Introduction to cosimmr

Emma Govan

10 April, 2024

Learning outcomes

- Understand how cosimmr extends simmr

Learning outcomes

- Understand how cosimmr extends simmr
- How to use cosimmr:

Learning outcomes

- Understand how cosimmr extends simmr
- How to use cosimmr:
- Loading data into cosimmr

Learning outcomes

- Understand how cosimmr extends simmr
- How to use cosimmr:
- Loading data into cosimmr
- Running cosimmr

Learning outcomes

- Understand how cosimmr extends simmr
- How to use cosimmr:
- Loading data into cosimmr
- Running cosimmr
- Outputs from cosimmr

Revision: simmr model

- The simmr model defined earlier:

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

Revision: simmr model

- The simmr model defined earlier:

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

- We also have prior distributions on p (CLR) and σ^2 (gamma)

The CLR prior

- the *centralised log ratio* (CLR) or *softmax* transformation:

$$[p_1, \ldots, p_K] = \left[\frac{\exp(f_1)}{\sum_j \exp(f_j)}, \ldots, \frac{\exp(f_K)}{\sum_k \exp(f_k)}\right]$$

The CLR prior (continued)

- In CLR regression, we put a prior on the f_k

The CLR prior (continued)

- In CLR regression, we put a prior on the f_k
- The CLR transformation guarantees that all the dietary proportions will sum to 1

The CLR prior (continued)

- In CLR regression, we put a prior on the f_k
- The CLR transformation guarantees that all the dietary proportions will sum to 1
- In simmr the prior on f_k is:

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

Expanding the model

- We currently assume all consumers are identical

Expanding the model

- We currently assume all consumers are identical
- We might want to include covariates (height, weight, etc)

Random effects for individuals

- We don’t necessarily need a grouping structure (e.g. pack, sex, etc) to be able to include random effects in a SIMM

Random effects for individuals

- We don't necessarily need a grouping structure (e.g. pack, sex, etc) to be able to include random effects in a SIMM
- In a SIMM we might reasonably assume that every consumer is eating something slightly different and want to quantify the overall mean diet as well as the variability between consumers

Random effects for individuals

- We don't necessarily need a grouping structure (e.g. pack, sex, etc) to be able to include random effects in a SIMM
- In a SIMM we might reasonably assume that every consumer is eating something slightly different and want to quantify the overall mean diet as well as the variability between consumers
- We can do this by modelling each consumer's dietary proportion p_{ik} with a normally distributed prior on the CLR transform of p

How we include covariates

- We rewrite f so it incorporates a matrix of covariate values.

How we include covariates

- We rewrite f so it incorporates a matrix of covariate values.
- f is given as

$$f_{ik} = X_{ic}\beta_{ck}$$

How we include covariates

- We rewrite f so it incorporates a matrix of covariate values.
- *f* is given as

$$f_{ik} = X_{ic}\beta_{ck}$$

- Where X is the matrix of covariate values and β is given a multivariate normal distribution.

$$\beta \sim mvn(\mu_{\beta}, \Sigma_{\beta})$$

Why covariates are useful

- Avoid pseudoreplication (Hulbert 1984)

Why covariates are useful

- Avoid pseudoreplication (Hulbert 1984)
- Samples may be replicated but treatments aren't!

The updated model

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

The updated model (cont)

- This model allows users to include covariates

The updated model (cont)

- This model allows users to include covariates
- We also get unique results for each consumer

The package cosimmr

Background of the package

- Uses Fixed Form Variational Bayes

Background of the package

- Uses Fixed Form Variational Bayes
- This is an optimisation-based algorithm

Background of the package

- Uses Fixed Form Variational Bayes
- This is an optimisation-based algorithm
- Quicker than sampling-based algorithms (e.g. JAGS)

FFVB

- Works by minimising the divergence between an assumed distribution and the posterior distribution

FFVB

- Works by minimising the divergence between an assumed distribution and the posterior distribution
- Tran et al 2021 is a useful paper if you'd like more detail!

How to use cosimmr

- https://github.com/emmagovan/cosimmr

How to use cosimmr

- https://github.com/emmagovan/cosimmr
- Might need to currently download individual functions - work in progress!

Structure of data to be loaded in

```
cosimmr_in <- cosimmr_load(formula,
                         source_names,
                         source_means,
                         source_sds,
                         correction_means,
                         correction_sds,
                         concentration_means,
                         scale_x = TRUE)
```

- Same format as simmr

Structure of data to be loaded in

```
cosimmr_in <- cosimmr_load(formula,
                         source_names,
                         source_means,
                         source_sds,
                         correction_means,
                         correction_sds,
                         concentration_means,
                         scale_x = TRUE)
```

- Same format as simmr
- Main difference is the *formula*

Structure of data to be loaded in

```
cosimmr_in <- cosimmr_load(formula,
                           source_names,
                           source_means,
                           source_sds,
                           correction_means,
                           correction_sds,
                           concentration_means,
                           scale_x = TRUE)
```

- Same format as simmr
- Main difference is the *formula*
- To run the same as simmr just use `mixture ~ 1`

Structure of data to be loaded in

```
cosimmr_in <- cosimmr_load(formula,
                         source_names,
                         source_means,
                         source_sds,
                         correction_means,
                         correction_sds,
                         concentration_means,
                         scale_x = TRUE)
```

- Same format as simmr
- Main difference is the *formula*
- To run the same as simmr just use `mixture ~ 1`
- Input as `mixture ~ covariate1 + covariate2`

Structure of data to be loaded in

```
cosimmr_in <- cosimmr_load(formula,
                         source_names,
                         source_means,
                         source_sds,
                         correction_means,
                         correction_sds,
                         concentration_means,
                         scale_x = TRUE)
```

- Same format as simmr
- Main difference is the *formula*
- To run the same as simmr just use `mixture ~ 1`
- Input as `mixture ~ covariate1 + covariate2`
- `scale_x` for covariates

Structure of data to be loaded in

```
cosimmr_in <- cosimmr_load(formula,
                           source_names,
                           source_means,
                           source_sds,
                           correction_means,
                           correction_sds,
                           concentration_means,
                           scale_x = TRUE)
```

- Same format as simmr
- Main difference is the *formula*
- To run the same as simmr just use `mixture ~ 1`
- Input as `mixture ~ covariate1 + covariate2`
- `scale_x` for covariates
- `as.factor` for categorical variables

Running cosimmr

```
cosimmr_out = cosimmr_ffvb(cosimmr_in)
```

- Again similar to simmr

Running cosimmr

```
cosimmr_out = cosimmr_ffvb(cosimmr_in)
```

- Again similar to simmr
- Can adjust priors and FFVB parameters if you wish

Alligator Data example

- To illustrate we will use data that comes with cosimmr

Alligator Data example

- To illustrate we will use data that comes with cosimmr
- From Nifong et al, 2015

Loading Data

```r
data("alligator_data")

cosimmr_alli <-cosimmr_load(
 formula = as.matrix(alligator_data$mixtures) ~ alligator_data$length,
 source_names = alligator_data$source_names,
 source_means = as.matrix(alligator_data$source_means),
 source_sds = as.matrix(alligator_data$source_sds),
 correction_means = as.matrix(alligator_data$TEF_means),
 correction_sds = as.matrix(alligator_data$TEF_sds))
```

Isospace Plot

- We can create an isospace plot to confirm all the individuals lie within the mixing polygon, similarly to simmr

```
plot(cosimmr_alli)
```

Isospace Plot

- We can create an isospace plot to confirm all the individuals lie within the mixing polygon, similarly to simmr

```
plot(cosimmr_alli)
```

- Like simmr, its important to check that all individuals lie within the mixing polygon

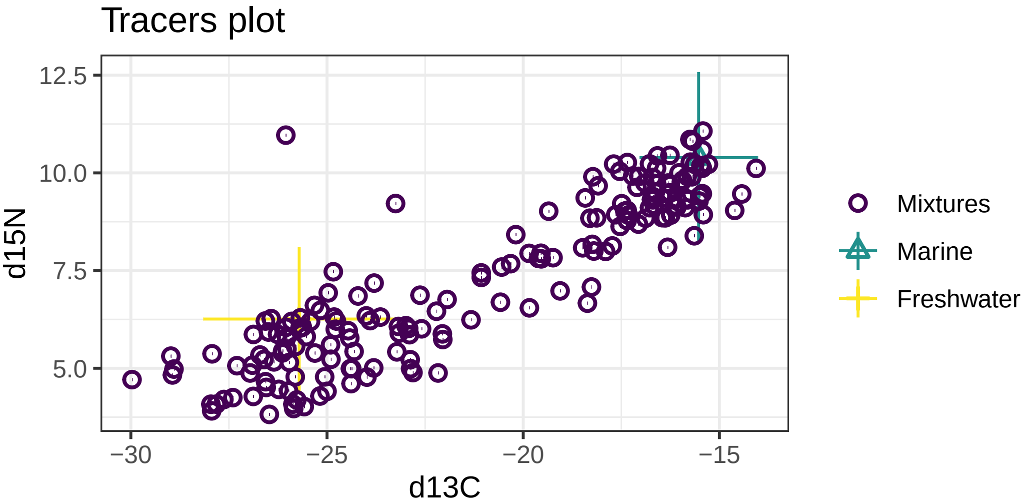
Isospace Plot

- We can create an isospace plot to confirm all the individuals lie within the mixing polygon, similarly to simmr

```
plot(cosimmr_alli)
```

- Like simmr, its important to check that all individuals lie within the mixing polygon
- If they don't it indicates something is wrong - perhaps with data collection or with TEF calculation

Isospace Plot



Running cosimmr

```
cosimmr_alli_out = cosimmr_ffvb(cosimmr_alli)
```

Outputs

Summary

```r
summary(cosimmr_alli_out, type = c("statistics"))
```

```
Summary for Individual 1


            mean    sd
Marine     0.748 0.017
Freshwater 0.252 0.017
d13C       2.010 0.504
d15N       0.549 0.200
```

- Defaults to info on first individual

Summary

```
summary(cosimmr_alli_out, type = c("statistics"))
```

```
Summary for Individual 1


            mean    sd
Marine     0.748 0.017
Freshwater 0.252 0.017
d13C       2.010 0.504
d15N       0.549 0.200
```

- Defaults to info on first individual
- Other options - quantiles, correlations

Summary

```r
summary(cosimmr_alli_out, type = c("statistics"))
```

```
Summary for Individual 1


            mean    sd
Marine     0.748 0.017
Freshwater 0.252 0.017
d13C       2.010 0.504
d15N       0.549 0.200
```

- Defaults to info on first individual
- Other options - quantiles, correlations
- Gives an overview of their diet

Plots

- Multiple options

```
plot(cosimmr_alli_out,
     type = c("isospace", "beta_histogram", "p_ind"),
     ind = 1)
```

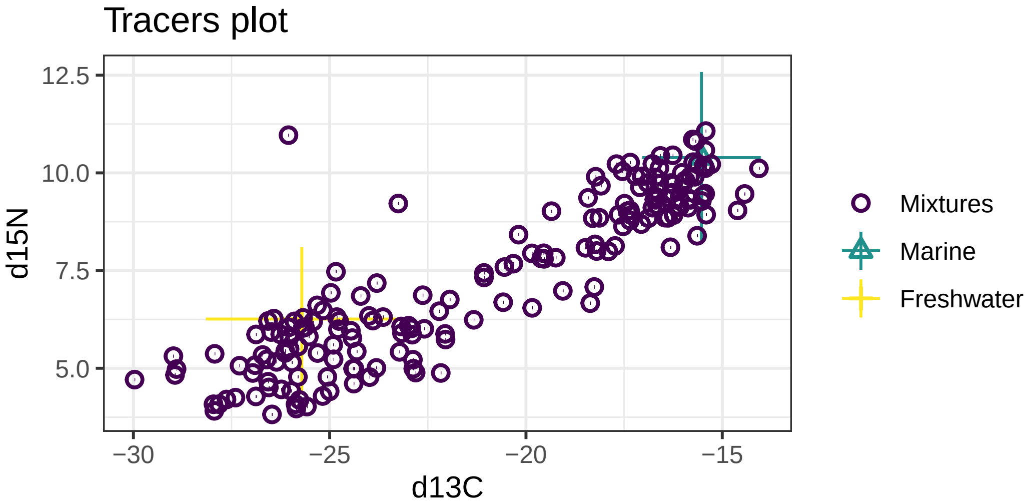
Plots

- Multiple options

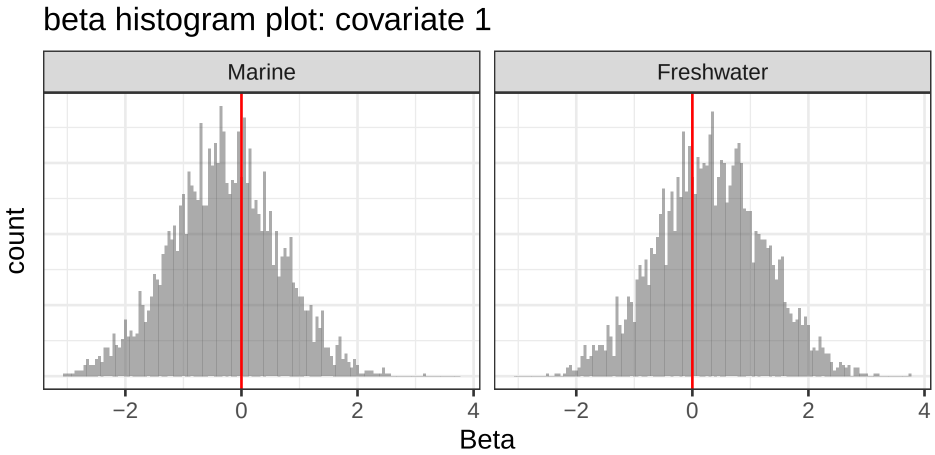
```
plot(cosimmr_alli_out,
     type = c("isospace", "beta_histogram", "p_ind"),
     ind = 1)
```

- Again specify the individual

Isospace Plot

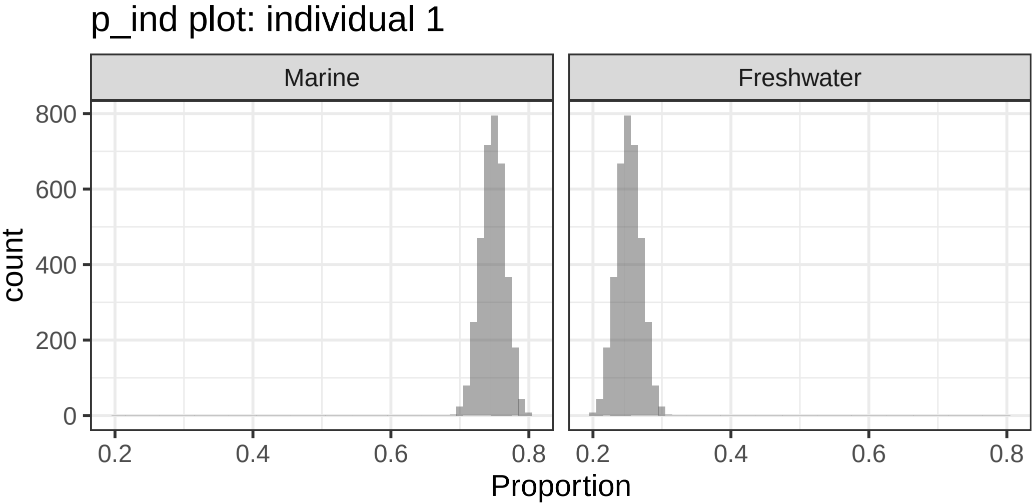


Beta Histogram



- This is useful for seeing what covariates are "important" to your model

Proportions Plot



- See estimates for food being consumed by an individual

Accessing outputs for your own plots

- Easily done!

```
cosimmr_alli_out$output$BUGSoutput$sims.list$p
```

Accessing outputs for your own plots

- Easily done!

```
cosimmr_alli_out$output$BUGSoutput$sims.list$p
```

- Will not be a matrix!

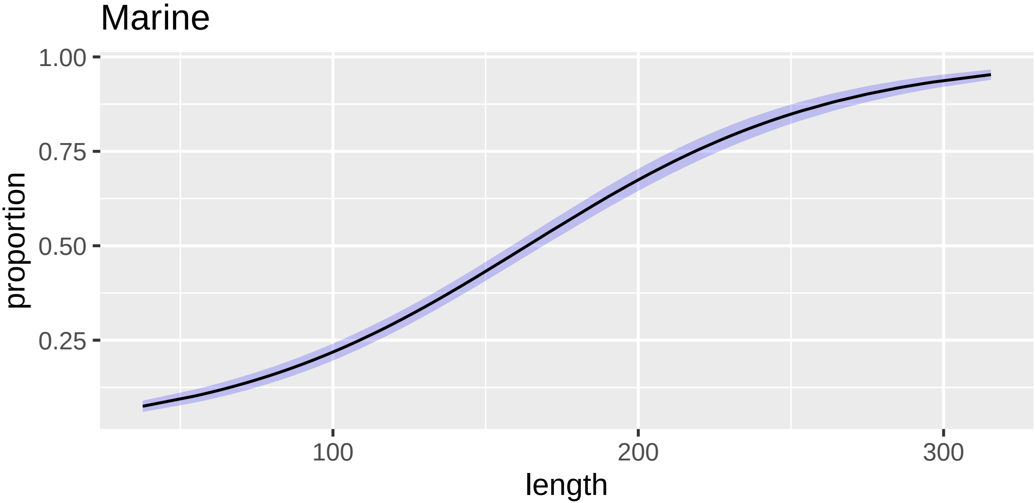
Accessing outputs for your own plots

- Easily done!

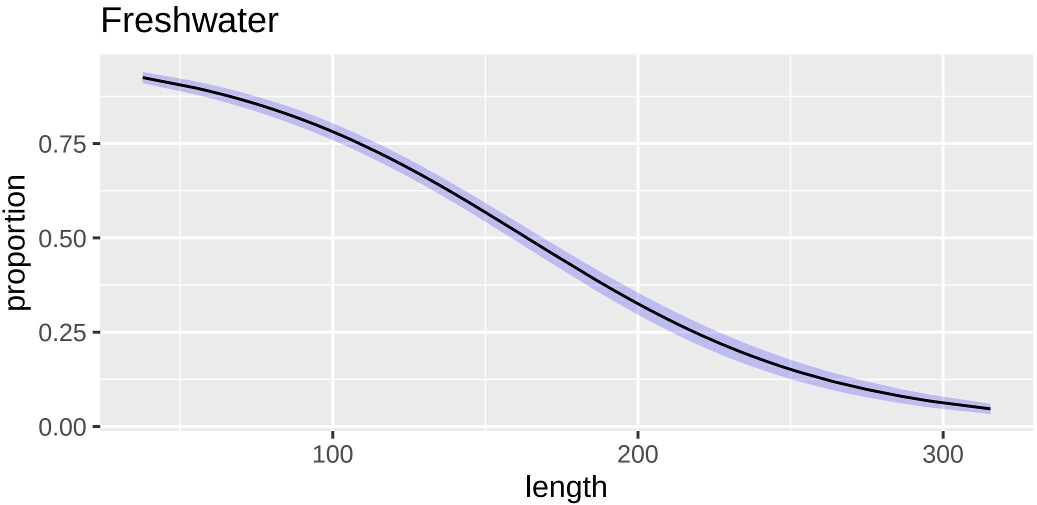
```
cosimmr_alli_out$output$BUGSoutput$sims.list$p
```

- Will not be a matrix!
- $\mathit{no.individuals} * \mathit{no.samples} * \mathit{no.isotopes}$

Proportions changing over covariate



Proportions changing over covariate



Predict function

```
x_pred = data.frame(length = c(100,210,302))
print(x_pred)
```

```
##   length
## 1    100
## 2    210
## 3    302
```

```
alli_pred = predict(cosimmr_alli_out, x_pred)
```

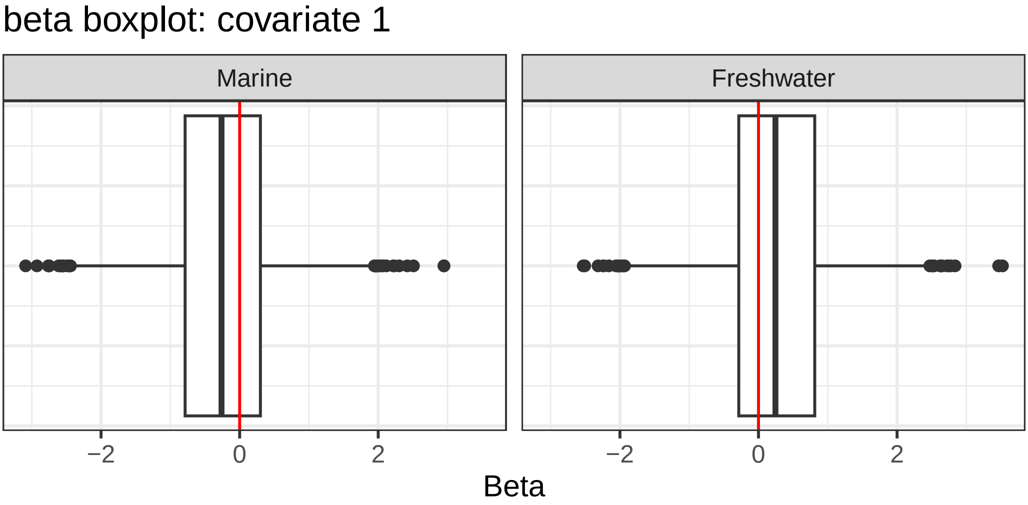
Summary

```
summary(alli_pred, ind = c(1,2,3), type = "statistics")
```

```
Summary for Individual 1

      mean    sd
[1,] 0.219 0.022
[2,] 0.781 0.022

Summary for Individual 2

      mean    sd
[1,] 0.717 0.029
[2,] 0.283 0.029

Summary for Individual 3

     mean    sd
[1,] 0.94 0.016
[2,] 0.06 0.016
```

Plot

```r
plot(alli_pred, type = "beta_boxplot")
```



Speed Comparison to MixSIAR

```
Unit: seconds
    expr       min       lq      mean    median        uq       max
 cosimmr  111.0862  120.658  125.8095  126.5247  132.2483  139.9875
    JAGS 1329.9057 1348.693 1383.0132 1378.8913 1409.9970 1449.0150
```

- Designed to be easy to use - don't need to know details of the algorithm working in the background!

Speed Comparison to MixSIAR

```
Unit: seconds
    expr       min       lq      mean    median        uq       max
 cosimmr  111.0862  120.658  125.8095  126.5247  132.2483  139.9875
    JAGS 1329.9057 1348.693 1383.0132 1378.8913 1409.9970 1449.0150
```

- Designed to be easy to use - don't need to know details of the algorithm working in the background!
- Quicker than MixSIAR

Summary

- We looked at the differences between simmr and cosimmr

Summary

- We looked at the differences between simmr and cosimmr
- We showed how to include random effects in a SIMM

Summary

- We looked at the differences between simmr and cosimmr
- We showed how to include random effects in a SIMM
- We learned how to run code in cosimmr

References

- Nifong, J. C., C. A. Layman, and B. R. Silliman. 2015. Size, sex and individual-level behaviour drive intrapopulation variation in cross-ecosystem foraging of a top-predator. Journal of Animal Ecology 84:35–48.
- Tran, Minh-Ngoc, Trong-Nghia Nguyen, and Viet-Hung Dao. "A practical tutorial on variational Bayes." arXiv preprint arXiv:2103.01327 (2021).
- Hulbert, Stuart H. "Pseudoreplication and the design of ecological field experiments." Ecological monographs 54.2 (1984): 187-211.

Any questions, bugs found, suggestions - please email me emmagovan@gmail.com or report at https://github.com/emmagovan/cosimmr/issues. Thanks!