



# Technical note: MixSIAPy – Bayesian stable isotope mixing models in Python with multi-backend inference and formula-based compositional regression

5 Jing Yang<sup>1</sup>, Heng Dai<sup>2,3</sup>, Ming Ye<sup>4</sup>, Su Cheng<sup>5</sup>, Yonghua Zhao<sup>1</sup>, Hui Qian<sup>6</sup>

<sup>1</sup>School of Land Engineering, Chang'an University, Xi'an 710054, China

<sup>2</sup>State Key Laboratory of Geomicrobiology and Environmental Changes, China University of Geosciences, Wuhan 430074, Hubei, China

10 <sup>3</sup>Hubei Key Laboratory of Yangtze Catchment Environmental Aquatic Science, China University of Geosciences, Wuhan 430074, Hubei, China

<sup>4</sup>Department of Earth, Ocean, and Atmospheric Science, Florida State University, Tallahassee 32306, USA

<sup>5</sup>Shaanxi Bi Yuan Chenchuang Technology Co., Ltd., Xi'an 710065, China

<sup>6</sup>School of Water and Environment, Chang'an University, Xi'an 710054, Shaanxi, China

15 Correspondence to Yonghua Zhao ([yonghuaz@chd.edu.cn](mailto:yonghuaz@chd.edu.cn)) and Hui Qian ([qianhui@chd.edu.cn](mailto:qianhui@chd.edu.cn))

**Abstract.** Bayesian stable isotope mixing models quantify source contributions, but environmental studies increasingly require testing how multiple interacting drivers alter entire source compositions. The widely used R package MixSIAR restricts covariate structures, while post *hoc* regression of estimated proportions neither propagates posterior uncertainty nor preserves compositional dependence. Its reliance on a separately installed JAGS engine also constrains direct evaluation of gradient-based samplers and accelerator-aware computation. We here introduce MixSIAPy, an open-source Python framework addressing these statistical and computational limitations while retaining established MixSIAR model structures. Its principal methodological contribution is formula-based compositional regression embedded directly in the mixing-model likelihood. Multiple categorical and numerical predictors and their interactions act jointly on source compositions in isometric log-ratio coordinates, with regression, grouping, source, and error parameters estimated in one posterior. A single PyMC model can be fitted through four interchangeable No-U-Turn Sampler backends, enabling CPU sampling and optional GPU execution. A shared engine supports scripted, local graphical, and hosted web workflows with reproducible outputs. Across 20 paired benchmarks, 19 converged in both implementations and showed close posterior correspondence; MixSIAPy reached an acceptable posterior faster in 13 cases. Backend performance was model dependent, and GPUs became advantageous only for some models at high chain parallelism. In an additional crop-water-uptake case study, the model identified a biomass-associated shift from shallow to middle soil water in linseed, whereas broader effects remained uncertain. MixSIAPy thus enables interacting environmental controls to be tested within source-partitioning models instead of through separate analyses of uncertain estimated proportions.



## 1 Introduction

Stable isotope mixing models are widely used in hydrology to quantify the relative contributions of water and solute sources to observed mixtures (Popp et al., 2025). Applications include hydrograph separation (He et al., 2020), urban water-supply tracing (Jameel et al., 2018), plant water uptake (Allen et al., 2019; Couvreur et al., 2020; Landgraf et al., 2022), and pollution-source apportionment in surface water and groundwater (Yu et al., 2020; Gibrilla et al., 2020; Ji et al., 2022; He et al., 2022; Su et al., 2025; Zhang et al., 2025). Beyond hydrology, these models support analyses of animal diets (Hopkins and Ferguson, 2012; Swan et al., 2020), aquatic food webs (Silberberger et al., 2021; Lobry et al., 2022; Heikkinen et al., 2022), and sediment sources (Cooper and Krueger, 2017; Upadhayay et al., 2017). Inference is nevertheless challenging because source signatures may overlap, vary among samples, and change during transfer to the mixture (Phillips et al., 2014; Swan et al., 2020). Hydrological applications face additional uncertainty from spatiotemporal source variability and, in plant-water studies, from water-sampling and isotope-measurement procedures (Rothfuss and Javaux, 2017; Beyer et al., 2020). Bayesian mixing models address these uncertainties by combining tracer mass balance with probability models for source signatures, discrimination factors, mixture variability, and prior information (Moore and Semmens, 2008; Parnell et al., 2010; Erhardt and Bedrick, 2013; Parnell et al., 2013; Phillips et al., 2014). They therefore estimate source proportions while quantifying the uncertainty and dependence among them.

Several software tools implement isotope-based source apportionment. Established examples include IsoSource (Phillips and Gregg, 2003), MixSIR (Moore and Semmens, 2008), and SIAR (Parnell et al., 2010). More recent tools include FRUITS (Fernandes et al., 2014), simmr (Govan and Parnell, 2019), EMMTE (Cao et al., 2023), and the covariate-enabled cosimmr extension (Govan et al., 2024). These tools address different scientific and computational requirements. Among them, MixSIAR provides a particularly informative reference for software redevelopment because it integrates a broad range of environmentally relevant model structures (Stock et al., 2018). These structures include alternative source-data representations, concentration dependence, informative priors, covariate and hierarchical effects, and multiple error models. Its distributed examples also provide transparent benchmark cases spanning diverse datasets and model configurations. This breadth is important because source estimates can vary when implementations represent source uncertainty, covariance, priors, or residual variation differently (Wang et al., 2019; Swan et al., 2020; Qiu et al., 2025).

The current MixSIAR implementation uses an R script to generate a statistical model written in the Bayesian inference Using Gibbs Sampling (BUGS) language. The model is then fitted using Just Another Gibbs Sampler (JAGS), an external engine that must be installed separately (Plummer, 2003). This established architecture remains suitable for conventional analyses. However, reliance on a separately installed JAGS engine introduces an additional software dependency and makes portable deployment and evaluation of alternative inference engines less straightforward. In addition, the conditional node- or block-level updates used by JAGS may be inefficient for strongly correlated or high-dimensional posterior distributions. Gradient-



based methods use posterior geometry to update continuous parameters jointly and can improve sampling efficiency in such models (Hoffman and Gelman, 2014; Betancourt, 2017). Modern probabilistic-programming frameworks combine these methods with automatic differentiation and reusable computational graphs. Among them, PyMC (<https://www.pymc.io/welcome.html>, accessed on 09/20/2026) provides a mature and widely used Python framework that integrates with the broader scientific-computing and application-development ecosystem (Salvatier et al., 2016; Abril-Pla et al., 2023). Reimplementing MixSIAR in PyMC therefore provides a foundation for integrated workflows, interchangeable inference backends, and optional hardware-accelerated computation.

Computational modernization alone, however, does not address the limited flexibility of conventional covariate specification. MixSIAR represents covariate effects in isometric log-ratio (ILR) coordinates, thereby preserving the positivity and unit-sum constraints of source proportions (Aitchison, 1982; Egozcue et al., 2003; Parnell et al., 2013; Greenacre, 2021). Its standard interface does not provide a general formula system for jointly specifying multiple categorical and numerical predictors and their interactions. Post *hoc* regression of estimated proportions is not equivalent because it treats uncertain posterior quantities as observed responses and fails to propagate their joint compositional uncertainty. Incorporating a formula-generated design matrix into the Bayesian mixing model instead allows predictor effects, interactions, source proportions, and other model parameters to be estimated jointly.

Here, we introduce MixSIAPy, an open-source Python implementation and extension of MixSIAR. MixSIAPy reimplements the established MixSIAR model configurations as a PyMC computational graph. A common model specification can be fitted using four interchangeable Bayesian inference backends: native PyMC, Nutpie, NumPyro, and BlackJAX. These backends support CPU execution and, where compatible, optional GPU execution (Abril-Pla et al., 2023; Phan et al., 2019; Cabezas et al., 2024). MixSIAPy also introduces formula-based compositional regression for multiple predictors and their interactions. Its shared analysis engine is accessible through Python scripts, a locally installed graphical interface, and a hosted web application. We assessed numerical correspondence with MixSIAR using 20 matched benchmark analyses and compared computational performance across implementations, inference backends, and devices. We additionally demonstrated formula-based regression using a crop-water-uptake experiment (Schmutz and Schöb, 2023). Together, these evaluations assessed numerical fidelity, computational trade-offs, and the capacity to examine interacting environmental controls directly within a Bayesian mixing model.

## 2 Methodological and computational development

### 2.1 Foundation of Bayesian mixing models

For  $N$  mixture observations,  $K$  candidate sources, and  $J$  isotope tracers, let  $\mathbf{y}_i$  denote the  $J$ -dimensional observed tracer vector for mixture observation  $i$ . Conditional on the model parameters, mixture observations are independent and follow



$$\mathbf{y}_i \sim \text{Normal}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad (1)$$

where  $\boldsymbol{\mu}_i$  is the expected tracer vector and  $\boldsymbol{\Sigma}_i$  is the covariance matrix determined by source and discrimination variability and the selected error model.

The expected vector  $\boldsymbol{\mu}_i$  is calculated from the source proportions, source signatures, discrimination corrections, and, when specified, source-specific tracer concentrations. For each tracer, the source and discrimination means are averaged using weights proportional to the product of the source proportion and tracer concentration. When concentration dependence is not considered, concentrations are treated as equal across sources, and the expected value reduces to the ordinary source-proportion-weighted mean.

The covariance matrix  $\boldsymbol{\Sigma}_i$  is determined by source and discrimination variability and the selected error model. In models containing process error, the variance contributed by each source is weighted by the square of its source proportion. Raw source observations can retain covariance among tracers, whereas summary source data define diagonal source covariance matrices. In the process  $\times$  residual model, residual parameters scale the propagated process variability. The residual-only model estimates residual variability separately, whereas the process-only model uses the variability propagated from the source and discrimination distributions (Stock et al., 2018).

The baseline source composition is assigned a Dirichlet prior and mapped to  $(K - 1)$  isometric log-ratio (ILR) coordinates. Covariate and group effects are added in ILR space, and the inverse transformation returns positive source proportions that sum to one (Aitchison, 1982; Egozcue et al., 2003; Parnell et al., 2013). MixSIAPy accepts either raw source observations or summaries consisting of source means, standard deviations, and sample sizes, and either representation may be stratified by one categorical factor. Source, discrimination, proportion, covariate, group-effect, and error parameters are estimated jointly, allowing their uncertainty to propagate into the posterior source proportions. Detailed equations, prior specifications, and error-model definitions are provided in the MixSIAPy User Manual in the Supplement.

## 2.2 Formula-based compositional regression

The conventional MixSIAR interface supports up to two categorical factors and one continuous covariate entered through separate arguments, but it does not construct interactions among predictors. MixSIAPy introduces a methodological extension that embeds formula-defined multivariable regression directly within the Bayesian mixing model. A user-specified formula is automatically parsed into a design matrix containing multiple numerical and categorical predictors and their two-way interactions. These effects are estimated jointly with source proportions, source uncertainty, grouping effects, and error parameters. The methodological novelty therefore lies not in ILR regression or formula expansion separately, but in integrating formula-generated multivariable and interaction effects into the mixing-model likelihood. This extension retains the established ILR representation of compositional proportions while generalizing the predictor structures that can be evaluated



within the model (Aitchison, 1982; Egozcue et al., 2003; Parnell et al., 2013). For observation  $i$ , the ILR predictor is

$$\boldsymbol{\eta}_i = \boldsymbol{\eta}_0 + \mathbf{X}_i \mathbf{B} + \sum_{g=1}^G \mathbf{u}_{g,lg(i)}, \quad (2)$$

where  $\boldsymbol{\eta}_i$  is the  $(K - 1)$  ILR coordinates that determine the  $K$  source proportions for observation  $i$ ;  $\boldsymbol{\eta}_0$  is the baseline composition in ILR coordinates;  $\mathbf{X}_i$  contains the values of the  $P$  formula-generated regression terms for observation  $i$ . The coefficient matrix  $\mathbf{B}$  contains  $(K - 1)$  coefficients for each regression term, corresponding to the  $(K - 1)$  independent ILR coordinates. The final term represents optional random or nested deviations for the groups to which observation  $i$  belongs. The resulting ILR coordinates are transformed back into  $K$  positive source proportions that sum to one. Thus, each predictor changes the relative balance among all sources simultaneously rather than modifying one source proportion independently.

MixSIAPy parses the user-supplied formula and automatically converts it into a regression design matrix  $\mathbf{X}$ . Formula-based predictors can be combined with random or nested grouping effects specified through the conventional factor arguments. Taking the crop-water-uptake case as an example,

```
mix = load_mix_data(  
    filename="crop_water_consumer.csv",  
    iso_names=["delta2H", "delta18O"],  
    factors=["spcomp"],  
    fac_random=[True],  
    fac_nested=[False],  
    composition_formula="species * biomass + species * diversity",  
)
```

MixSIAPy automatically expands “*species \* biomass + species \* diversity*” to “*species + biomass + diversity + species:biomass + species:diversity*”. The symbol “*\**” requests the main effects of both predictors and their interaction, whereas “*:*” denotes the interaction alone. Numerical variables are centred and scaled as continuous covariates, while nonnumeric variables are encoded as categorical contrasts. In this example, “*species*” and “*diversity*” are categorical predictors, biomass is a standardized continuous predictor, and the two interaction terms allow the biomass relationship and planting-diversity contrast to differ among species.

The grouping variable “*spcomp*” is specified separately as a random effect and represents additional variation among planting compositions that is not explained by the formula terms. Thus, the formula describes systematic relationships with species, biomass, and diversity, whereas the random effect represents unexplained differences among particular planting compositions. Regression coefficients, random effects, source parameters, and error parameters are estimated jointly, propagating their uncertainty into the predicted water-source proportions.

### 2.3 Multi-backend Bayesian inference and optional GPU execution

MixSIAR expresses its statistical model in BUGS and delegates posterior sampling to the separately installed JAGS engine, which generally applies conditional samplers to individual stochastic nodes or parameter blocks (Plummer, 2003). MixSIAPy



instead constructs the model as a PyMC computational graph and uses the No-U-Turn Sampler (NUTS) by default. NUTS uses posterior gradients to update continuous parameters jointly and adapts its sampling behaviour during warm-up, which can improve exploration of correlated and high-dimensional posterior distributions (Hoffman and Gelman, 2014; Betancourt, 2017; Abril-Pla et al., 2023).

170

Model construction is separated from inference execution. The same validated model specification can therefore be sampled using native PyMC, Nutpie, NumPyro, or BlackJAX, subject to package availability and model compatibility. Native PyMC is the default and most broadly compatible option. Nutpie provides compiled CPU sampling, whereas NumPyro and BlackJAX use JAX and support CPUs or compatible GPUs (Phan et al., 2019; Cabezas et al., 2024). MixSIAPy verifies the requested backend and device before sampling and records the resolved configuration, package versions, and timing information with the results. An unavailable GPU request raises an error rather than silently reverting to CPU execution. Importantly, GPU execution is not expected to accelerate every analysis because JAX compilation, device initialization, and data transfer can dominate small or short-running models. Benefits are more likely when model computation or chain-level parallelism is sufficient to offset these costs. Section 3 evaluates backend efficiency and CPU–GPU scaling across representative model configurations.

180

The run argument selects one of seven sampling schedules: test, very short, short, normal, long, very long, or extreme. The test schedule checks data and model configuration, while the remaining presets provide progressively larger warm-up and sampling budgets. Sampling effort can be increased for computationally demanding models, but neither the selected backend nor the sampling budget guarantees convergence. All fitted models must therefore be evaluated using posterior and sampler diagnostics.

185

## 2.4 Shared scripted and graphical workflow

MixSIAPy separates data handling, model construction, inference dispatch, posterior processing, visualization, and result export into reusable package components (Fig. 1a). The package also provides example datasets, executable analyses, automated tests, numerical-validation tools, documentation, and deployment resources. A common Python interface connects these components and avoids separate statistical implementations for different forms of access.

190

The analysis follows four steps:

- Step I: Users specify the input files, predictor structure, prior, error model, inference backend, computational device, and sampling settings.
- Step II: The mixture, source, and discrimination data are loaded into validated analysis objects. Validation checks tracer definitions, source names, factor levels, nesting relationships, concentration data, source sample sizes, prior dimensions, and missing values before model construction.

195

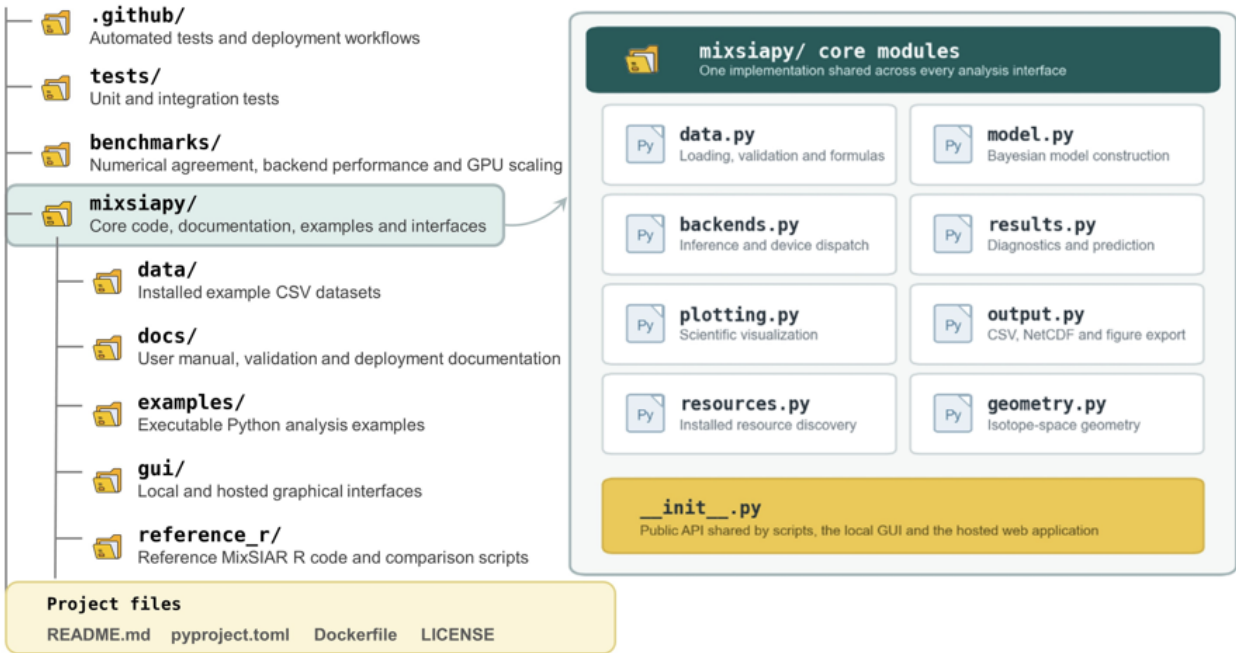


- Step III: The validated data and settings are converted into a PyMC computational graph and dispatched to the selected inference backend and device. Posterior draws and sampler statistics are returned in a common format, allowing subsequent processing to remain independent of the backend used.
- Step IV: Convergence and sampling quality are assessed using rank-normalized  $\hat{R}$ , effective sample sizes, Monte Carlo standard errors, and backend-specific warnings where available. The posterior results are then summarized, visualized, and used to generate source-proportion predictions and contrasts. Outputs include summary tables, figures, NetCDF posterior files, the resolved model configuration, and environment information.

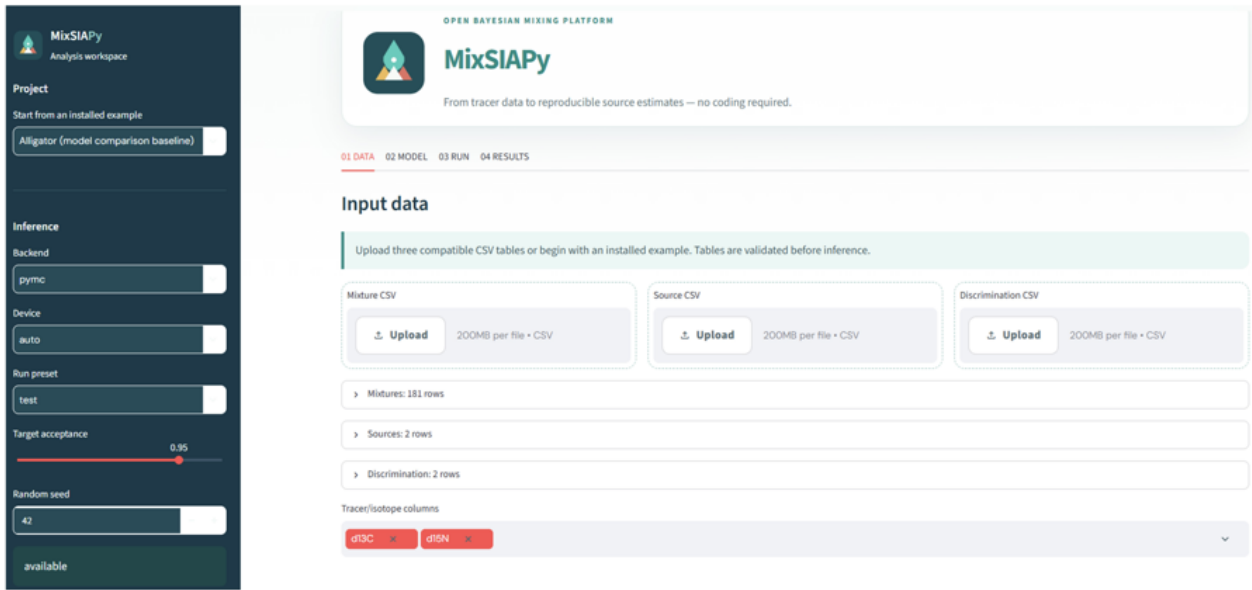
Users can access this workflow through a Python script, a locally installed graphical user interface, or a hosted web application. The local interface opens in a web browser while computation remains on the user's computer, allowing access to locally installed backends and compatible GPUs. The hosted application requires no local Python installation but is limited by the computational and temporary-storage resources of the public deployment (Fig. 1b). Both graphical interfaces use the same validation, model-construction, inference, and result-processing functions as the Python interface. They can also export an executable Python script, allowing an interactively configured analysis to be reproduced, modified, or automated without reconstructing its model specification.



(a) MixSIAPy package organization



(b) Graphical user interface of MixSAIPy



215 **Figure 1: (a) Organization of the MixSIAPy package files, example data and analyses, core statistical modules, tests, validation tools, documentation, and deployment resources. The public application programming interface connects the core modules to the available user interfaces. (b) Graphical user interface for data input, model and sampling configuration, execution monitoring, visualization, and result export. The hosted web application is available via Hugging Face Spaces (<https://huggingface.co/spaces/jyang-research/MixSIAPy>).**



## 220 3 Evaluation design

### 3.1 Numerical agreement benchmark

Numerical agreement was evaluated using 12 datasets distributed with MixSIAR. Alternative configurations of the alligator and killerwhale datasets produced 20 paired benchmark analyses, each fitted independently in MixSIAR and MixSIAPy. The benchmarks covered one to 22 tracers, alternative source data representations, concentration dependence, generalist and  
 225 informative priors, fixed and hierarchical effects, continuous covariates, and all three error models. Both implementations used identical data and model specifications within each pair. Complete configurations are provided in Table A1.

MixSIAR initially used its normal sampling schedule of three chains, 100 000 iterations per chain, 50 000 burn-in iterations, and thinning by 50. MixSIAPy used native PyMC on the CPU with three chains, 5000 warm-up iterations and 1000 retained  
 230 draws per chain. Both implementations therefore retained 3000 posterior draws. Their total iteration counts were not considered directly comparable because JAGS and NUTS use different transition mechanisms. Analyses failing the diagnostic requirements were refitted independently using prespecified longer sampling schedules. A paired analysis was included in the numerical agreement assessment only when both implementations produced an acceptable posterior. Analyses that remained unacceptable under the longest schedule were classified as inconclusive. An acceptable posterior required rank-normalized  
 235 split- $\hat{R}$  values no greater than 1.05 and bulk and tail effective sample sizes of at least 100 for all prespecified comparison parameters (Vehtari et al., 2021). MixSIAPy fits were additionally required to contain no divergent transitions, no transitions reaching the maximum tree depth, and a minimum Bayesian fraction of missing information greater than 0.30. These additional criteria identify numerical problems specific to Hamiltonian Monte Carlo and were not applicable to the JAGS fits.

240 Depending on model structure, the comparison included source proportions, covariate effects and predictions, random-effect standard deviations, source parameters, and error parameters. Agreement was assessed between posterior distributions rather than individual draws. For each matched parameter, the absolute difference between posterior means was standardized by the pooled posterior standard deviation:

$$D_{mean} = \frac{|\mu_R - \mu_P|}{\sqrt{(\sigma_R^2 + \sigma_P^2)/2}}, \quad (3)$$

245 where the  $\mu_R$  and  $\mu_P$  subscripts are the posterior means from MixSIAR and MixSIAPy, respectively. Smaller values indicate closer posterior means relative to posterior uncertainty. Overlap between corresponding 95 % credible intervals was calculated as

$$O_{95} = \frac{\max[0, \min(U_R, U_P) - \max(L_R, L_P)]}{\min(U_R - L_R, U_P - L_P)}, \quad (4)$$

where  $L$  and  $U$  are the lower and upper credible limits. The statistic ranges from zero for disjoint intervals to one when the  
 250 shorter interval is fully contained within the longer interval. Absolute differences between mean source proportions were additionally reported as:



$$\Delta_{pk} = |\bar{p}_{k,R} - \bar{p}_{k,P}|. \quad (5)$$

### 3.2 Evaluation of computational performance

Three comparisons addressed complementary aspects of computational performance. First, MixSIAR–JAGS and MixSIAPy–  
 255 PyMC were compared across all 20 benchmarks using two comparison strategies. The fixed-output comparison strategy used  
 the normal schedule of each implementation, with 3000 retained draws. It measured performance under the standard workload,  
 regardless of whether the posterior met the diagnostic requirements. The matched-convergence comparison strategy instead  
 used the earliest independently fitted schedule that produced an acceptable posterior. Reported wall time referred to the selected  
 successful fit and excluded preceding unsuccessful fits.

260 Second, native PyMC, Nutpie, NumPyro, and BlackJAX were compared on the CPU using the snail, lake, geese, wolves, and  
 cladocera models. These models represented contrasting tracer dimensions, source-data formats, covariate structures,  
 concentration settings, and error models. Each backend used three chains, 5,000 warm-up iterations per chain, 1,000 retained  
 draws per chain, and a target acceptance probability of 0.95. Sampler execution time was recorded for each run and summarized  
 265 by backend across the five models. Configurations failing the diagnostic requirements were retained in the figure but were not  
 used to draw conclusions about reliable backend performance.

Third, CPU and GPU execution were compared within NumPyro for the same five models. Each model was fitted using 4, 8,  
 16, 32, and 64 vectorized chains on both the CPU and an NVIDIA GPU. Each chain used 500 warm-up iterations and retained  
 270 500 posterior draws, with a target acceptance probability of 0.95. Every CPU–GPU configuration was repeated three times.  
 GPU speed-up was calculated as CPU execution time divided by GPU execution time, such that values greater than one  
 indicated faster GPU execution. The high chain counts were used to evaluate parallel scaling and are not recommended as  
 routine convergence settings.

275 All experiments were performed under Windows 10 Pro using R 4.4.1, MixSIAR 3.1.12, rjags 4.17, Python 3.13.9, PyMC  
 5.28.3, and ArviZ 0.23.4. The CPU-backend and GPU experiments were conducted under Ubuntu 18.04 in WSL2 using Python  
 3.11.13, MixSIAPy 0.1.0, PyMC 5.20.1, Nutpie 0.13.4, JAX and jaxlib 0.4.38, NumPyro 0.17.0, and BlackJAX 1.2.5. The  
 platform contained an Intel Core i7-14650HX processor, 96 GB of physical memory, and an NVIDIA GeForce RTX 4060  
 Laptop GPU with 8 GB of video memory.

### 280 3.3 Application to crop-water-uptake from different soil layers

Formula-based compositional regression was evaluated using data from the crop-water-uptake experiment of Schmutz and  
 Schöb (2023). Xylem-water  $\delta^2\text{H}$  and  $\delta^{18}\text{O}$  measurements from six crop species including barley, faba bean, linseed, oilseed



rape, pea, and wheat, were modelled as mixtures of water from shallow (0–10 cm), middle (15–30 cm), and deep (50–75 cm) soil layers. The present analysis focused on monocultures and species mixtures and included 176 xylem-water observations: 17 from monocultures and 159 from mixtures. The source data comprised 24 shallow-, 24 middle-, and 16 deep-soil-water observations. Raw source measurements were used to retain covariance between  $\delta^2\text{H}$  and  $\delta^{18}\text{O}$ . Discrimination means and variances were set to zero because xylem water was assumed to retain the isotopic composition of the absorbed soil water.

The fixed predictor structure was specified using the formula “*species \* biomass + species \* diversity*”, as described in Section 2.2. The formula jointly represented baseline differences among species, species-specific associations between biomass and water-source composition, and species-specific contrasts between monocultures and mixtures. Planting composition was included separately as a random effect to account for additional variation among the particular crop combinations grown together. The model therefore retained the species-specific diversity contrasts considered in the original study while adding species-specific biomass associations within a single joint mixing model. Posterior sampling used native PyMC with four chains, 3000 warm-up iterations per chain, 2000 retained draws per chain, and a target acceptance probability of 0.99.

Three population-level posterior summaries were generated, with the planting-composition random deviation set to zero. First, water-source proportions were predicted for mixture-grown plants at the observed median biomass of each species. Second, biomass-associated change was calculated within mixtures as the difference between predictions at the species-specific 90th and 10th biomass percentiles. Third, the mixture–monoculture contrast was calculated at the midpoint of the biomass range represented in both planting conditions for each species. Posterior medians and 95% credible intervals were reported for the shallow-, middle-, and deep-water proportions and their corresponding contrasts. Because biomass was observed rather than experimentally manipulated, biomass-related coefficients and contrasts were interpreted as conditional associations rather than causal effects.

## 4 Results and discussion

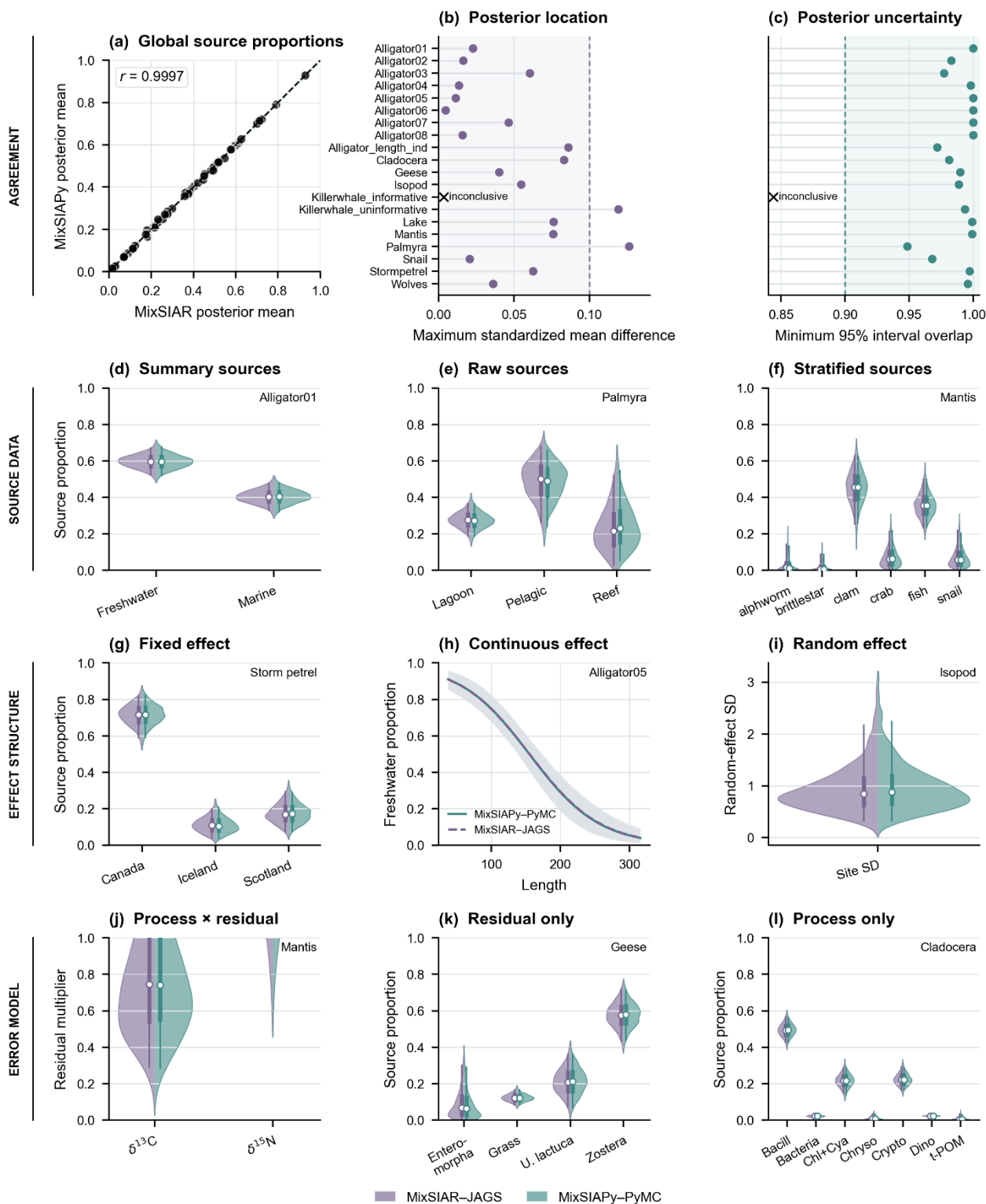
### 4.1 Numerical correspondence with MixSIAR

Nineteen of the 20 paired benchmark analyses satisfied the convergence requirements in both implementations. These analyses covered all eight standard alligator configurations, raw and summarized source data, factor-stratified sources, concentration dependence, alternative fixed and hierarchical effects, continuous covariates, and all three error models (Table A1 in Appendix). The informative-prior killerwhale analysis remained inconclusive under the longest sampling schedule: the MixSIAR fit failed the rank-normalized split- $\hat{R}$  and effective-sample-size requirements, whereas the MixSIAPy fit retained divergent transitions and maximum-tree-depth warnings. This analysis was therefore excluded from the posterior-agreement summaries.



315 The 19 retained pairs contained 57 matched global source-proportion components. Posterior means from the two  
implementations were tightly aligned with the line of equality, with a Pearson correlation coefficient of  $r = 0.9997$  across the  
57 matched source components (Fig. 2a). The median and mean absolute differences in posterior mean source proportions  
were 0.0017 and 0.0035, respectively, and 50 of the 57 components differed by no more than 0.01. The median standardized  
posterior-mean difference was 0.0291, and 17 of the 19 analyses had a maximum value below 0.10 (Fig. 2b). All matched 95  
320 % credible intervals overlapped, and the minimum overlap ratio across the retained analyses was 0.949 (Fig. 2c).

Agreement also extended to model-specific posterior quantities. Representative comparisons showed similar posterior  
distributions for models using summarized source data, raw source observations retaining tracer covariance, factor-stratified  
source distributions, fixed and continuous effects, random-effect variation, and each of the three error models (Fig. 2d–l).  
325 Exact equality was not expected because JAGS and NUTS used different parameterizations, transition mechanisms, and  
independent Monte Carlo realizations. Nevertheless, the observed differences were small relative to posterior uncertainty and  
did not alter the ecological interpretation of any retained benchmark analysis.





330 **Figure 2: Numerical agreement between MixSIAR and MixSIAPy. Only benchmark analyses satisfying the diagnostic requirements**  
**in both implementations were included in the posterior-agreement summaries. (a) Posterior mean global source proportions; the**  
**dashed diagonal denotes equality, and each point represents one matched source component. (b) Maximum standardized posterior-**  
 335 **mean difference within each benchmark analysis. (c) Minimum overlap between matched 95 % credible intervals within each**  
**benchmark analysis. Representative posterior comparisons are shown for (d) summarized source data in Alligator01, (e) raw source**  
**observations retaining tracer covariance in Palmyra, (f) habitat-stratified source distributions in Mantis, (g) a categorical fixed effect**  
**in Stormpetrel, (h) a continuous body-length effect in Alligator05, (i) a random-effect standard deviation in Isopod, (j) combined**  
**process and residual error in Mantis, (k) residual-only error in Geese, and (l) process-only error in Cladocera. The informative-**  
**prior killerwhale analysis is marked as inconclusive and thus is not used for comparison.**

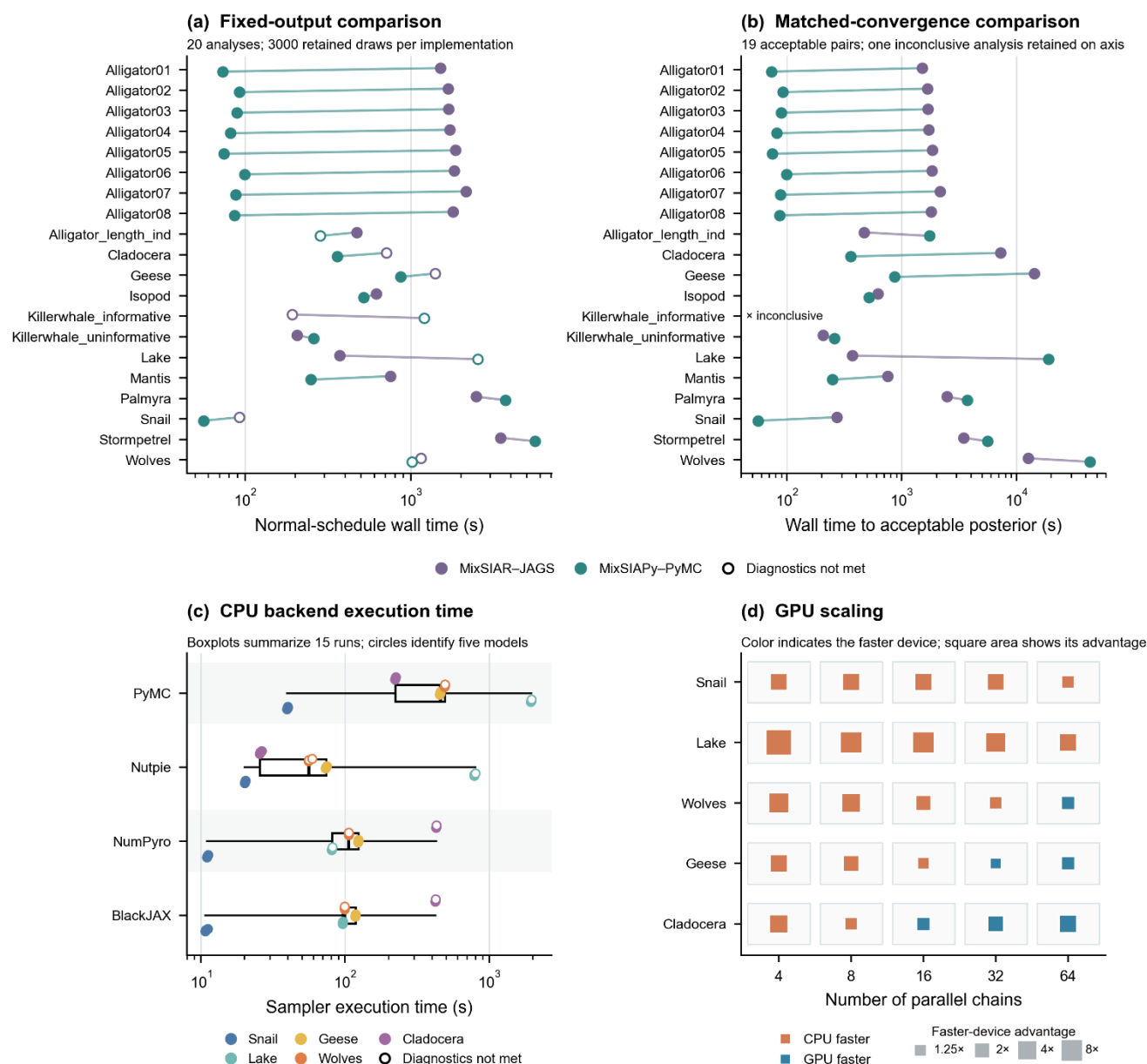
## 4.2 Computational performance across implementations, backends, and devices

340 Under the fixed-output comparison strategy, MixSIAPy completed 15 of the 20 benchmark analyses faster than MixSIAR  
 (Fig. 3a). However, only 13 pairs satisfied the convergence requirements in both implementations under their normal sampling  
 schedules; 16 MixSIAPy fits and 15 MixSIAR fits were acceptable individually. Equal numbers of retained draws therefore  
 did not ensure comparable posterior quality. When performance was instead evaluated using the earliest independently fitted  
 schedule that produced an acceptable posterior, MixSIAPy was faster in 13 of the 19 conclusive comparisons (Fig. 3b). Its  
 345 largest gains occurred across the eight standard alligator analyses and in the geese and cladocera models, whereas MixSIAR  
 was faster in six cases, most notably for lake. Thus, MixSIAPy often reduced the time required to obtain a diagnostically  
 acceptable posterior, but the advantage was model dependent rather than universal.

Sampler execution time also varied substantially among the four MixSIAPy backends (Fig. 3c). Across the five models and  
 350 three independent runs per model, the descriptive median execution times were 456 s for native PyMC, 56 s for Nutpie, 106 s  
 for NumPyro, and 99 s for BlackJAX. These pooled summaries should not be interpreted as a universal backend ranking  
 because model complexity and diagnostic outcomes differed. Among configurations satisfying the diagnostic requirements,  
 BlackJAX was fastest for snail and was the only acceptable backend for lake under the tested workload, whereas Nutpie was  
 fastest for geese and cladocera. Multi-backend support therefore provides more than alternative routes to the same calculation:  
 355 it allows users to change the computational strategy when a particular implementation is inefficient or diagnostically  
 inadequate, without rewriting the underlying statistical model.

GPU execution provided a further scaling option, but only when the parallel workload was sufficiently large (Fig. 3d). For  
 cladocera, the GPU became faster at 16 parallel chains and reached a median speed-up of 3.07 at 64 chains. Geese crossed the  
 360 CPU–GPU break-even point at 32 chains and reached a speed-up of 1.45 at 64 chains, while wolves benefited only at 64  
 chains. GPU execution remained slower for snail and lake at every tested chain count. These results identify the practical  
 boundary of GPU acceleration: compilation, device initialization, and data transfer dominate small workloads, whereas  
 sufficiently parallel and computationally demanding analyses can amortize these costs. High chain counts are not required for  
 routine inference, but they provide a controlled test of the available parallel scaling.

365



**Figure 3: Computational performance across implementations, CPU backends, and computational devices:** (a) Wall time for 20 paired benchmark analyses under the normal sampling schedules, with each implementation retaining 3000 posterior draws. Lines connect corresponding MixSIAR and MixSIAPy fits, and the line color identifies the faster implementation. Filled and open circles denote fits that satisfied or failed the diagnostic requirements, respectively. (b) Wall time of the earliest independently fitted schedule that produced an acceptable posterior. Nineteen analyses yielded acceptable results in both implementations; the informative-prior killer-whale analysis remained inconclusive. (c) Sampler execution time for native PyMC, Nutpie, NumPyro, and BlackJAX on the CPU. Each boxplot summarizes 15 observations comprising five benchmark models and three independent runs per model. Colored circles identify the models, and open circles denote configurations that failed the diagnostic requirements. (d) NumPyro CPU-GPU scaling from 4 to 64 parallel chains. Each square represents the median of three paired runs. Orange and blue indicate faster CPU



and GPU execution, respectively, while square area represents the faster-device advantage, calculated from the CPU-to-GPU wall-time ratio and its reciprocal.

The principal computational contribution of MixSIAPy is therefore flexibility rather than guaranteed acceleration. A common PyMC model can be evaluated using several inference implementations and, where appropriate, CPU or GPU execution. This architecture permits backend and device selection to be guided by posterior diagnostics and measured efficiency for the model at hand, while preserving an identical statistical specification. The benchmarks consequently establish both the benefits and the limits of this flexibility: MixSIAPy often reached acceptable posteriors faster than MixSIAR, alternative backends could rescue or accelerate particular models, and GPU acceleration was useful only above model- and workload-specific thresholds.

### 4.3 Crop water-uptake patterns from compositional regression

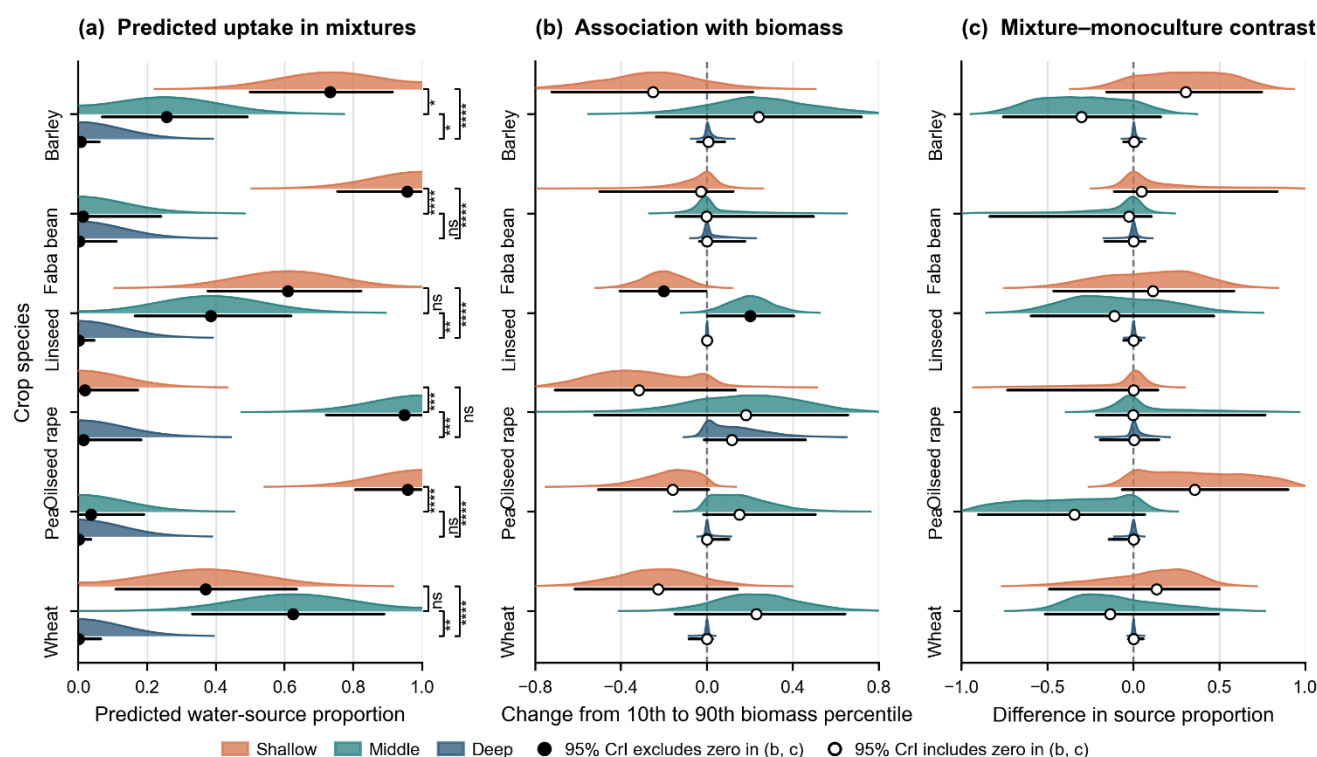
The crop-water model converged without divergent transitions or maximum-tree-depth warnings. The maximum rank-normalized split- $\hat{R}$  was 1.01, and the minimum bulk and tail effective sample sizes were 341 and 393, respectively. At the species-specific median biomass in mixtures, shallow soil water had the largest posterior median contribution for barley, faba bean, linseed, and pea. The corresponding proportions were 0.73, 0.96, 0.61, and 0.96, respectively (Fig. 4a). Pairwise posterior comparisons supported greater shallow-water uptake than both middle- and deep-layer uptake for barley, faba bean, and pea. For linseed, the shallow and middle contributions were not clearly separated, although both exceeded the deep-water contribution. Middle-layer water had the largest median contribution for oilseed rape and wheat, with proportions of 0.95 and 0.62, respectively. Its predominance over shallow water was strongly supported for oilseed rape but remained uncertain for wheat. The median deep-water contribution was below 0.02 for every species. Overall, these predictions reproduced the principal species-level patterns reported by Schmutz and Schöb (2023), while quantifying uncertainty in the differences among soil layers.

The species-by-biomass interaction identified a conditional relationship that was not estimated directly in the original MixSIAR mixing-model analysis. Between the species-specific 10th and 90th biomass percentiles, the median shallow-water proportion for linseed decreased by 0.20. Its 95 % credible interval ranged from -0.40 to -0.004 (Fig. 4b). Over the same biomass range, the middle-water proportion increased by 0.20, with a 95 % credible interval of 0.003 to 0.40. The corresponding intervals included zero for the other species. No species showed a resolved biomass-associated increase in deep-water uptake. Thus, greater biomass was associated with a redistribution from shallow to middle soil water in linseed, rather than a general shift towards deeper uptake. This relationship is conditional on the included predictors and should not be interpreted as a causal biomass effect.

After accounting for biomass and variation among planting compositions, no species-specific mixture-minus-monoculture contrast had a 95 % credible interval that excluded zero (Fig. 4c). This result does not demonstrate the absence of a planting-diversity relationship. The analysis included only 17 monoculture observations, and the two- and three-species treatments were



combined into one mixture category. Both features limited the precision of the contrasts. The estimates therefore represent  
 410 conditional comparisons between plants of similar biomass after accounting for variation among particular planting  
 compositions.



**Figure 4: Water-source compositions estimated using formula-based compositional regression.** Ridgelines show the posterior  
 415 distributions of the shallow-, middle-, and deep-water proportions. Points indicate posterior medians, and horizontal bars indicate  
 95 % credible intervals. (a) Predicted water-source proportions for mixture-grown plants at the observed species-specific median  
 biomass. Brackets indicate pairwise comparisons among soil-water sources within each species. Stars summarize the strength of  
 420 posterior evidence for pairwise differences among water sources; they are based on two-sided posterior tail probabilities  $p_{tail}$  (\*\*\*\*  
 $\leq 0.0001$ , \*\*\*  $\leq 0.001$ , \*\*  $\leq 0.01$ , \*  $\leq 0.05$ , and ns  $> 0.05$ ). (b) Change in each source proportion between the species-specific 10th and  
 90th biomass percentiles within mixtures; positive values indicate an increasing source contribution with biomass. (c) Mixture-  
 minus-monoculture contrasts evaluated at the midpoint of the biomass range represented in both planting conditions; positive values  
 indicate a larger source contribution in mixtures. In panels (b) and (c), filled points indicate that the 95 % credible interval excludes  
 zero, whereas open points indicate that it includes zero; the vertical dashed line marks zero change. All panels show population-level  
 predictions with the planting-composition random deviation set to zero.

This application illustrates the principal regression advantage of MixSIAPy. The standard MixSIAR interface does not provide  
 425 a general formula system for jointly specifying several predictors and their interactions. MixSIAPy estimated species-specific  
 biomass associations and diversity contrasts within one mixing model, while retaining the planting-composition random effect.  
 Source, discrimination, residual, regression, and random-effect uncertainties were propagated jointly to the predicted water-  
 source compositions. This integrated approach also preserved dependence among the three source proportions and avoided a  
 two-stage regression on previously estimated proportions.



#### 430 4.4 Limitations and future development

Formula-based compositional regression expands the range of covariate relationships that can be represented within a mixing model, but it does not overcome limited information in the tracer data or study design. Models containing several predictors and interactions require adequate replication and overlapping covariate ranges. Regression coefficients may remain weakly identified when source signatures overlap, predictor combinations are sparsely represented, or categorical levels contain few  
435 observations. In the crop-water-uptake application, the small number of monoculture observations and the aggregation of two- and three-species treatments limited the precision of the diversity contrasts. Future applications would benefit from more balanced sampling across species, planting-diversity levels, and biomass ranges. Predictions should generally remain within the observed covariate support, and associations with observational predictors such as biomass should not be interpreted causally.

440 Computational performance was conditional on model structure, posterior geometry, software versions, sampling settings, and hardware. Compiled and JAX-based backends incur initialization and compilation costs, while GPU execution introduces additional device-management and data-transfer overhead. The reported speed ratios therefore characterize the evaluated configurations rather than universal differences among backends. Broader benchmarking on larger hierarchical models and  
445 additional hardware platforms will be needed to define more general guidance for backend and device selection.

The hosted graphical interface is additionally constrained by the CPU, memory, execution-time, and temporary-storage limits of its public deployment. It is suitable for model configuration, teaching, test runs, and relatively small analyses, whereas demanding hierarchical models and long sampling schedules are more appropriately run through the Python interface or the  
450 local graphical application. Future development may extend cloud resource management and accelerator support, but these capabilities will remain dependent on the available hosting infrastructure.

## 5 Conclusions

MixSIAPy provides an open-source Python implementation of established Bayesian stable isotope mixing models. It retains the principal source, discrimination, concentration, prior, covariate, and error structures of MixSIAR. It additionally provides  
455 formula-based compositional regression and four interchangeable NUTS backends. Across 19 conclusive paired benchmarks, MixSIAPy and MixSIAR produced closely corresponding posterior distributions. MixSIAPy reached an acceptable posterior faster in 13 comparisons. However, performance varied among models, and no backend or computational device was uniformly optimal. GPU acceleration became advantageous only for selected models under sufficiently parallel workloads.

460 The crop-water-uptake application demonstrated the main statistical extension. Species, biomass, planting diversity, and their interactions were estimated jointly within the isotope mixing likelihood. Planting composition was simultaneously represented



as a random effect. The model identified a biomass-associated redistribution from shallow to middle soil water in linseed. Biomass relationships in the other species and species-specific diversity contrasts remained uncertain. By estimating these relationships jointly, MixSIAPy propagated uncertainty from the isotope mixing model into the regression results. It also  
 465 avoided treating previously estimated source proportions as error-free response variables.

A shared analysis engine supports Python scripts, a local graphical interface, and a hosted web application. These routes provide consistent model configuration, posterior diagnosis, visualization, and reproducible export. MixSIAPy therefore connects established isotope mixing models with the wider Python scientific-computing ecosystem. More broadly, it enables  
 470 interacting environmental controls on water and solute sources to be evaluated directly within source-apportionment models. Reliable inference still depends on informative tracers, adequate sampling, appropriate covariate coverage, and careful evaluation of posterior diagnostics.

## Appendix: Benchmark configurations

**Table A1: Complete configuration of the 20 paired benchmark analyses.**

| ID | Benchmark   | Mixtures and isotope tracers                                | Source-data organization     | Effects and source stratification                                        | Error model               | Primary quantities compared                                                 |
|----|-------------|-------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------|
| 1  | Alligator01 | $N = 181$ ; $\delta^{13}\text{C}$ and $\delta^{15}\text{N}$ | Source means, SDs, and $n_k$ | No effect                                                                | Process $\times$ residual | Global source proportions and residual multipliers                          |
| 2  | Alligator02 |                                                             |                              | Habitat fixed effect                                                     | Process $\times$ residual | Habitat-specific proportions and contrasts                                  |
| 3  | Alligator03 |                                                             |                              | Sex fixed effect                                                         | Process $\times$ residual | Sex-specific proportions and contrasts                                      |
| 4  | Alligator04 |                                                             |                              | Size-class fixed effect                                                  | Process $\times$ residual | Size-class-specific proportions and contrasts                               |
| 5  | Alligator05 |                                                             |                              | Continuous length effect                                                 | Process $\times$ residual | ILR length coefficient and source-proportion prediction curves              |
| 6  | Alligator06 |                                                             |                              | Sex and size-class fixed effects                                         | Process $\times$ residual | Adjusted sex and size-class contrasts and fitted compositions               |
| 7  | Alligator07 |                                                             |                              | Sex fixed effect and continuous length effect; common length coefficient | Process $\times$ residual | Sex contrast, common length coefficient, and sex-specific prediction curves |



|    |                           |                                                             |                                                    |                                                        |                    |                                                                                     |
|----|---------------------------|-------------------------------------------------------------|----------------------------------------------------|--------------------------------------------------------|--------------------|-------------------------------------------------------------------------------------|
| 8  | Alligator08               |                                                             |                                                    | Combined sex–size-class fixed effect                   | Process × residual | Group-specific source proportions                                                   |
| 9  | Alligator_length_ind      |                                                             |                                                    | Continuous length effect and individual random effect  | Process only       | Length slope, individual-effect SD, and predicted compositions                      |
| 10 | Cladocera                 | $N = 14$ ; 22 fatty-acid tracers                            | Source means and SDs with large $n_k$              | Sample ID fixed effect; each mixture fitted separately | Process only       | Individual compositions and process-derived variances                               |
| 11 | Geese                     | $N = 251$ ; $\delta^{13}\text{C}$ and $\delta^{15}\text{N}$ | Source means, SDs, and $n_k$                       | Group fixed effect                                     | Residual only      | Group-specific proportions and concentration-weighted mixture means                 |
| 12 | Isopod                    | $N = 30$ ; 8 fatty-acid tracers                             | Source means and SDs with large $n_k$              | Site random effect                                     | Residual only      | Global and site-level proportions, partial pooling, and site-effect SD              |
| 13 | Killerwhale_informative   | $N = 12$ ; $\delta^{13}\text{C}$ and $\delta^{15}\text{N}$  | Source means, SDs, and $n_k$                       | No effect                                              | Process × residual | Informative-prior updating, global proportions, and error parameters                |
| 14 | Killerwhale_uninformative |                                                             |                                                    | No effect                                              | Process × residual | Global proportions, error parameters, and comparison with the informative-prior fit |
| 15 | Lake                      | $N = 21$ ; $\delta^{13}\text{C}$ and $\delta^{15}\text{N}$  | Raw observations retaining tracer covariance       | Continuous Secchi-depth effect                         | Residual only      | ILR slope and source-proportion prediction curves                                   |
| 16 | Mantis                    | $N = 31$ ; $\delta^{13}\text{C}$ and $\delta^{15}\text{N}$  | Source means, SDs, and $n$ , stratified by habitat | Habitat fixed effect and habitat-specific sources      | Process × residual | Habitat contrasts, source proportions, and residual multipliers                     |
| 17 | Palmyra                   | $N = 92$ ; $\delta^{13}\text{C}$ and $\delta^{15}\text{N}$  | Raw observations retaining tracer covariance       | Taxon fixed effect                                     | Process × residual | Population- and taxon-specific proportions                                          |
| 18 | Snail                     | $N = 58$ ; $\delta^{13}\text{C}$                            | Raw observations                                   | No effect                                              | Process × residual | Global source proportions and tracer variance                                       |



|    |             |                                                             |                                                     |                                                           |                           |                                                                         |
|----|-------------|-------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------|---------------------------|-------------------------------------------------------------------------|
| 19 | Stormpetrel | N = 135;<br>$\delta^{13}\text{C}$ and $\delta^{15}\text{N}$ | Raw observations retaining tracer covariance        | Region fixed effect                                       | Process $\times$ residual | Regional proportions and contrasts                                      |
| 20 | Wolves      | N = 66;<br>$\delta^{13}\text{C}$ and $\delta^{15}\text{N}$  | Source means, SDs, and $n_k$ , stratified by region | Region and pack random effects; pack nested within region | Process $\times$ residual | Global, regional, and pack-level proportions and both random-effect SDs |

475 Notes:  $N$  is the number of mixture observations,  $n_k$  is the reported sample size for source  $k$ , SD denotes standard deviation. “No effect” denotes an intercept-only compositional model. “Process  $\times$  residual” denotes the combined process- and residual-error model.

### Code and data availability

MixSIAPy, the benchmark scripts, example data, and the hosted application source as well as the user manual are available at  
 480 <https://github.com/jyangfsu/MixSIAPy>. The public graphical application is available at <https://huggingface.co/spaces/jyang-research/MixSIAPy>.

### Author contribution

JY: conceptualization, software, formal analysis, investigation, validation, visualization, funding acquisition, and writing (original draft, review, and editing). YZ and HQ: methodology, supervision, resources, project administration, and writing  
 485 (review and editing). HD, SC, and MY: writing (review and editing).

### Competing interests

At least one of the (co-)authors is a member of the editorial board of Hydrology and Earth System Sciences.

### Acknowledgements

During manuscript preparation, the authors used ChatGPT for language editing. The authors reviewed the resulting text and  
 490 take full responsibility for the content.



## Financial support

This work was supported by the National Natural Science Foundation of China (grant 42302288), the Hubei Province Science and Technology Innovation Platform Project (grant 2025CSA007), and the Fundamental Research Funds for the Central Universities, Chang'an University (grant 300102356102).

## References

- Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fonnesbeck, C. J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., Osthege, M., Vieira, R., Wiecki, T., and Zinkov, R.: PyMC: a modern and comprehensive probabilistic programming framework in Python, *PeerJ Comput. Sci.*, 9, e1516, <https://doi.org/10.7717/peerj-cs.1516>, 2023.
- Aitchison, J.: The statistical analysis of compositional data, *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 44, 139–177, <https://doi.org/10.1111/j.2517-6161.1982.tb01195.x>, 1982.
- Allen, S. T., Kirchner, J. W., Braun, S., Siegwolf, R. T. W., and Goldsmith, G. R.: Seasonal origins of soil water used by trees, *Hydrol. Earth Syst. Sci.*, 23, 1199–1210, <https://doi.org/10.5194/hess-23-1199-2019>, 2019.
- Betancourt, M.: A conceptual introduction to Hamiltonian Monte Carlo, *arXiv [preprint]*, arXiv:1701.02434, <https://doi.org/10.48550/arXiv.1701.02434>, 9 January 2017.
- Beyer, M., Kühnhammer, K., and Dubbert, M.: In situ measurements of soil and plant water isotopes: a review of approaches, practical considerations and a vision for the future, *Hydrol. Earth Syst. Sci.*, 24, 4413–4440, <https://doi.org/10.5194/hess-24-4413-2020>, 2020.
- Cabezas, A., Corenflos, A., Lao, J., Louf, R., Carnec, A., Chaudhari, K., Cohn-Gordon, R., Coullon, J., Deng, W., Duffield, S., Durán-Martín, G., Elantkowski, M., Foreman-Mackey, D., Gregori, M., Iguaran, C., Kumar, R., Lysy, M., Murphy, K., Orduz, J. C., Patel, K., Wang, X., and Zinkov, R.: BlackJAX: composable Bayesian inference in JAX, *arXiv [preprint]*, arXiv:2402.10797, <https://doi.org/10.48550/arXiv.2402.10797>, 16 February 2024.
- Cao, X., He, W., He, W., Shi, Y., An, T., Wang, X., Liu, F., Zhao, Y., Zhou, P., Chen, C., and He, J.: EMMTE: An Excel VBA tool for source apportionment of nitrate based on the stable isotope mixing model. *Sci. Total Environ.*, 868, 161728, <https://doi.org/10.1016/j.scitotenv.2023.161728>
- Couvreux, V., Rothfuss, Y., Meunier, F., Bariac, T., Biron, P., Durand, J.-L., Richard, P., and Javaux, M.: Disentangling temporal and population variability in plant root water uptake from stable isotopic analysis: when rooting depth matters in labeling studies, *Hydrol. Earth Syst. Sci.*, 24, 3057–3075, <https://doi.org/10.5194/hess-24-3057-2020>, 2020.
- Cooper, R. J. and Krueger, T.: An extended Bayesian sediment fingerprinting mixing model for the full Bayes treatment of geochemical uncertainties, *Hydrol. Process.*, 31, 1900–1912, <https://doi.org/10.1002/hyp.11154>, 2017.



- 520 Egozcue, J. J., Pawlowsky-Glahn, V., Mateu-Figueras, G., and Barceló-Vidal, C.: Isometric logratio transformations for compositional data analysis, *Math. Geol.*, 35, 279–300, <https://doi.org/10.1023/A:1023818214614>, 2003.
- Erhardt, E. B., and Bedrick, E. J.: A Bayesian framework for stable isotope mixing models, *Environ. Ecol. Stat.*, 20, 377–397, <https://doi.org/10.1007/s10651-012-0224-1>, 2013.
- Fernandes, R., Millard, A. R., Brabec, M., Nadeau, M.-J., and Grootes, P. M.: Food reconstruction using isotopic transferred signals (FRUITS): a Bayesian model for diet reconstruction, *PLoS One*, 9, e87436, <https://doi.org/10.1371/journal.pone.0087436>, 2014.
- 525 Gibrilla, A., Fianko, J. R., Ganyaglo, S., Adomako, D., Anornu, G., and Zakaria, N.: Nitrate contamination and source apportionment in surface and groundwater in Ghana using dual isotopes ( $^{15}\text{N}$  and  $^{18}\text{O}$ - $\text{NO}_3$ ) and a Bayesian isotope mixing model, *J. Contam. Hydrol.*, 233, 103658, <https://doi.org/10.1016/j.jconhyd.2020.103658>, 2020.
- 530 Govan, E., Jackson, A. L., Bearhop, S., Inger, R., Stock, B. C., Semmens, B. X., Ward, E. J., and Parnell, A.: cosimmr: an R package for fast fitting of stable isotope mixing models with covariates, *arXiv [preprint]*, arXiv:2408.17230, <https://doi.org/10.48550/arXiv.2408.17230>, 30 August 2024.
- Govan, E. and Parnell, A.: simmr: A Stable Isotope Mixing Model, R package version 0.5.1.216, <https://CRAN.R-project.org/package=simmr>, 2019.
- 535 Greenacre, M.: Compositional data analysis, *Annu. Rev. Stat. Appl.*, 8, 271–299, <https://doi.org/10.1146/annurev-statistics-042720-124436>, 2021.
- He, Z., Unger-Shayesteh, K., Vorogushyn, S., Weise, S. M., Duethmann, D., Kalashnikova, O., Gafurov, A., and Merz, B.: Comparing Bayesian and traditional end-member mixing approaches for hydrograph separation in a glacierized basin, *Hydrol. Earth Syst. Sci.*, 24, 3289–3309, <https://doi.org/10.5194/hess-24-3289-2020>, 2020.
- 540 He, S., Li, P., Su, F., Wang, D., and Ren, X.: Identification and apportionment of shallow groundwater nitrate pollution in Weining Plain, northwest China, using hydrochemical indices, nitrate stable isotopes, and the new Bayesian stable isotope mixing model (MixSIAR), *Environ. Pollut.*, 298, 118852, <https://doi.org/10.1016/j.envpol.2022.118852>, 2022.
- Heikkinen, R., Hämäläinen, H., Kiljunen, M., Kärkkäinen, S., Schilder, J., and Jones, R. I.: A Bayesian stable isotope mixing model for coping with multiple isotopes, multiple trophic steps and small sample sizes, *Methods Ecol. Evol.*, 13, 2586–  
545 2602, <https://doi.org/10.1111/2041-210X.13989>, 2022.
- Hoffman, M. D., and Gelman, A.: The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo, *J. Mach. Learn. Res.*, 15, 1593–1623, 2014.
- Hopkins, J. B., and Ferguson, J. M.: Estimating the diets of animals using stable isotopes and a comprehensive Bayesian mixing model, *PLoS One*, 7, e28478, <https://doi.org/10.1371/journal.pone.0028478>, 2012.
- 550 Jameel, Y., Brewer, S., Fiorella, R. P., Tipple, B. J., Terry, S., and Bowen, G. J.: Isotopic reconnaissance of urban water supply system dynamics, *Hydrol. Earth Syst. Sci.*, 22, 6109–6125, <https://doi.org/10.5194/hess-22-6109-2018>, 2018.



- Ji, X., Shu, L., Li, J., Zhao, C., Chen, W., Chen, Z., Shang, X., Dahlgren, R. A., Yang, Y., and Zhang, M.: Tracing nitrate sources and transformations using  $\Delta^{17}\text{O}$ ,  $\delta^{15}\text{N}$ , and  $\delta^{18}\text{O}\text{-NO}_3^-$  in a coastal plain river network of eastern China, *J. Hydrol.*, 610, 127829, <https://doi.org/10.1016/j.jhydrol.2022.127829>, 2022.
- 555 Landgraf, J., Tetzlaff, D., Dubbert, M., Dubbert, D., Smith, A., and Soulsby, C.: Xylem water in riparian willow trees (*Salix alba*) reveals shallow sources of root water uptake by in situ monitoring of stable water isotopes, *Hydrol. Earth Syst. Sci.*, 26, 2073–2092, <https://doi.org/10.5194/hess-26-2073-2022>, 2022.
- Lobry, J., Mounier, F., Ballutaud, M., Chevillot, X., Gascuel, D., Budzinski, H., Labadie, P., and Drouineau, H.: ESCROpath, a Bayesian mixing model to quantify diets and trophic flows in aquatic food webs, *Methods Ecol. Evol.*, 13, 894–907, <https://doi.org/10.1111/2041-210X.13808>, 2022.
- 560 Moore, J. W., and Semmens, B. X.: Incorporating uncertainty and prior information into stable isotope mixing models, *Ecol. Lett.*, 11, 470–480, <https://doi.org/10.1111/j.1461-0248.2008.01163.x>, 2008.
- Parnell, A. C., Inger, R., Bearhop, S., and Jackson, A. L.: Source partitioning using stable isotopes: coping with too much variation, *PLoS One*, 5, e9672, <https://doi.org/10.1371/journal.pone.0009672>, 2010.
- 565 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.: Bayesian stable isotope mixing models, *Environmetrics*, 24, 387–399, <https://doi.org/10.1002/env.2221>, 2013.
- Phan, D., Pradhan, N., and Jankowiak, M.: Composable effects for flexible and accelerated probabilistic programming in NumPyro, *arXiv [preprint]*, arXiv:1912.11554, <https://doi.org/10.48550/arXiv.1912.11554>, 24 December 2019.
- 570 Phillips, D. L., and Gregg, J. W.: Source partitioning using stable isotopes: coping with too many sources, *Oecologia*, 136, 261–269, <https://doi.org/10.1007/s00442-003-1218-3>, 2003.
- Phillips, D. L., Inger, R., Bearhop, S., Jackson, A. L., Moore, J. W., Parnell, A. C., Semmens, B. X., and Ward, E. J.: Best practices for use of stable isotope mixing models in food-web studies, *Can. J. Zool.*, 92, 823–835, <https://doi.org/10.1139/cjz-2014-0127>, 2014.
- 575 Plummer, M.: JAGS: a program for analysis of Bayesian graphical models using Gibbs sampling, in: *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, edited by: Hornik, K., Leisch, F., and Zeileis, A., Vienna, Austria, 1–10, 2003.
- Popp, A. L., Beria, H., Sprenger, M., Ala-Aho, P., Coenders-Gerrits, M., Groh, J., Klaus, J., Knapp, J. L. A., Koren, G., Bakiri, I., Fei, E. X., Gillon, M., Harman, C., Hissler, C., Holmes, T., Jeelani, G., Kalvans, A., Montemagno, A., Öztürk, E. Z.,   
 580 Rožič, P.Ž., Stadnyk, T., Stumpp, C., Valiente, N., von Freyberg, J., Vreča, P., Zuecco, G., van Meerveld, I., Penna, D., and Kirchner, J. W.: Recent advances in tracer-aided mixing modeling of water in the critical zone, *Rev. Geophys.*, 63, e2024RG000866, <https://doi.org/10.1029/2024RG000866>, 2025.



- Qiu, X., Zhang, M., Wang, S., Meng, H., and Che, C.: Comparison of stable isotope mixing models for examining plant root water uptake, *PLoS One*, 20, e0318771, <https://doi.org/10.1371/journal.pone.0318771>, 2025.
- 585 Rothfuss, Y., and Javaux, M.: Reviews and syntheses: isotopic approaches to quantify root water uptake: a review and comparison of methods, *Biogeosciences*, 14, 2199–2224, <https://doi.org/10.5194/bg-14-2199-2017>, 2017.
- Salvatier, J., Wiecki, T. V., and Fonnesbeck, C.: Probabilistic programming in Python using PyMC3, *PeerJ Comput. Sci.*, 2, e55, <https://doi.org/10.7717/peerj-cs.55>, 2016.
- Schmutz, A., and Schöb, C.: Crops grown in mixtures show niche partitioning in spatial water uptake, *J. Ecol.*, 111, 1151–  
590 1165, <https://doi.org/10.1111/1365-2745.14088>, 2023.
- Silberberger, M. J., Koziorowska-Makuch, K., Kuliński, K., and Kędra, M.: Stable isotope mixing models are biased by the choice of sample preservation and pre-treatment: implications for studies of aquatic food webs, *Front. Mar. Sci.*, 7, 621978, <https://doi.org/10.3389/fmars.2020.621978>, 2021.
- Stock, B. C., Jackson, A. L., Ward, E. J., Parnell, A. C., Phillips, D. L., and Semmens, B. X.: Analyzing mixing systems using  
595 a new generation of Bayesian tracer mixing models, *PeerJ*, 6, e5096, <https://doi.org/10.7717/peerj.5096>, 2018.
- Su, H., Liu, Y., Yang, J., Liu, M., Dai, H., Wu, Y., Ye, M., Zhao, M., and Zhang, F.: Unraveling spatiotemporal variability of water geochemistry and sulfate sources in a multi-tributary river system, *J. Hydrol.*, 662, 134073, <https://doi.org/10.1016/j.jhydrol.2025.134073>, 2025.
- Swan, G. J. F., Bearhop, S., Redpath, S. M., Silk, M. J., Goodwin, C., Inger, R., and McDonald, R. A.: Evaluating Bayesian  
600 stable isotope mixing models of wild animal diet and the effects of trophic discrimination factors and informative priors, *Methods Ecol. Evol.*, 11, 139–149, <https://doi.org/10.1111/2041-210X.13311>, 2020.
- Upadhayay, H. R., Bodé, S., Griepentrog, M., Huygens, D., Bajracharya, R. M., Blake, W. H., Dercon, G., Mabit, L., Gibbs, M., Semmens, B. X., Stock, B. C., Cornelis, W., and Boeckx, P.: Methodological perspectives on the application of compound-specific stable isotope fingerprinting for sediment source apportionment, *J. Soils Sediments*, 17, 1537–1553,  
605 <https://doi.org/10.1007/s11368-017-1706-4>, 2017.
- Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C.: Rank-normalization, folding, and localization: an improved R-hat for assessing convergence of MCMC, *Bayesian Anal.*, 16, 667–718, <https://doi.org/10.1214/20-BA1221>, 2021.
- Wang, J., Lu, N., and Fu, B.: Inter-comparison of stable isotope mixing models for determining plant water source partitioning,  
610 *Sci. Total Environ.*, 666, 685–693, <https://doi.org/10.1016/j.scitotenv.2019.02.262>, 2019.
- Yu, L., Zheng, T., Zheng, X., Hao, Y., Yuan, R.: Nitrate source apportionment in groundwater using Bayesian isotope mixing model based on nitrogen isotope fractionation, *Sci. Total Environ.*, 718, 137242, <https://doi.org/10.1016/j.scitotenv.2020.137242>.



- 615 Zhang, J., Zhang, L., Zheng, T., Jin, M., Kang, F., Jiang, J., Yuan, Z., and Luo, J.: Tracing nitrate contamination sources and transformations in a rural–urban karst groundwater system in north China using multiple isotopes and Simmr modelling, Water Resour. Res., 61, e2025WR040156, <https://doi.org/10.1029/2025WR040156>.