilla, 2014; Tan and Nott, 2018) and was implemented in R via Rcpp (Eddelbuettel and François, 2011). In `cosimmrSTAN` we have instead implemented VB via STAN and by default the package uses the 'fullrank' algorithm in STAN. This uses a full-rank covariance matrix. Both Variational Bayes with Cholesky Decomposed Variance and 'fullrank' VB in STAN allow for correlations between parameters due to the covariance matrix used. While VB is not guaranteed to converge (Yao et al., 2018), we have found no issues with convergence for the examples we have looked at.

Users can adjust different VB parameters when using the `cosimmr_stan` function, by altering the options within `vb_control`. These include: the threshold that the algorithm uses to determine convergence (`tot_rel_obj`), the number of iterations (iter), the number of samples (n_samples), the maximum number of iterations to use in the warm-up adaptation period (adapt_iter), and the specific type of VB algorithm used ('meanfield' or 'fullrank'). 'fullrank' is set as the default, as it allows for correlation between variables due to the covariance matrix used.

The Pareto k diagnostic (Vehtari et al., 2015) is produced by STAN when running VB algorithms. If it is high then the model may not be converging. A table of the mean, minimum, maximum, and counts of the number of Pareto k values below 0.5, between 0.5 and 0.7, between 0.7 and 1.0, and above 1.0 can be produced using the convergence_check function in cosimmrSTAN. A value of between 0.7 and 1.0 may indicate problems with the model convergence (Vehtari et al., 2017) and values above 1.0 indicate that caution is needed with the model. If there are a lot of values produced over 1.0 then we recommend decreasing tot_rel_obj or increasing n_samples or adapt_iter, or using MCMC. However it is important to note that the Pareto-k-diagnostic is based around leave-one-out cross-validation, and so values above 1.0 may not mean the model is wrong, it may be due to the fact there are low numbers of data points relative to the number of parameters being estimated. In the examples we run later in this paper we find that we obtain Pareto-k-values of between 0.7 and 1.2 but these models are still returning results that are comparable with the results obtained by MixSIAR which uses MCMC.

The Variational Bayes algorithm has a built-in stopping function which occurs when the change in the ELBO falls below a certain threshold (which users can alter by changing tot_rel_obj). This stopping rule ensures that the algorithm has reached a result, but it is important for the user to be mindful that it does not guarantee correct results. However, we have found no problems for the examples we have tested.

5.4 Working Examples

In this section we will present three examples of models run using the cosimmrSTAN package, with fixed and random effects. The data used in these examples are available in the cosimmrSTAN package and all code for running examples is available at github.com/emmagovan/cosimmrSTAN_paper.

5.4.1 Fixed Effects model

The first example will be a fixed effects model using geese data from Inger et al. (2006). The fixed covariate in this example is named ‘Time’. This refers to 8 different distinct periods over 2 winters in which the geese data were collected. Therefore this covariate is treated as a factor. To begin we load in the `cosimmrSTAN` package and read in the data.

```
> geese_data = cosimmrSTAN::geese_data
> Time = as.factor(geese_data$groups)
> formula = geese_data$mixtures ~ Time

> in_geese = with(
        geese_data,
        cosimmrSTAN_load(
        formula,
        source_names = source_names,
        source_means = source_means,
        source_sds = source_sds,
        correction_means = correction_means,
        correction_sds = correction_sds,
        concentration_means = concentration_means,
        scale_x = FALSE))
```

The `cosimmrSTAN_load` function has multiple options for users - they can supply source means and standard deviations, or they can supply raw source data and have it fitted. If they supply source means and standard deviations they can use these directly or treat them as sample means and standard deviations and fit them by setting `hierarchical_fitting = TRUE`. `scale_x` allows users to choose whether or not to scale their x values. `scale_x` is set to `TRUE` as default but it is set to `FALSE` here as our covariate is a factor.

The next step is to create an iso-space plot. It is recommended that users create this plot and check it before proceeding to run a SIMM. It is important to check that all mixtures lie within the mixing polygon created by the outermost error bars on each food source, otherwise this indicates that perhaps TDFs are not being accounted for, or that a source is missing. Mixtures lying within the mixing polygon does not guarantee that all sources are accounted for or that the TDFs are correct, but if this assumption is violated then there is an issue that needs to be corrected. Phillips et al. (2014) discusses in more detail best practices for running SIMMs and guidance for interpreting iso-space plots. Users have the option to colour mixtures by a covariate. The code for generating the iso-space plot (with mixtures coloured by the covariate Time) is shown below. If users wish to colour by a specific covariate then the name of the covariate is specified by `cov_name`.

```
> plot(in_geese, cov_name = "Time")
```

The iso-space plot for this example is shown in Figure 5.1. We can see that all the mixtures lie within the mixing polygon and can see the dietary shift across the different time periods.

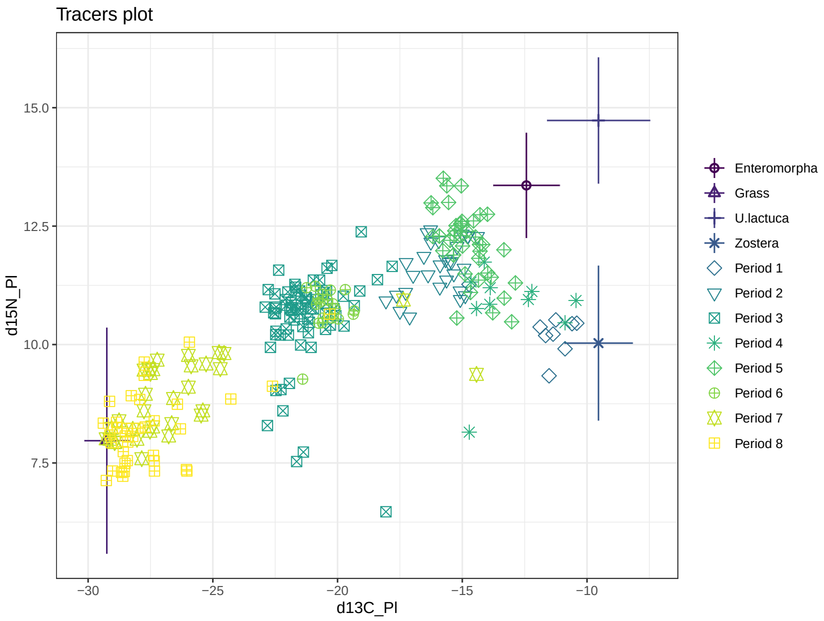


Figure 5.1: Iso-space plot for Geese example showing food sources and mixtures, coloured by Time.

We can then run the model using the code below:

```
> out_geese = cosimmr_stan(in_geese)
```

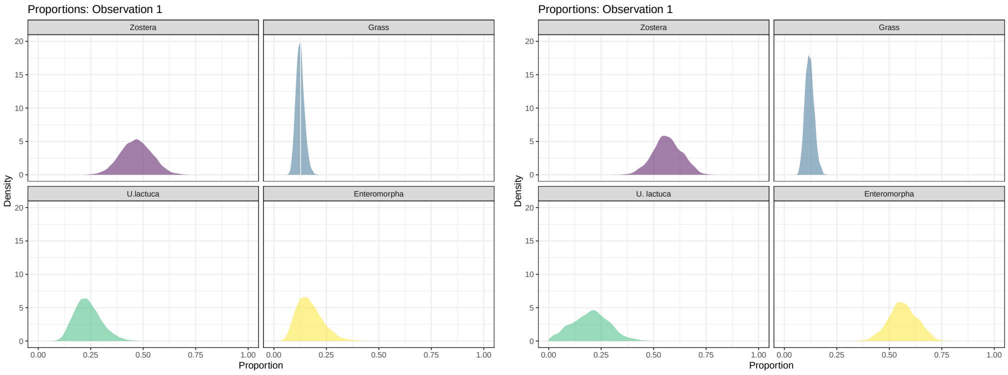
The cosimmr_stan function defaults to using the VB algorithm but users can instead choose type = "STAN_MCMC" if they wish to use the MCMC algorithm instead. The algorithm also defaults to not including ξ but this can be included by setting error_type = "processxresidual". Other options within the cosimmrSTAN function include mcmc_control and vb_control which are both discussed in greater detail in Section 5.3.

Once the output object is created, users can create multiple different plots. A histogram or density plot of the proportions of each food source being consumed can be created (prop_histogram, prop_density), density plots or boxplots of β values for fixed or random covariates can be created (beta_fixed_boxplot, beta_fixed_density, beta_random_boxplot, beta_random_density), a lineplot or boxplot showing the source consumption over different values of a covariate can be created (covariates_plot), or a density plot showing ω values for each of the K food sources can be created (omega_density). The code below is used to create a proportion plot for observation 1, seen in Figure 5.2a, a proportion plot for observation 250 is shown in Figure 5.3 and a plot of β values associated with the Time covariate. This creates a plot for all eight periods but we just show the plot for period 8 in Figure 5.4 and omit the others for brevity.

```
> plot(out_geese,
       type = c("prop_histogram",
                "beta_fixed_histogram),
        obs = c(1,250)
```

We can see in Figure 5.2 that MixSIAR produces similar results to cosimmrSTAN in terms of proportion estimates, but if we look at Table 5.1 we see that the Variational Bayes algorithm in cosimmrSTAN returns results over 110 times faster than MixSIAR.

Figure 5.3 shows the proportion of each source consumed by observation 250, a goose in period 8. We can see that these individuals are consuming a lot of grass,



(a) `cosimmrSTAN`

(b) MixSIAR

Figure 5.2: Density plot showing proportion of different food sources for observation 1 for geese example for `cosimmrSTAN` and MixSIAR

Model	Lower quartile	Mean	Median	Upper quartile
cosimmrSTAN VB	66.7	67.4	67.3	68.1
cosimmrSTAN MCMC	1480.3	1504.7	1488.9	1497.9
MixSIAR	7289.3	7804.3	7471.9	8360.4

Table 5.1: Table showing computation time (seconds) of `cosimmrSTAN` and MixSIAR (‘long’ run needed for convergence) for the geese example, showing the lower quartile, mean, median, and upper quartile. There were 10 evaluations of each function.

and if we look at the β plot in Figure 5.4 we can see that the β value for grass for period 8 is positive, which indicates that this period is consuming more grass than other periods.

We can also see summary statistics for observation 1, produced using the ‘summary’ function. ‘statistics’ are produced below but users can also choose to produce ‘quantiles’ or ‘correlations’.

```
>summary(out_geese, type = "statistics")
    Summary for Observation 1


                mean    sd
P(Zostera)      0.461 0.075
P(Grass)        0.123 0.019
```

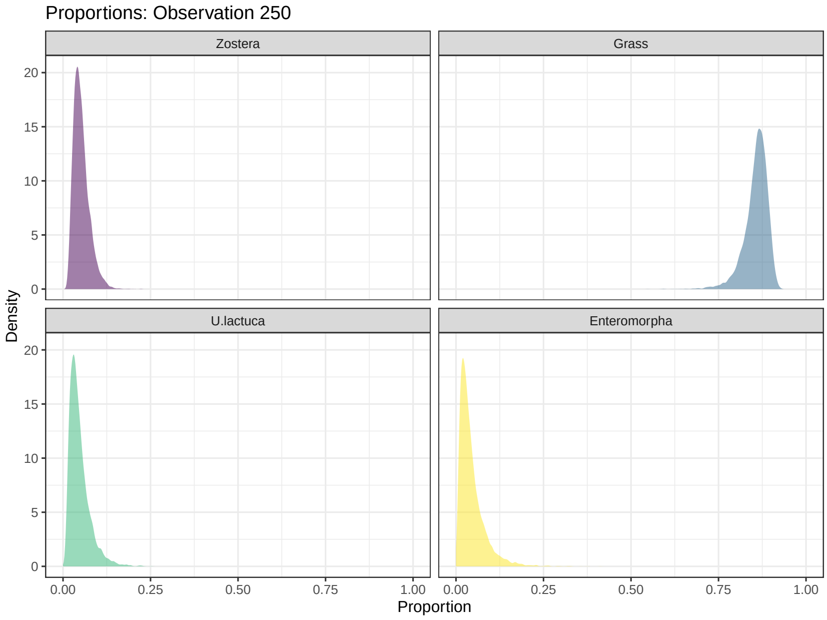


Figure 5.3: Proportion plot for Geese observation 250 showing proportions of each source consumed.

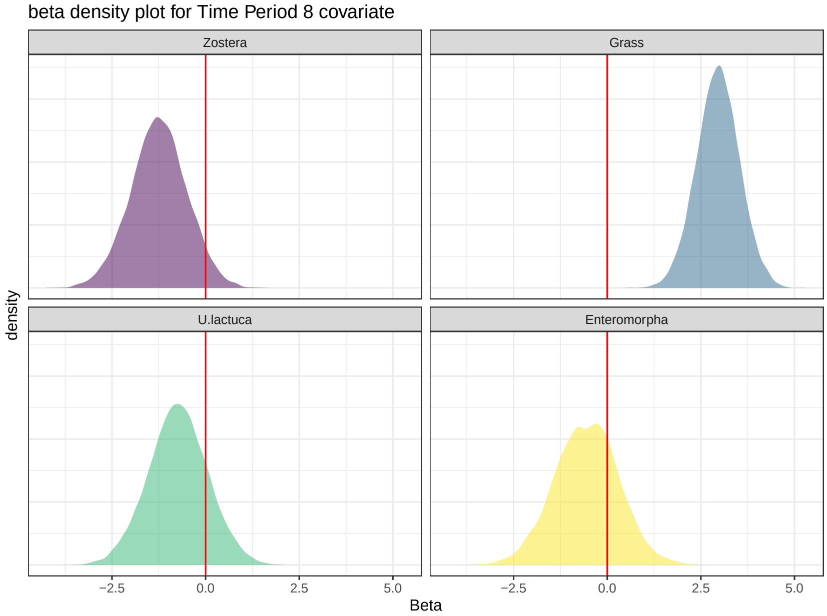


Figure 5.4: Density plot for Geese example showing beta values for each food source for Period 8. The value for Grass is positive, indicating that there is more consumption of Grass in Period 8.

```
P(U.lactuca)     0.234 0.064
P(Enteromorpha)  0.182 0.066
sd_d13C_Pl       1.443 0.096
sd_d15N_Pl       0.081 0.094
```

We can create a boxplot showing the difference in source consumption across different levels of the covariate. The code for this is shown below:

```
> plot(out_geese,
       type = "covariates_plot",
       one_plot = FALSE,
       cov_name = "Time)
```

The resulting boxplot is seen in Figure 5.5. This plot shows the change in consumption of the Grass food source. Plots for all food sources are generated but are omitted here for brevity. In this plot we can see that grass consumption is highest in periods 7 and 8, and how it varies across the different time periods, going from a less significant part of the diet at certain time periods to becoming the majority of the diet at others.

In this example we see that `cosimmrSTAN` produces comparable results to MixSIAR, but is much faster. We can see the impact of the Time covariate on the diet of the geese through β plots as well as by looking at boxplots comparing consumption of different sources over all 8 periods. This is important as it allows us to see what sources are important for the geese at different times over winter.

5.4.2 Random Effects model

In this example we will run a random effects model with the wolves data adapted from Semmens et al. (2009). This data set looks at wolf consumption of three food sources: deer, marine mammals, and salmon. The data are from 8 gray wolf (*Canis lupus*) packs in British Columbia, Canada. We will treat 'Pack' as a random effect for this example. Like in the previous example, we treat Pack as a factor. To begin we load in the `cosimmrSTAN` package and read in the data.

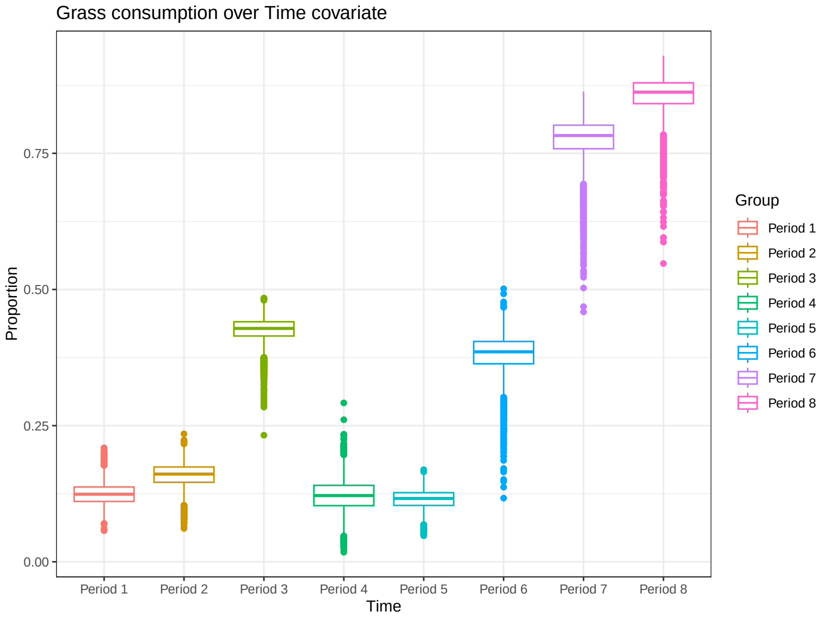


Figure 5.5: Boxplot for geese example showing change in Grass consumption over each time period. Consumption of grass is highest in Periods 7 and 8.

```
> library(cosimmrSTAN)
> wolves_data = cosimmrSTAN::wolves_data
> Pack = as.factor(wolves_data$pack)
> formula = wolves_data$y ~ (1|Pack)
> in_wolves = with(wolves_data, cosimmrSTAN_load(formula,
                            source_names = source_names,
                            source_means = s_mean,
                            source_sds = s_sd,
                            correction_means = c_mean,
                            correction_sds = c_sd))
```

It is recommended that users produce an iso-space plot and examine it before running a SIMM. The iso-space plot for this example is produced by running the code below. The plot is seen in Figure 5.6. It is important that all mixtures lie within the mixing polygon.

```
> plot(in_wolves, cov_name = "Pack")
```

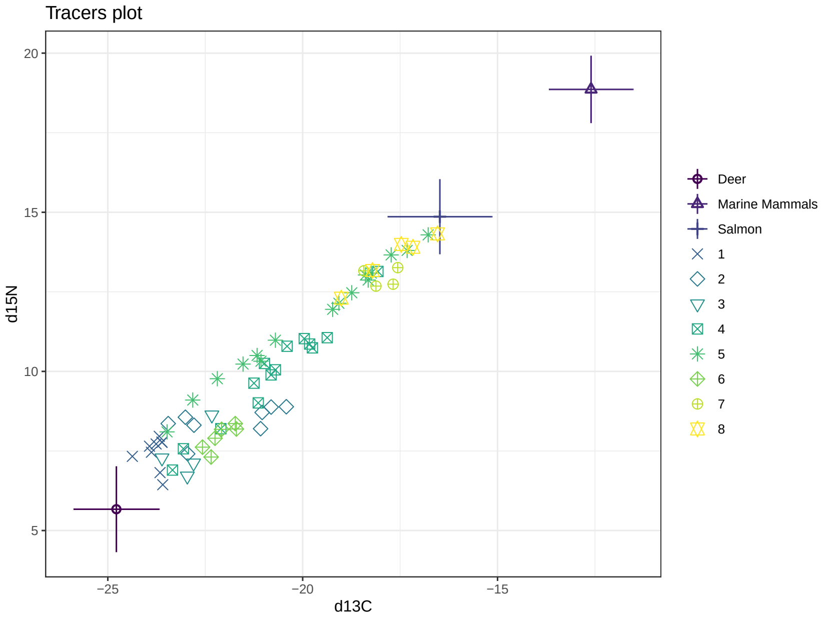


Figure 5.6: Iso-space plot for Wolves example showing Deer, Marine Mammal, and Salmon food sources. Mixtures are coloured by Pack.

Users can then run the model using the `cosimmr_stan` function as shown below.

```
> out_wolves = cosimmr_stan(in_wolves)
```

The output object can then be used to generate summaries and plots. We first create a summary showing 'statistics' for Observation 1. Users can specify multiple observations if they wish, otherwise the function defaults to observation 1.

```
> summary(out_wolves, type = "statistics")

Summary for Observation 1

                        mean    sd
P(Deer)                0.869 0.037
P(Salmon)              0.040 0.038
P(Marine Mammals)      0.091 0.031
sd_wolves_data$y.d13C 1.143 0.170
```

```
sd_wolves_data$y.d15N 0.955 0.189
```

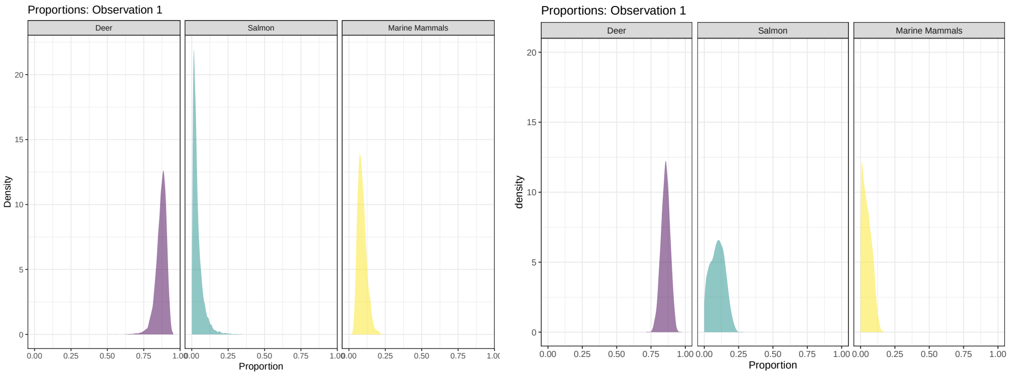
From the ‘statistics’ produced we can see that for Observation 1 (an individual in Pack 1) the majority of their diet is comprised of deer, with salmon and marine mammals making up a smaller proportion of their diet.

Users can create different plots using the `plot` function. We can see the proportions plot for both `cosimmrSTAN` as well as MixSIAR in Figure 5.7. These show the proportions of deer, salmon, and marine mammal consumed by individual 1. From examining these plots we can see that `cosimmrSTAN` returns comparable estimates to MixSIAR in terms of the proportion of each food source consumed, but if we look at Table 5.2 we can see that `cosimmrSTAN` returns those proportions over 100 times faster when using the Variational Bayes algorithm. Slight differences in results may come from the slightly different error structures used by `cosimmrSTAN` and MixSIAR. As a default `cosimmrSTAN` does not include the multiplicative ξ term discussed in Section 5.2.1 but for this model MixSIAR does include it.

```
> plot(out_wolves,
        type = c("prop_histogram",
                 "beta_random_histogram",
                 "covariates_plot"),
        one_plot = FALSE,
        cov_name = "Pack")
```

Model	Lower quartile	Mean	Median	Upper quartile
cosimmrSTAN VB	17.0	21.3	21.1	24.8
cosimmrSTAN MCMC	408.4	446.2	475.4	482.2
MixSIAR	2141.7	2377.8	2167.2	2643.8

Table 5.2: Table showing computation time (seconds) of `cosimmrSTAN` and MixSIAR (‘normal’ run (100,000 chains and 50,000 burn-in) needed for convergence) for the wolves example, showing the lower quartile, mean, median, and upper quartile. There were 10 evaluations of each function.



(a) `cosimmrSTAN`

(b) MixSIAR

Figure 5.7: Density plot showing proportion of different food sources for observation 1 for wolves example for `cosimmrSTAN` and MixSIAR

Figure 5.8 shows the β values corresponding to Pack 1. We can see that the deer values are especially removed from zero, which indicates that deer consumption in Pack 1 is impacted by the fact that the individual is in Pack 1.

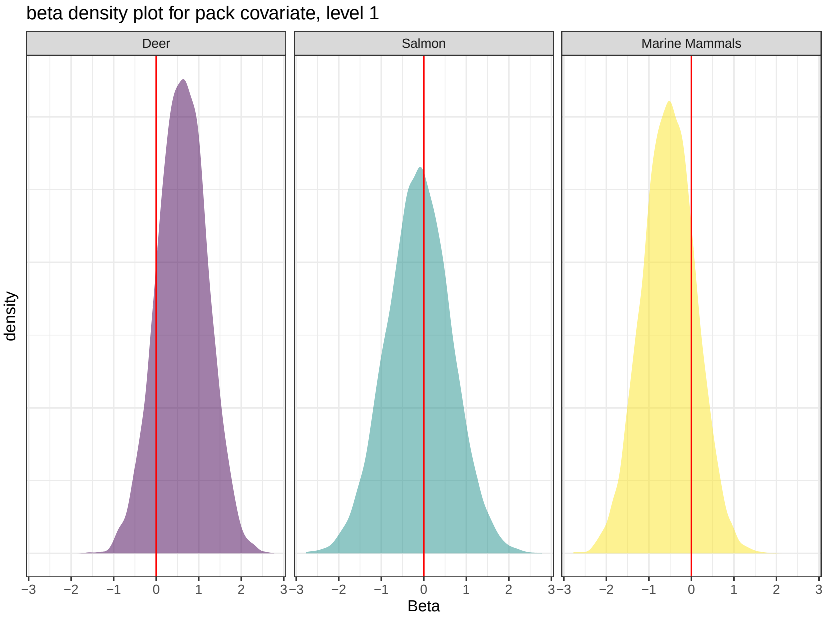


Figure 5.8: Density plot for wolves example showing β values corresponding to Pack 1. The deer value is especially removed from 1, indicating more deer is consumed by individuals in Pack 1.

Figure 5.9 shows the change in deer consumption over the Pack covariate. We can see how consumption changes across the packs and that individuals in packs 1, 2, 3 and 6 appear to be consuming the most deer. Similar plots can be produced for all of the food sources, but these are omitted for brevity.

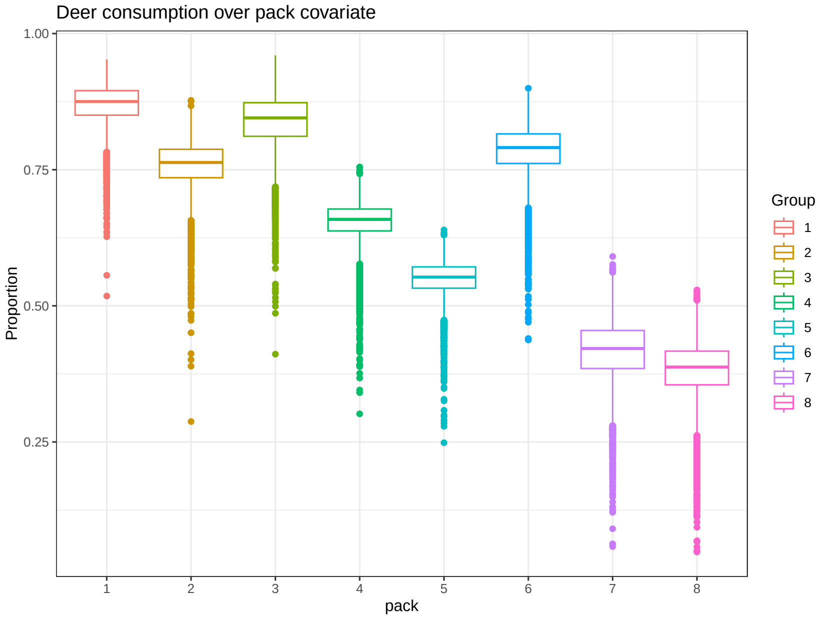


Figure 5.9: Covariates plot for wolves example showing change in Deer consumption over Pack covariate.

Users can also plot the ω_k values for each of their $k = 1, \dots, K$ food sources. The code for this is as follows:

```
> plot(out_wolves, type = "omega_density")
```

The resulting plot in shown in Figure 5.10. We can see the ω values for each of the three foods sources. A higher value here indicates more variability in consumption of the source between packs, i.e. different packs consume this source differently. A lower value indicates less variability, so packs are more similar in their consumption of this source. In this example all food sources seem to be similarly variable across packs.

In this example we can again see that `cosimmrSTAN` is capable of obtaining results similar to MixSIAR, in a much shorter time frame. The inclusion of pack as a

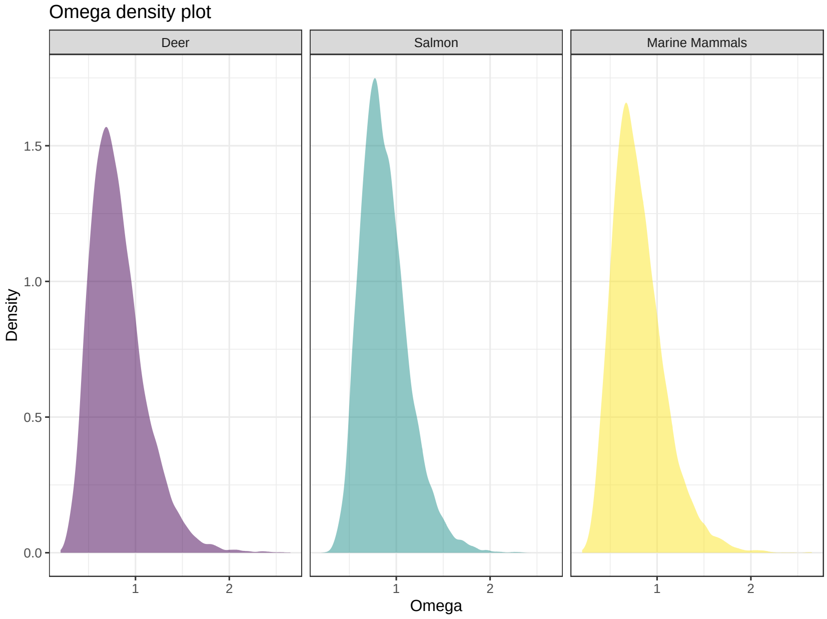


Figure 5.10: Density plot for wolves example showing ω value for each of Deer, Salmon, and Marine Mammals food sources. ω describes the overall variability of source consumption between groups.

covariate allows us to see the difference in diet across different wolf packs, and to see what food sources may be important to different wolf populations. Looking at ω values allows us to note which food sources vary the most between packs.

5.4.3 Mixed Effects Model

We will finally run a mixed effects model using the alligator data from (Nifong et al., 2015). This dataset contains data collected on *Alligator mississippiensis* in Georgia, USA. As well as isotope data, stomach content analysis was also carried out on these animals. The data set includes Length as a fixed covariate and ‘Sclass’ (size class) as a random covariate. There are just two food sources, ‘Marine’ and ‘Terrestrial’, and all of the food that the alligators were observed eating is grouped into one of these two sources.

To begin we load in the `cosimmrSTAN` package and read in the data. This model is not the best fitting model for the Alligator data. As shown in Govan et al. (2024), the model which only includes Length as a fixed covariate is selected as the best model when using either MixSIAR or `cosimmr`. However, for illustrative purposes, to show the capability of the `cosimmrSTAN` package to handle multiple covariates of both fixed and random type this model has been chosen.

```
> library(cosimmrSTAN)
> alligator_data = cosimmrSTAN::alligator_data
> Length = alligator_data$length
> Sclass = as.factor(alligator_data$sclass)
> formula = alligator_data$mixtures ~ Length + (1|Sclass)
> in_alli<-with(alligator_data,
                cosimmrSTAN_load(formula,
                                 source_names = source_names,
                                 source_means = source_means,
                                 source_sds = source_sds,
                                 correction_means = TEF_means,
                                 correction_sds = TEF_sds))
```

The iso-space plot can be generated using the `plot` function. The code for generating an iso-space plot in `cosimmrSTAN` is shown below and the iso-space plot generated is shown in Figure 5.11. This plot is coloured by the covariate Length. It is important that all of the mixtures lie within the mixing polygon created by the food sources and so it is highly recommended that users create and check this plot before running their model.

```
plot(in_alli, cov_name = "Length")
```

The data can then be run through the `cosimmr_stan` function. This function defaults to using the Variational Bayes algorithm. The code to run the model is as follows:

```
out_alli = cosimmr_stan(in_alli)
```

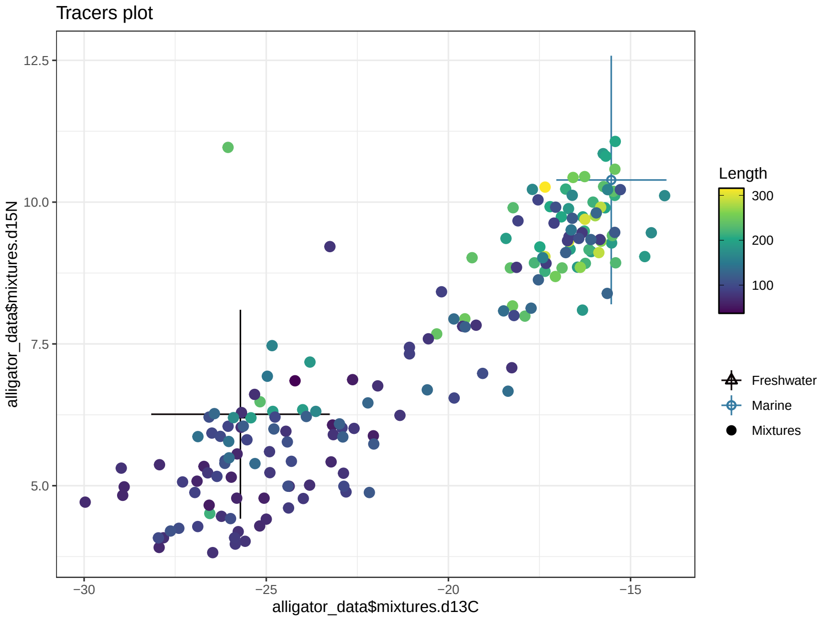
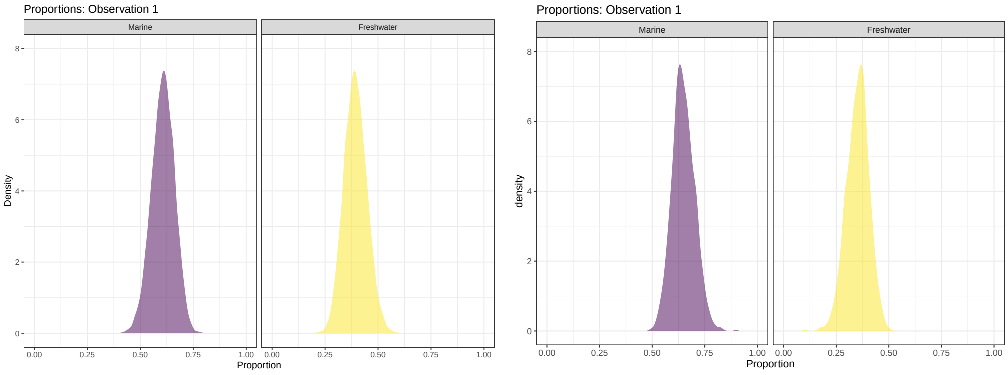


Figure 5.11: Iso-space plot showing alligator example. There are two food sources, Marine and Freshwater. Consumers are coloured by Length.

Once the model is run users can create different plots, such as the plot of proportions, or the density plot of the β values, both fixed and random. The code for this is shown:

```
> plot(out_alli,
    type = c("prop_histogram",
             "beta_fixed_density",
             "beta_random_density"))
```

The proportion plot can be seen in Figure 5.12. This shows the estimate of each food source consumed by observation 1, which is an alligator of length 186cm, and size class ‘Adult’. This plot shows estimates produced by both `cosimmrSTAN` and MixSIAR. We can see that they are giving similar results in terms of the estimated proportions. The time taken for each model run can be seen in Table 5.3, and here we can see that `cosimmrSTAN` is producing similar estimates to MixSIAR, but is producing them over 70 times faster when using the Variational Bayes algorithm.



(a) `cosimmrSTAN` (b) MixSIAR

Figure 5.12: Density plot showing proportion of marine and freshwater consumption estimates for observation 1 for alligator example for `cosimmrSTAN` and MixSIAR

Model	Lower quartile	Mean	Median	Upper quartile
cosimmrSTAN VB	35.6	36.4	36.1	37.1
cosimmrSTAN MCMC	417.2	439.5	419.6	467.7
MixSIAR	2562.0	2582.5	2590.6	2619.2

Table 5.3: Table showing computation time (seconds) of `cosimmrSTAN` and MixSIAR (‘short’ run needed for convergence) for alligator example, showing the lower quartile, mean, median, and upper quartile. There were 10 evaluations of each function.

A histogram showing estimates of β values for the ‘Length’ covariate can be seen in Figure 5.13 and for the ‘Sclass’ covariate in Figure 5.14 for the ‘Small Juvenile’ category. We can see that the values for Length are not centred around zero for either food source, indicating that ‘Length’ impacts the amount of Marine or Freshwater sources being consumed.

We can see a plot of the ω values in Figure 5.15. This shows the variation in food consumption between size classes. We can see that the variation is the same in both food sources as there are only two food sources. If consumption of one source increases then the other has to decrease.

A lineplot showing the change in consumption of both freshwater and marine sources as the Length of an alligator increases is shown in Figure 5.16. This plot

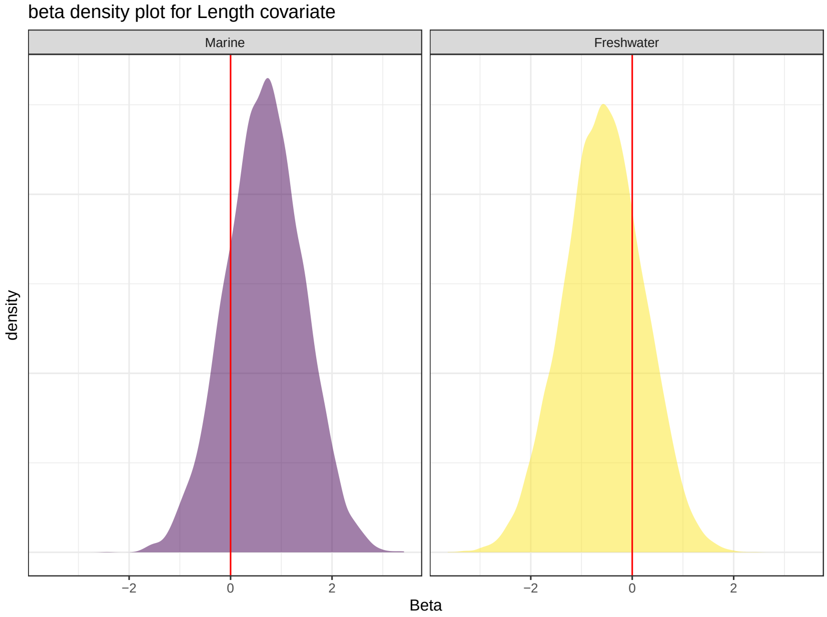


Figure 5.13: Density plot showing estimated β values for Length covariate for both marine and freshwater food sources.

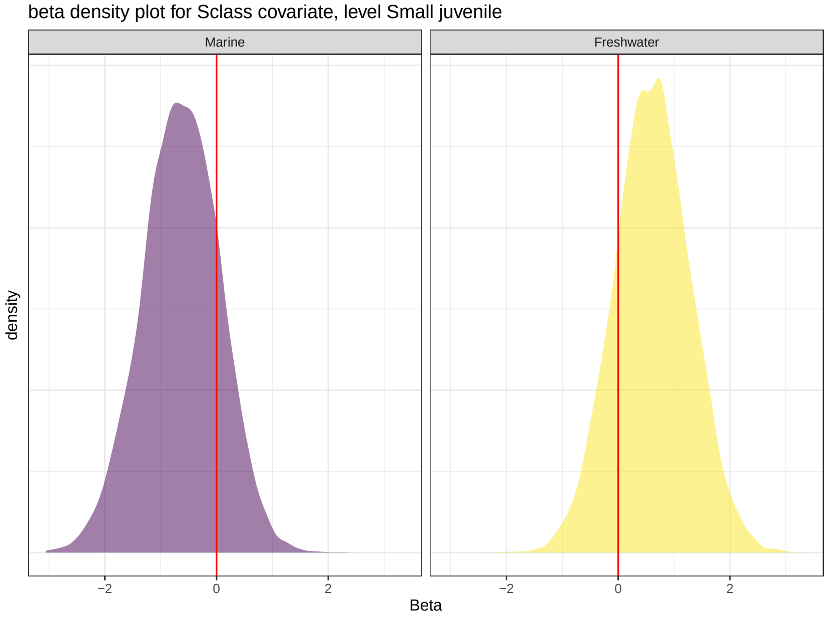


Figure 5.14: Density plot showing estimated β values for Sclass covariate for Small Juveniles for both marine and freshwater food sources.

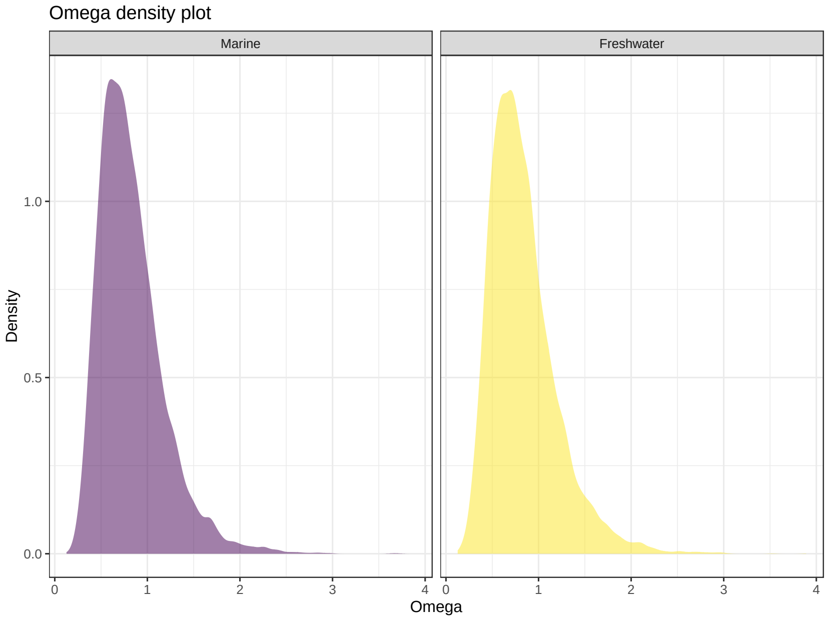


Figure 5.15: Density plot showing ω values for each food source for Alligator example.

is generated by using the `plot` function. Within this function, a data frame is generated which has length varying regularly from the minimum to maximum values. All other covariates are assumed to be fixed at their median value (if numeric) or in the first level of factors. This plot highlights what we could see in the plot of β values, that ‘Length’ influences the consumption of freshwater and marine sources. This aligns with the findings from the original Nifong et al. (2015) paper, which had the same conclusion via stomach content analysis. We see, for example, that marine consumption changes from less than 20% to over 80% as the length of an alligator increases. In Figure 5.17 we can see the change in freshwater consumption over the different Size classes. The code for generating these plots is shown below:

```
> plot(out_alli, type = "covariates_plot", cov_name = "Length")
> plot(out_alli, type = "covariates_plot", one_plot = FALSE, cov_name = "Sclass")
```

Users can use the predict function to predict for values of the covariate not present in the original data set. For example, code is shown below to predict proportion

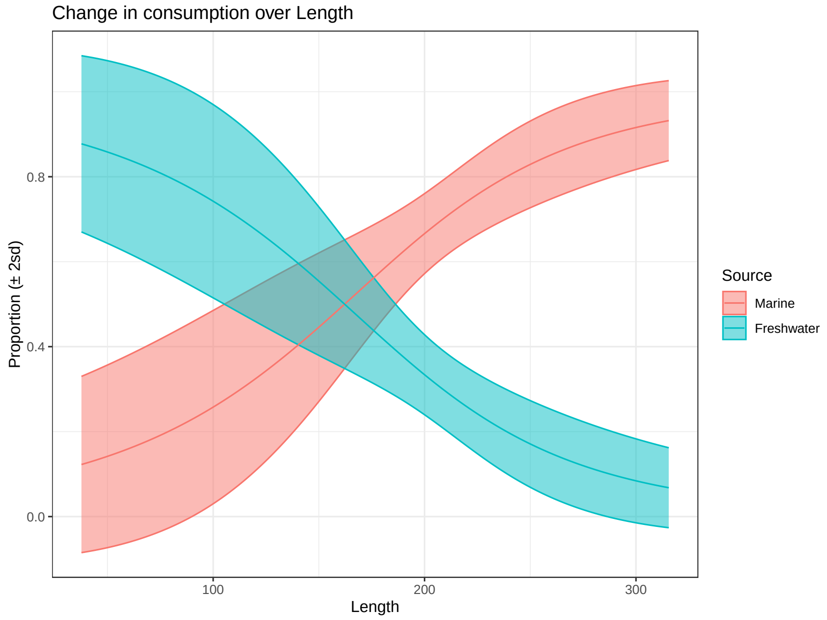


Figure 5.16: Lineplot showing change in consumption of freshwater and marine sources as Length increases.

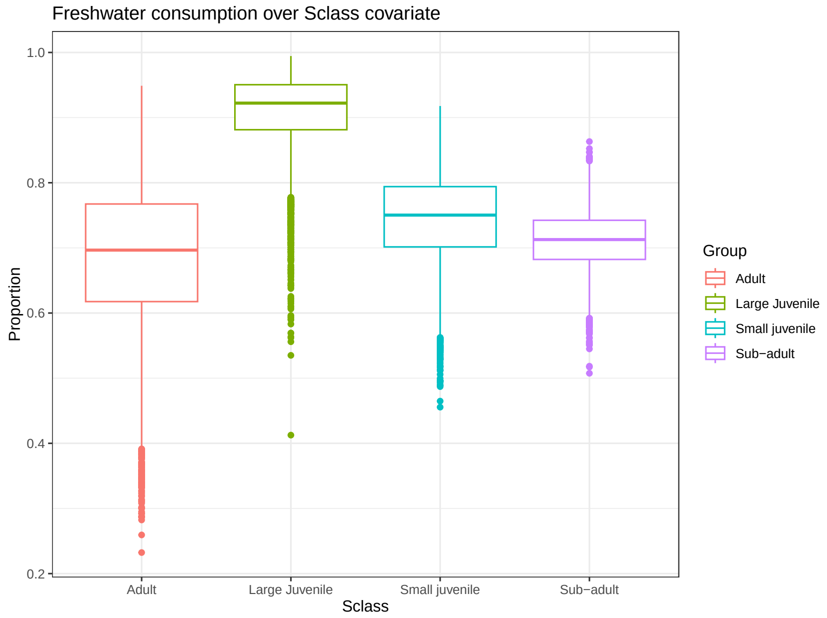


Figure 5.17: Boxplot showing difference in consumption of freshwater sources over different Size classes.

values of alligators of length 50cm and 200cm, in size classes ‘Adult’ and ‘Small juvenile’ respectively.

```
> x_df_alli = data.frame(Length = c(50, 200),
             Sclass = as.factor(c("Adult", "Small juvenile")))
> pred_alli_out = predict(out_alli, x_pred = x_df_alli)
```

The ‘summary’ and ‘plot’ functions can then be used to analyse the object created by the ‘predict’ function:

```
> summary(pred_alli_out, type = "statistics", obs = c(1,2))

Summary for Observation 1

              mean    sd
P(Marine)     0.142 0.107
P(Freshwater) 0.858 0.107

Summary for Observation 2

              mean    sd
P(Marine)     0.311 0.178
P(Freshwater) 0.689 0.178
```

This example highlights `cosimmrSTAN`s ability to handle multiple covariates in a much shorter timeframe than MixSIAR. It highlights the usefulness of the other functions in `cosimmrSTAN`, for example the `predict` function which allows us to examine combinations of covariates which aren’t present in the original data set.

5.5 Conclusion

`cosimmrSTAN` is a useful package for fast running of SIMMs. It allows for users to include both fixed and random covariates, as well as the inclusion of raw source data or fitted source data. It is designed for ease-of-use and has built in plot and summary functions to allow for easy visualisation of results. The built-in predict function allows for users to examine combinations of covariates not present in the original data set. The ability to plot ω values allows us to look at variability of source consumption between different groups.

Variational Bayes offers a speed advantage over previous algorithms popularly used in SIMMs, such as MCMC. In this paper we have illustrated a speed improvement of between 70 and 110 times faster than MixSIAR. This means that users will be able to run more complex models in a shorter time frame, as well as running potential different combinations of covariates and choosing the correct combination for their question.

Potential extensions for this package include allowing effects to be nested. This is available in MixSIAR and we plan to include this in future iterations of this package. Non-linear effects could also be a fruitful avenue of exploration. The allowance of different sources to be associated with different covariates, for example if we are looking at different regions, could be useful and is something we can add in future versions of `cosimmrSTAN`.

CHAPTER 6

Final Remarks

In this chapter, we review and summarise the work presented in this manuscript, reflect on obstacles or difficulties encountered, and provide proposals for future research and additions to the work described.

In this final chapter, we present an overview of each of the previous chapters, emphasising the novel techniques used in each, the limitations within each chapter and potential extensions that could be carried out at a later date.

Stable Isotope Mixing Models (SIMMs) are widely cited and used in ecology. They are important in the examination and study of an animal's diet. SIMMs allow for less invasive studying of an animal's diet, and allow for ecologists to have a measurable way to compare niches between species. They can also be used in other fields such as geology and pollution, and are important in seeing how different pollutants contribute to an overall polluted area.

However, the use of Markov chain Monte Carlo (MCMC) in SIMMs means that models can become prohibitively slow, sometimes taking millions of iterations in order to converge, depending on the complexity of the model. This formed the motivation for this thesis: using Variational Bayes in order to accelerate model fitting. The use of Variational Bayes allows for models to converge much more quickly. While VB is not guaranteed to converge, we have found throughout this thesis that VB converged to give consistent results with MCMC, while offering a significant (2-100 times) speed improvement over MCMC, depending on the complexity of the model.

Chapter 3 introduced the package `simmr`, which allows users to run a SIMM via MCMC or FFVB. `simmr` is designed for ease-of-use. It has built-in plot and summary functions, as well as posterior predictive and prior visualisation functions which allow for users to examine how well their data is fitting the model as well as looking at how their posterior changes from the prior. Users can choose between MCMC or FFVB when they are running a SIMM. `simmr` is an accessible package designed with ecologists in mind. The main limitation of `simmr` is the lack of ability to include covariates, and this is what we focused on in Chapter 4. The inability to include covariates means that users can only run simpler models through `simmr`, and they should be sure that covariates are not important in their system before running the model through `simmr`, as otherwise they violate assumptions about their data being independent.

Chapter 4 introduced the package `cosimmr`, which allows users to run SIMMs with fixed covariates included. `cosimmr` offered a speed improvement of up to one order of magnitude over MixSIAR, another package designed for running of SIMMs which uses MCMC. This speed advantage means that users can run more complex models in a shorter time frame, as well as allowing users to run multiple models and use model selection to pick the most accurate model for their data. `cosimmr` also allowed for users to include more complex error terms, to account for dietary specialisation. One of the main limitations of `cosimmr` is the fact that the package cannot handle random covariates, and it also does not allow for raw source data, or have any capability to fit source data. This means that unless the user is fitting their source data themselves before including it in `cosimmr`, they are making the assumption that their measured mean and standard deviation are the true mean and standard deviation, and not a sample mean and standard deviation.

Chapter 5 focuses on the package `cosimmrSTAN`. This package is designed for ease-of-use and it is modelled around the pre-existing `simmr` and `cosimmr` framework, however cosimmrSTAN utilises STAN's VB function and allows for users to fit fixed and random covariates, as well as allowing for raw source data or fitted source data, depending on the users wishes. This package offers between 70 and 100 times speed improvement on MixSIAR, depending on the complexity of the model. The use of Variational Bayes offers a speed improvement over other packages, and the use of STAN to implement this package allows for more complex models to be specified. The package is also designed to be accessible for non-expert users, and does not require a detailed knowledge of statistics in order to run complex models. It also has built-in plotting, summary, and predict functions which allow users to gain further insights. The main limitation of the cosimmrSTAN package is the fact that users cannot implement nested effects. This could be a useful feature in the future as it would allow for users to run more complex models.

These new packages will allow for users to perform more complex analyses of animal diets, as well as being useful in other fields, such as in the study of pollution. Recent studies utilising simmr include Capece et al. (2025), which investigates carbon storage in a salt marsh. While these packages were initially designed with animal diets in mind, they have the potential to be widely used outside of this field.

There are several avenues that could provide future expansions to this work, and these avenues can be broadly categorised into: software improvements, and ecological improvements. For software, developments could include interactive iso-space plots, which allow for users to click on any individual on the plot and get a summary of their source consumption. Development of a shiny app (Chang et al., 2024) could allow for interactive plots for users.

Richer covariate models are also a fruitful avenue for expansion. While we are able to run many of the examples that are available in MixSIAR (Stock et al., 2018), we are unable to run models with nested effects. Future development could include developing a package that allows for the inclusion of nested effects. These models could be created by utilising machine learning models such as Bayesian Additive Regression Trees (Chipman et al., 2010), or through the use of splines. Another possibility for future work is the development of one 'master' package in R that allows for users to run any of the models outlined in this thesis through one R package.

There are also several improvements we could make in order for our models to become more ecologically accurate. Firstly, a better understanding of isotopic routing, how isotopes transfer into tissues and isotope formation would be a useful avenue for future research. Having more biological knowledge about the system we are working in leads to more accurate statistical models. There is also the possibility of developing software to deal with these issues. Collaboration with ecologists to discover the main issues they are facing and how we can best help with that could be a fruitful avenue for research.

There is also the issue of combining sources. While we generally default to recommending sources be combined a posteriori unless there is a sound biological basis for combining a priori, combining a posteriori does mean that we are using a slightly different prior distribution. An important development might be changing the definition of the prior distribution if users might be combining sources after (Stock et al., 2018).

Finally, allowing for users to incorporate data from different tissues into the same model might prove useful. Different tissues have different turnover rates (Tieszen et al., 1983a) and therefore can tell you about different time spans of the animals diet. Future models could allow for users to include data from several different tissue types simultaneously, giving users a view of the animal's diet in both the shorter and longer term.

The field of Stable Isotope Mixing Models is an exciting one, with many possible future developments which will allow for users to gain a deeper understanding into the diets of animals, as well as their subsequent trophic interactions and their place in the ecosystem. The use of `simmr`, `cosimmr`, and `cosimmrSTAN` will allow for ecologists to run more complex models more quickly, leading to greater insight and understanding of our environment and the interactions between animals.

All methodology in this thesis is freely reproducible from code available at `https://github.com/emmagovan` in the repositories `simmr_paper_SIMM_package_scripts`, cosimmr_paper and cosimmrSTAN_paper, for chapters 3, 4 and 5.

Bibliography

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data.* Chapman and Hall. 14, 33, 57, 95

Aksu, S., Emiroğlu, Ö., Balzani, P., Britton, J. R., Köse, E., Kurtul, I., Başkurt, S., Mol, O., Çınar, E., Haubrock, P. J., et al. (2023). High trophic similarity between non-native common carp and gibel carp in Turkish freshwaters: Implications for management. *Aquaculture and Fisheries.* 50

Albouy, C., Velez, L., Coll, M., Colloca, F., Le Loc'h, F., Mouillot, D., and Gravel, D. (2014). From projected species distribution to food-web structure under climate change. *Global change biology*, 20(3):730–741. 1

Arnoldi, J.-F., Bortoluzzi, J. R., Rowlands, H., Harrod, C., Parnell, A., Payne, N., Donohue, I., and Jackson, A. L. (2023). Identifying the limits where variation in consumer stable isotope values reflect variation in diet. *bioRxiv.* 9, 44

Balena, F. and Fawcette, J. (1999). *Programming Microsoft Visual Basic 6.0*, volume 1. Microsoft press Washington. 16, 31

Bearhop, S., Adams, C. E., Waldron, S., Fuller, R. A., and Macleod, H. (2004). Determining trophic niche width: a novel approach using stable isotope analysis. *Journal of animal ecology*, 73(5):1007–1012. 9

Belis, C., Favez, O., Mircea, M., Diapouli, E., Manousakas, M., Vratolis, S., Gilardoni, S., Paglione, M., Decesari, S., Mocnik, G., Mooibroek, D., Salvador, P., Takahama, S., Vecchi, R., and Paatero, P. (2019). European guide on air pollution source apportionment with receptor models - revised version 2019. *Publications Office of the European Union, Luxembourg.* 24

Ben-David, M. and Schell, D. (2001). Mixing models in analyses of diet using multiple stable isotopes: a response. *Oecologia*, 127(2):180–184. 13

Betancourt, M. and Girolami, M. (2015). Hamiltonian monte carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79(30):2–4. 100

Billheimer, D. (2001). Compositional receptor modeling. *Environmetrics: The official journal of the International Environmetrics Society*, 12(5):451–467. 24

Brewer, M., Tetzlaff, D., Malcolm, I., and Soulsby, C. (2011). Source distribution modelling for end-member mixing in hydrology. *Environmetrics*, 22(8):921–932. 24

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of markov chain monte carlo*. CRC press. 100

Campodonico, V. A., Pasquini, A. I., Lecomte, K. L., García, M. G., and Depetris, P. J. (2019). Chemical weathering in subtropical basalt-derived laterites: A mass balance interpretation (misiones, ne argentina). *Catena*, 173:352–366. 24

Capece, L. R., Bailey, M., Morrison, M., Phillips, A. A., Sharpnack, L., Webb, S. M., Brenner, D. C., Gomes, M., and Raven, M. R. (2025). Evaluating sulfurization as a blue carbon sink in a southern california salt marsh. *Limnology and Oceanography*. 125

Caro, A., Got, P., Bouvy, M., Troussellier, M., and Gros, O. (2009). Effects of long-term starvation on a host bivalve (codakia orbicularis, lucinidae) and its symbiont population. *Applied and Environmental Microbiology*, 75(10):3304–3313. 9

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1). 4, 93

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., and Borges, B. (2024).

shiny: Web Application Framework for R. R package version 1.9.1.9000, https://github.com/rstudio/shiny. 125

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266 – 298. 125

Christensen, W. F. (2004). Chemical mass balance analysis of air quality data when unknown pollution sources are present. *Atmospheric Environment*, 38(26):4305–4317. 24

Clark, T. S. and Linzer, D. A. (2015). Should i use fixed or random effects? *Political science research and methods*, 3(2):399–408. 97

Cooper, R. J., Krueger, T., Hiscock, K. M., and Rawlins, B. G. (2014). Sensitivity of fluvial sediment source apportionment to mixing model assumptions: A bayesian model comparison. *Water Resources Research*, 50(11):9031–9047. 24

Csárdi, G. (2019). *cranlogs: Download Logs from the 'RStudio' 'CRAN' Mirror*. R package version 2.1.1. xiv, 26, 68

DB, R. (1988). Using the sir algorithm to simulate posterior distributions. In *Bayesian statistics 3. Proceedings of the third Valencia international meeting, 1-5 June 1987*, pages 395–402. Clarendon Press. 16

DeNiro, M. J. and Epstein, S. (1978). Influence of diet on the distribution of carbon isotopes in animals. *Geochimica et cosmochimica acta*, 42(5):495–506. 2, 6, 8, 12, 21

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18. 52, 101

Farquhar, G. D., Ehleringer, J. R., Hubick, K. T., et al. (1989). Carbon isotope discrimination and photosynthesis. *Annual review of plant physiology and plant molecular biology*, 40(1):503–537. 8

Fernandes, R., Millard, A. R., Brabec, M., Nadeau, M.-J., and Grootes, P. (2014). Food reconstruction using isotopic transferred signals (FRUITS): A Bayesian model for diet reconstruction. *PloS one*, 9(2):e87436. 16, 31, 32, 53

Franco-Trecu, V., Drago, M., Riet-Sapriza, F. G., Parnell, A., Frau, R., and Inchausti, P. (2013). Bias in diet determination: Incorporating traditional methods in bayesian mixing models. *PLoS One*, 8(11):e80019. 42

Fry, B. (2006). *Stable isotope ecology*, volume 521. Springer. 9

Galloway, A., Eisenlord, M., Dethier, M., Holtgrieve, G., and Brett, M. (2014). Quantitative estimates of isopod resource utilization using a Bayesian fatty acid mixing model. *Marine Ecology Progress Series*, 507:219–232. vii, 68, 73

Galloway, A. W., Britton-Simmons, K. H., Duggins, D. O., Gabrielson, P. W., and Brett, M. T. (2012). Fatty acid signatures differentiate marine macrophytes at ordinal and family ranks. *Journal of Phycology*, 48(4):956–965. 74

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian Analysis*, 1(3):515 – 534. 98

Gelman, A. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press. 97

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472. 38, 101

Govan, E., Jackson, A. L., Bearhop, S., Inger, R., Stock, B. C., Semmens, B. X., Ward, E. J., and Parnell, A. C. (2024). cosimmr: an r package for fast fitting of stable isotope mixing models with covariates. *arXiv preprint arXiv:2408.17230*. 99, 101, 114

Govan, E., Jackson, A. L., Inger, R., Bearhop, S., and Parnell, A. C. (2023). simmr: A package for fitting stable isotope mixing models in R. *arXiv preprint arXiv:2306.07817*. 51, 53, 99

Greer, A. L., Horton, T. W., and Nelson, X. J. (2015). Simple ways to calculate stable isotope discrimination factors and convert between tissue types. *Methods in Ecology and Evolution*, 6(11):1341–1348. 53

Hardin, G. (1960). The competitive exclusion principle: an idea that took a century to be born has implications in ecology, economics, and genetics. *science*, 131(3409):1292–1297. 6

Harrison, X. A., Donaldson, L., Correa-Cano, M. E., Evans, J., Fisher, D. N., Goodwin, C. E., Robinson, B. S., Hodgson, D. J., and Inger, R. (2018). A brief introduction to mixed effects modelling and multi-model inference in ecology. *PeerJ*, 6:e4794. 97

Healy, K., Guillerme, T., Kelly, S. B., Inger, R., Bearhop, S., and Jackson, A. L. (2018). SIDER: an R package for predicting trophic discrimination factors of consumers based on their ecology and phylogenetic relatedness. *Ecography*, 41(8):1393–1400. 13, 23, 54, 92

Henry, R. C. (1997). History and fundamentals of multivariate air quality receptor models. *Chemometrics and intelligent laboratory systems*, 37(1):37–42. 24

Hertz, E., Trudel, M., Tucker, S., Beacham, T. D., and Mazumder, A. (2017). Overwinter shifts in the feeding ecology of juvenile chinook salmon. *ICES Journal of Marine Science*, 74(1):226–233. 42

Higgs, N. D., Newton, J., and Attrill, M. J. (2016). Caribbean spiny lobster fishery is underpinned by trophic subsidies from chemosynthetic primary production. *Current Biology*, 26(24):3393–3398. 9

Hobson, K. A. (1999). Tracing origins and migration of wildlife using stable isotopes: A review. *Oecologia*, 120(3):314–326. 22

Hobson, K. A. and Clark, R. (1993). Turnover of 13 c in cellular and plasma fractions of blood: implications for nondestructive sampling in avian dietary studies. *The Auk*, 110(3):638–641. 17

Hooper, R. P., Christophersen, N., and Peters, N. E. (1990). Modelling streamwater chemistry as a mixture of soilwater end-members—an application to the Panola mountain catchment, Georgia, USA. *Journal of Hydrology*, 116(1-4):321–343. 7, 51, 91

Hopke, P. K. (1991). *Receptor Modeling for Air Quality Management.* Elsevier. 7, 24, 51, 92

Hopkins III, J. B. and Ferguson, J. M. (2012). Estimating the diets of animals using stable isotopes and a comprehensive Bayesian mixing model. *PLoS one*, 7(1):e28478. 53, 55, 56

Hurlbert, S. H. (1984). Pseudoreplication and the design of ecological field experiments. *Ecological monographs*, 54(2):187–211. 54

Hutchinson, G. (1957). E. 1957. concluding remarks. In *Cold Spring Harb. Symp. Quant. Biol*, volume 22, pages 415–427. 2, 9

Inc., T. M. (2022). Matlab version: 9.13.0 (r2022b). 16, 31

Inger, R. and Bearhop, S. (2008). Applications of stable isotope analyses to avian ecology. *Ibis*, 150(3):447–461. 12, 22, 33, 50, 51, 53, 92

Inger, R., Ruxton, G. D., Newton, J., Colhoun, K., Robinson, J. A., Jackson, A. L., and Bearhop, S. (2006). Temporal and intrapopulation variation in prey choice of wintering geese determined by stable isotope analysis. *Journal of Animal Ecology*, 75(5):1190–1200. vii, 4, 18, 19, 23, 35, 68, 69, 103

Jackson, M. C. and Britton, J. R. (2014). Divergence in the trophic niche of sympatric freshwater invaders. *Biological invasions*, 16:1095–1103. 7

Jackson, M. C., Donohue, I., Jackson, A. L., Britton, J. R., Harper, D. M., and Grey, J. (2012). Population-level metrics of trophic structure based on stable isotopes and their application to invasion ecology. *PloS one*, 7(2):e31757. 7

Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. (2015). Automatic variational inference in stan. *Advances in neural information processing systems*, 28. 2, 101

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86. 3, 15, 35

Lee, H., Calvin, K., Dasgupta, D., Krinner, G., Mukherji, A., Thorne, P., Trisos, C., Romero, J., Aldunce, P., Barret, K., et al. (2023). Ipcc, 2023: Climate change 2023: Synthesis report, summary for policymakers. contribution of working groups i, ii and iii to the sixth assessment report of the intergovernmental panel on climate change [core writing team, h. lee and j. romero (eds.)]. ipcc, geneva, switzerland. 1

Lescord, G. L., Johnston, T. A., Ponton, D. E., Amyot, M., Lock, A., and Gunn, J. M. (2022). The speciation of arsenic in the muscle tissue of inland and coastal freshwater fish from a remote boreal region. *Chemosphere*, 308:136140. 91

Lewandowski, D., Kurowicka, D., and Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9):1989–2001. 99

Lingwall, J. W., Christensen, W. F., and Reese, C. S. (2008). Dirichlet based bayesian multivariate receptor modeling. *Environmetrics: The official journal of the International Environmetrics Society*, 19(6):618–629. 24

Lipscombe, R. S., Meyer, L., Butcherine, P., Morris, S., Huveneers, C., Scott, A., and Butcher, P. A. (2024). A taste of youth: Seasonal changes in the diet of immature white sharks in eastern australia. *Frontiers in Marine Science*, 11:1359785. 2, 91

Liu, G., Ma, F., Liu, G., Guo, J., Duan, X., and Gu, H. (2020). Quantification of water sources in a coastal gold mine through an end-member mixing analysis combining multivariate statistical methods. *Water*, 12(2):580. 24

Lurgi, M., Lopez, B. C., and Montoya, J. M. (2012). Climate change impacts on body size and food web structure on mountain ecosystems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1605):3050–3057. 1

Manlick, P. J. and Newsome, S. D. (2022). Stable isotope fingerprinting traces essential amino acid assimilation and multichannel feeding in a vertebrate consumer. *Methods in Ecology and Evolution*, 13(8):1819–1830. 50

Mathers, R. and Montgomery, W. (1997). Quality of food consumed by over wintering pale-bellied brent geese Branta bernicla hrota and wigeon Anas penelope. In *Biology and Environment: Proceedings of the Royal Irish Academy*, pages 81–89. JSTOR. 70

McDonald, R. A., Wilson-Aggarwal, J. K., Swan, G. J., Goodwin, C. E., Moundai, T., Sankara, D., Biswas, G., and Zingeser, J. A. (2020). Ecology of domestic dogs Canis familiaris as an emerging reservoir of guinea worm Dracunculus medinensis infection. *PLoS neglected tropical diseases*, 14(4):e0008170. 50

McKechnie, A. E. (2004). Stable isotopes: Powerful new tools for animal ecologists: News & views. *South African Journal of Science*, 100(3):131–134. 22

Miller, M., Friedlander, S., and Hidy, G. (1972). A chemical element balance for the pasadena aerosol. *Journal of Colloid and Interface Science*, 39(1):165–176. 7, 51, 92

Minagawa, M. and Wada, E. (1984). Stepwise enrichment of 15n along food chains: further evidence and the relation between δ15n and animal age. *Geochimica et cosmochimica acta*, 48(5):1135–1140. 2, 8, 12

Mircea, M., Calori, G., Pirovano, G., and Belis, C. (2020). European guide on air pollution source apportionment for particulate matter with source oriented models and their combined use with receptor models. *Publications Office of the European Union, Luxembourg.* 24

Moore, J. W. and Semmens, B. X. (2008). Incorporating uncertainty and prior information into stable isotope mixing models. *Ecology letters*, 11(5):470–480. 12, 13, 16, 31, 42, 51, 52, 57, 94

Munoz, S. E., Giosan, L., Blusztajn, J., Rankin, C., and Stinchcomb, G. E. (2019). Radiogenic fingerprinting reveals anthropogenic and buffering controls on sediment dynamics of the Mississippi river system. *Geology*, 47(3):271–274. 50

Mychajliw, A. M., Rick, T. C., Dagtas, N. D., Erlandson, J. M., Culleton, B. J., Kennett, D. J., Buckley, M., and Hofman, C. A. (2020). Biogeographic problem-solving reveals the late pleistocene translocation of a short-faced bear to the california channel islands. *Scientific reports*, 10(1):1–13. 23

Newsome, S. D., Martinez del Rio, C., Bearhop, S., and Phillips, D. L. (2007). A niche for isotopic ecology. *Frontiers in Ecology and the Environment*, 5(8):429–436. 2, 9

Nifong, J. C., Layman, C. A., and Silliman, B. R. (2015). Size, sex and individual-level behaviour drive intrapopulation variation in cross-ecosystem foraging of a top-predator. *Journal of Animal Ecology*, 84(1):35–48. vii, 68, 77, 79, 114, 119

Paez-Rosas, D., Suarez-Moncada, J., Arnes-Urgelles, C., Espinoza, E., Robles, Y., and Salinas-De-Leon, P. (2024). Assessment of nursery areas for the scalloped hammerhead shark (sphyrna lewini) across the eastern tropical Pacific using a stable isotopes approach. *Frontiers in Marine Science*, 10. 50

Palmer, M. J. and Douglas, G. B. (2008). A bayesian statistical model for end member analysis of sediment geochemistry, incorporating spatial dependences. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 57(3):313–327. 24

Park, E. S., Guttorp, P., and Henry, R. C. (2001). Multivariate receptor modeling for temporally correlated data by using mcmc. *Journal of the American Statistical Association*, 96(456):1171–1183. 24

Parnell, A. C., Inger, R., Bearhop, S., and Jackson, A. L. (2010). Source partitioning using stable isotopes: Coping with too much variation. *PloS one*, 5(3):e9672. 16, 31, 32, 52, 93

Parnell, A. C., Phillips, D. L., Bearhop, S., Semmens, B. X., Ward, E. J., Moore, J. W., Jackson, A. L., Grey, J., Kelly, D. J., and Inger, R. (2013). Bayesian stable isotope mixing models. *Environmetrics*, 24(6):387–399. 56

Pati, D., Bhattacharya, A., and Yang, Y. (2018). On statistical optimality of variational bayes. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of* *the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1579–1588. PMLR. 35

Pérez-Ramallo, P., Rissech, C., Lloveras, L., Lucas, M., Urbina, D., Urquijo, C., and Roberts, P. (2024). Unravelling social status in the first medieval military order of the iberian peninsula using isotope analysis. *Scientific Reports*, 14(1):11074. 2

Peterson, B. J. and Fry, B. (1987). Stable isotopes in ecosystem studies. *Annual review of ecology and systematics*, pages 293–320. 2, 8, 22

Phillips, D. L. (2001). Mixing models in analyses of diet using multiple stable isotopes: a critique. *Oecologia*, 127(2):166–170. 6

Phillips, D. L. (2012). Converting isotope values to diet composition: the use of mixing models. *Journal of Mammalogy*, 93(2):342–352. 50

Phillips, D. L. and Gregg, J. W. (2001). Uncertainty in source partitioning using stable isotopes. *Oecologia*, pages 171–179. 31

Phillips, D. L. and Gregg, J. W. (2003). Source partitioning using stable isotopes: Coping with too many sources. *Oecologia*, 136:261–269. 16, 30, 31, 52, 53

Phillips, D. L., Inger, R., Bearhop, S., Jackson, A. L., Moore, J. W., Parnell, A. C., Semmens, B. X., and Ward, E. J. (2014). Best practices for use of stable isotope mixing models in food-web studies. *Canadian Journal of Zoology*, 92(10):823–835. 17, 22, 45, 104

Phillips, D. L. and Koch, P. L. (2002). Incorporating concentration dependence in stable isotope mixing models. *Oecologia*, 130:114–125. 13, 31, 33, 51, 54, 92

Phillips, D. L., Newsome, S. D., and Gregg, J. W. (2005). Combining sources in stable isotope mixing models: alternative methods. *Oecologia*, 144:520–527. 18

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs Sampling. In *Proceedings of the 3rd international workshop on* *distributed statistical computing*, volume 124, pages 1–10. Vienna, Austria. 15, 16, 25, 52, 53

Polis, G. A. (1991). Complex trophic interactions in deserts: an empirical critique of food-web theory. *The American Naturalist*, 138(1):123–155. 1

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 5, 23, 50, 93

Radtke, R. L., Showers, W., Moksness, E., and Lenz, P. (1996). Environmental information stored in otoliths" insights from stable isotopes. *Marine Biology*, 127:161–170. 17, 22

Salimans, T. and Knowles, D. A. (2013). Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. *Bayesian Analysis*, 8(4):837 – 882. 2, 15, 58

Schmidt, S. N., Olden, J. D., Solomon, C. T., and Zanden, M. J. V. (2007). Quantitative approaches to the analysis of stable isotope food web data. *Ecology*, 88(11):2793–2802. 7

Semmens, B. X., Ward, E. J., Moore, J. W., and Darimont, C. T. (2009). Quantifying inter-and intra-population niche variability using hierarchical Bayesian stable isotope mixing models. *PloS one*, 4(7):e6187. 51, 108

Soulsby, C., Petry, J., Brewer, M., Dunn, S., Ott, B., and Malcolm, I. (2003). Identifying and assessing uncertainty in hydrological pathways: A novel approach to end member mixing in a scottish agricultural catchment. *Journal of Hydrology*, 274(1-4):109–128. 24

Stan Development Team (2024). RStan: the R interface to Stan. R package version 2.32.6. 93

Stock, B. C., Jackson, A. L., Ward, E. J., Parnell, A. C., Phillips, D. L., and Semmens, B. X. (2018). Analyzing mixing systems using a new generation of Bayesian tracer mixing models. *PeerJ*, 6:e5096. 16, 31, 32, 53, 55, 57, 93, 94, 95, 99, 100, 125, 126

Styring, A. K., Knipper, C., Müller-Scheeßel, N., Grupe, G., and Bogaard, A. (2022). The proof is in the pudding: Crop isotope analysis provides direct insights into agricultural production and consumption. *Environmental Archaeology*, 27(1):61–72. 23

Swan, G. J., Bearhop, S., Redpath, S. M., Silk, M. J., Goodwin, C. E., Inger, R., and McDonald, R. A. (2020). Evaluating bayesian stable isotope mixing models of wild animal diet and the effects of trophic discrimination factors and informative priors. *Methods in Ecology and Evolution*, 11(1):139–149. 50

Tan, L. S. and Nott, D. J. (2018). Gaussian variational approximation with sparse precision matrices. *Statistics and Computing*, 28:259–275. 2, 51, 58, 101

Tao, X., Paoletti, M. E., Haut, J. M., Ren, P., Plaza, J., and Plaza, A. (2021). Endmember estimation with maximum distance analysis. *Remote Sensing*, 13(4):713. 24

Teixeira, C. R., Botta, S., Daura-Jorge, F. G., Pereira, L. B., Newsome, S. D., and Simões-Lopes, P. C. (2021). Niche overlap and diet composition of three sympatric coastal dolphin species in the southwest Atlantic ocean. *Marine Mammal Science*, 37(1):111–126. 50

Thibault, M., Letourneur, Y., Cleguer, C., Bonneville, C., Briand, M. J., Derville, S., Bustamante, P., and Garrigue, C. (2024). C and N stable isotopes enlighten the trophic behaviour of the dugong (dugong dugon). *Scientific Reports*, 14(1):896. 2, 50

Thompson, M. S., Couce, E., Schratzberger, M., and Lynam, C. P. (2023). Climate change affects the distribution of diversity across marine food webs. *Global Change Biology*, 29(23):6606–6619. 1

Tieszen, L. L., Boutton, T. W., Tesdahl, K. G., and Slade, N. A. (1983a). Fractionation and turnover of stable carbon isotopes in animal tissues: implications for δ 13 c analysis of diet. *Oecologia*, 57:32–37. 17, 126

Tieszen, L. L., Boutton, T. W., Tesdahl, K. G., and Slade, N. A. (1983b). Fractionation and turnover of stable carbon isotopes in animal tissues: Implications for δ 13 c analysis of diet. *Oceologia*, 57:32–37. 22

Titsias, M. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. In *International conference on machine learning*, pages 1971–1979. PMLR. 2, 51, 58, 101

Tran, M.-N., Nguyen, T.-N., and Dao, V.-H. (2021). A practical tutorial on variational bayes. *arXiv preprint arXiv:2103.01327*. 46, 53, 59, 83

Ugalde, P. C., Gayo, E. M., Labarca, R., Santoro, C. M., and Quade, J. (2024). Camelids in the hyperarid core of the atacama desert 12,000–11,000 years ago? a stable isotope study and its consequences for early human settlement. *Quaternary Science Reviews*, 335:108750. 91

Vander Zanden, H. B., Soto, D. X., Bowen, G. J., and Hobson, K. A. (2016). Expanding the isotopic toolbox: applications of hydrogen and oxygen stable isotope ratios to food web studies. *Frontiers in Ecology and Evolution*, 4:20. 9

Vander Zanden, M. J., Casselman, J. M., and Rasmussen, J. B. (1999). Stable isotope evidence for the food web consequences of species invasions in lakes. *Nature*, 401(6752):464–467. 6

Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., Bürkner, P.-C., Paananen, T., and Gelman, A. (2024). loo: Efficient leave-one-out cross-validation and waic for Bayesian models. R package version 2.7.0. 77

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, 27:1413–1432. 102

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2015). Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*. 102

Ward, E. J., Semmens, B. X., and Schindler, D. E. (2010). Including source uncertainty and prior information in the analysis of stable isotope mixing models. *Environmental science & technology*, 44(12):4645–4650. 51, 55, 99, 100

Webb, C. O., Ackerly, D. D., McPeek, M. A., and Donoghue, M. J. (2002). Phylogenies and community ecology. *Annual review of ecology and systematics*, 33(1):475–505. 6

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. 25

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686. 25, 54

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018). Yes, but did it work?: Evaluating variational inference. In *International Conference on Machine Learning*, pages 5581–5590. PMLR. 52, 101

Zaryab, A., Nassery, H. R., Knoeller, K., Alijani, F., and Minet, E. (2022). Determining nitrate pollution sources in the Kabul plain aquifer (Afghanistan) using stable isotopes and Bayesian stable isotope mixing model. *Science of the Total Environment*, 823:153749. 50