Package ‘cosimmr’

May 8, 2026

Type Package

Title Fast Fitting of Stable Isotope Mixing Models with Covariates

Version 1.0.12

Date 2024-05-31

URL `https://github.com/emmagovan/cosimmr`, `https://emmagovan.github.io/cosimmr/`

Description Fast fitting of Stable Isotope Mixing Models in R. Allows for the inclusion of covariates. Also has built-in summary functions and plot functions which allow for the creation of isospace plots. Variational Bayes is used to fit these models, methods as described in: Tran et al., (2021) <doi:10.48550/arXiv.2103.01327>.

License GPL (>= 2)

Imports Rcpp (>= 1.0.10), R2jags, bayesplot, checkmate, ggplot2, viridis, reshape2, stats, ggnewscale

LinkingTo Rcpp, RcppArmadillo, RcppDist

Suggests testthat (>= 3.0.0), knitr, rmarkdown, readxl

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.1

Encoding UTF-8

Depends R (>= 3.5.0)

LazyData true

NeedsCompilation yes

Author Emma Govan [cre, aut],
Andrew Parnell [aut],
Ahmed Shalaby [ctb],
Alan Inglis [ctb]

Maintainer Emma Govan <`emmagovan@gmail.com`>

Repository CRAN

Date/Publication 2024-05-31 13:50:23 UTC

Contents

`alligator_data` *Alligator Data*

Description

Dataset from Nifong et al 2015 which contains 2 food sources, 181 individuals and 2 isotopes. This dataset includes multiple covariates as well as TDF means and sds.

Usage

```
alligator_data
```

Format

A list with the following elements

mixtures A two column matrix containing delta 13C and delta 15N values respectively

ID A character vector of unique ID values

tag_ID A character vector of tag ID

lat A numeric vector of latitude

long A numeric vector of longitude

date A character vector of date
year A numeric vector of year
habitat A character vector of habitat
sex A character vector of sex
length A numeric vector of length in cm
s_class A character vector of size class
sex_sclass A character vector for sex times size class
source_names A character vector of source names
source_means A data frame of source means for same tracers as in Mixtures
source_sds A data frame of standard deviations of sources for same tracers as in Mixtures
n_sources A numeric vector of number of sources
TEF_means A data frame of means for TEFs for same tracers as in Mixtures
TEF_sds A data frame of sds for TEFs for same tracers as in Mixtures

Source

<doi:10.1111/1365-2656.12306>

`cladocera_data` *Cladocera data from Galloway et al 2015.*

Description

Cladocera data from Galloway et al 2015. This dataset has 14 individuals on 7 food sources and 22 tracers. The id column can be used as a covariate. This dataset includes TDFs.

Usage

```
cladocera_data
```

Format

A list with the following elements

id numeric vector of ID number
group character vector of group ID
mixtures Data frame of tracer values. There are 22 fatty acids as tracers in this dataset.
tracer_names character vector of tracer names
source_means data frame of tracer means for each of the 7 food sources
source_sds data frame of tracer sds for each of the 7 food sources
n_sources numeric vector of the number of each food source collected
correction_means data frame with TDF means for each food source on each tracer
correction_sds data frame with TDF sds for each food source on each tracer

Source

<doi:10.1111/fwb.12394>

`cosimmr` *cosimmr: An R package for Stable Isotope Mixing Models*

Description

cosimmr is a package that has been developed to allow for running of Stable Isotope Mixing Models in R. It allows for the inclusion of covariates and has been designed to be easy to use for non-expert users. cosimmr uses Fixed Form Variational Bayes to run SIMMs, instead of MCMC. This allows for faster running of models without any issues with convergence

Author(s)

Emma Govan <emmagovan@gmail.com>, Andrew Parnell

`cosimmr_ffvb` *Run a `cosimmr_input` object through the Fixed Form Variational Bayes(FFVB) function*

Description

This is the main function of cosimmr. It takes a `cosimmr_input` object created via `cosimmr_load`, runs it in fixed form Variational Bayes to determine the dietary proportions, and then outputs a `cosimmr_output` object for further analysis and plotting via `plot.cosimmr_output`.

Usage

```
cosimmr_ffvb(
  cosimmr_in,
 prior_control = list(mu_0 = rep(0, (cosimmr_in$n_sources * cosimmr_in$n_covariates)),
   mu_log_sig_sq_0 = rep(0, cosimmr_in$n_tracers), sigma_0 = 1, tau_shape = rep(1,
    cosimmr_in$n_tracers), tau_rate = rep(1, cosimmr_in$n_tracers)),
 ffvb_control = list(n_output = 3600, S = 500, P = 50, beta_1 = 0.75, beta_2 = 0.75, tau
    = 500, eps_0 = 0.0011, t_W = 500)
)
```

Arguments

`cosimmr_in`	An object created via the function `cosimmr_load`
`prior_control`	A list of values including arguments named `mu_0` (prior for mu), and `sigma_0` (prior for sigma).
`ffvb_control`	A list of values including arguments named `n_output` (number of rows in theta output), `S` (number of samples taken at each iteration of the algorithm), `P` (patience parameter), `beta_1` and `beta_2` (adaptive learning weights), `tau` (threshold for exploring learning space), `eps_0` (fixed learning rate), `t_W` (rolling window size)

Value

An object of class `cosimmr_output` with two named top-level components:

`input`	The `cosimmr_input` object given to the `cosimmr_ffvb` function
`output`	A set of outputs produced by the FFVB function. These can be analysed using the `summary.cosimmr_output` and `plot.cosimmr_output` functions.

Author(s)

Emma Govan <emmagovan@gmail.com>, Andrew Parnell

References

Andrew C. Parnell, Donald L. Phillips, Stuart Bearhop, Brice X. Semmens, Eric J. Ward, Jonathan W. Moore, Andrew L. Jackson, Jonathan Grey, David J. Kelly, and Richard Inger. Bayesian stable isotope mixing models. Environmetrics, 24(6):387–399, 2013.

Andrew C Parnell, Richard Inger, Stuart Bearhop, and Andrew L Jackson. Source partitioning using stable isotopes: coping with too much variation. PLoS ONE, 5(3):5, 2010.

See Also

`cosimmr_load` for creating objects suitable for this function, `plot.cosimmr_input` for creating isospace plots, `summary.cosimmr_output` for summarising output, and `plot.cosimmr_output` for plotting output.

Examples

```
## See the package vignette for a detailed run through of these examples

# Data set 1: 10 obs on 2 isos, 4 sources, with tefs and concdep
data(geese_data_day1)
x = c(1,2,3,2,1,3,2,1,2)
cosimmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ x,
    source_names = source_names,
    source_means = source_means,
```

```
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)

# Plot
plot(cosimmr_1)

# Print
cosimmr_1

# FFVB run
cosimmr_1_out <- cosimmr_ffvb(cosimmr_1)

# Print it
print(cosimmr_1_out)

# Summary
summary(cosimmr_1_out, type = "correlations")
summary(cosimmr_1_out, type = "statistics")
ans <- summary(cosimmr_1_out, type = c("quantiles", "statistics"))

# Plot
plot(cosimmr_1_out, type = "beta_boxplot", cov_name = "x")
plot(cosimmr_1_out, type = "beta_histogram", cov_name = "x")
```

`cosimmr_load` *Function to load in cosimmr data and check for errors*

Description

This function takes in the mixture data, food source means and standard deviations, and (optionally) correction factor means and standard deviations, and concentration proportions. It performs some (non-exhaustive) checking of the data to make sure it will run through simmr. It outputs an object of class `cosimmr_input`.

Usage

```
cosimmr_load(
  formula,
  source_names,
  source_means,
  source_sds,
  correction_means = NULL,
  correction_sds = NULL,
```

```
  concentration_means = NULL,
  scale_x = TRUE
)
```

Arguments

`formula`	Formula giving in form y ~ x where y is a vector or matrix of mixture values and x is a vector or matrix of covariates
`source_names`	The names of the sources given as a character string
`source_means`	The means of the source values, given as a matrix where the number of rows is the number of sources and the number of columns is the number of tracers
`source_sds`	The standard deviations of the source values, given as a matrix where the number of rows is the number of sources and the number of columns is the number of tracers
`correction_means`	The means of the correction values, given as a matrix where the number of rows is the number of sources and the number of columns is the number of tracers. If not provided these are set to 0.
`correction_sds`	The standard deviations of the correction values, given as a matrix where the number of rows is the number of sources and the number of columns is the number of tracers. If not provided these are set to 0.
`concentration_means`	The means of the concentration values, given as a matrix where the number of rows is the number of sources and the number of columns is the number of tracers. These should be between 0 and 1. If not provided these are all set to 1.
`scale_x`	Whether or not you wish to scale the x values provided, or run the model using the original x values. Defaults to TRUE.

Details

For standard stable isotope mixture modelling, the mixture matrix will contain a row for each individual and a column for each isotopic value. `cosimmr` will allow for any number of isotopes and any number of observations, within computational limits. The source means/sds should be provided for each food source on each isotope. The correction means (usually trophic enrichment factors) can be set as zero if required, and should be of the same shape as the source values. The concentration dependence means should be estimated values of the proportion of each element in the food source in question and should be given in proportion format between 0 and 1. At present there is no means to include concentration standard deviations.

Value

An object of class `cosimmr_input` with the following elements:

`mixtures`	The mixture data
`source_names`	Source means
`sources_sds`	Source standard deviations

`correction_means`	Correction means
`correction_sds`	Correction standard deviations
`concentration_means`	Concentration dependence means
`n_obs`	The number of observations
`n_tracers`	The number of tracers/isotopes
`n_sources`	The number of sources
`n_groups`	The number of groups

Author(s)

Emma Govan <emmagovan@gmail.com>, Andrew Parnell

See Also

See `cosimmr_ffvb` for complete examples.

Examples

```
# A simple example with 10 observations, 2 tracers and 4 sources
data(geese_data_day1)
simmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ 1,
    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 = TRUE
  )
)



print(simmr_1)
```

`geese_data` *Geese stable isotope mixing data set*

Description

A real Geese data set with 251 observations on 2 isotopes, with 4 sources, and with corrections/trophic enrichment factors (TEFs or TDFs), and concentration dependence means. Taken from Inger et al (2016). See link for paper

Usage

```
geese_data
```

Format

A list with the following elements

mixtures A two column matrix containing delta 13C and delta 15N values respectively

source_names A character vector of the food source names

tracer_names A character vector of the tracer names (d13C, d15N, d34S)

source_means A matrix of source mean values for the tracers in the same order as `mixtures` above

source_sds A matrix of source sd values for the tracers in the same order as `mixtures` above

correction_means A matrix of TEFs mean values for the tracers in the same order as `mixtures` above

correction_sds A matrix of TEFs sd values for the tracers in the same order as `mixtures` above

concentration_means A matrix of concentration dependence mean values for the tracers in the same order as `mixtures` above

Source

<doi:10.1111/j.1365-2656.2006.01142.x>

`geese_data_day1` *A smaller version of the Geese stable isotope mixing data set*

Description

A real Geese data set with 9 observations on 2 isotopes, with 4 sources, and with corrections/trophic enrichment factors (TEFs or TDFs), and concentration dependence means. Taken from Inger et al (2016). See link for paper

Usage

```
geese_data_day1
```

Format

A list with the following elements

mixtures A two column matrix containing delta 13C and delta 15N values respectively

source_names A character vector of the food source names

tracer_names A character vector of the tracer names (d13C, d15N, d34S)

source_means A matrix of source mean values for the tracers in the same order as `mixtures` above

source_sds A matrix of source sd values for the tracers in the same order as `mixtures` above

correction_means A matrix of TEFs mean values for the tracers in the same order as `mixtures` above

correction_sds A matrix of TEFs sd values for the tracers in the same order as `mixtures` above

concentration_means A matrix of concentration dependence mean values for the tracers in the same order as `mixtures` above ...

Source

<doi:10.1111/j.1365-2656.2006.01142.x>

`iso_data` *Isopod Data*

Description

Isopod data from Galloway et al 2014. This dataset has 8 tracers (fatty acids), 30 individuals and 3 food sources. This dataset includes TDFs.

Usage

```
iso_data
```

Format

A list with the following elements

site A character vector with name of site for each individual

mixtures Data frame with 8 tracer values for 30 individuals

tracer_names character vector of tracer names

source_names character vector of food source names

source_means Data frame of source means with values for each food source on each tracer

source_sds Data frame of source sds with values for each food source on each tracer

n_sources numeric vector of number of each source obtained

correction_means Data frame of TDF means for each food source on each tracer

correction_sds Data frame of TDF sds for each food source on each tracer

Source

<doi:10.3354/meps10860>

`plot.cosimmr_input` *Plot the* `cosimmr_input` *data created from* `cosimmr_load`

Description

This function creates iso-space (AKA tracer-space or delta-space) plots. They are vital in determining whether the data are suitable for running in a SIMM.

Usage

```
## S3 method for class 'cosimmr_input'
plot(
  x,
  tracers = c(1, 2),
  title = "Tracers plot",
  xlab = colnames(x$mixtures)[tracers[1]],
  ylab = colnames(x$mixtures)[tracers[2]],
  sigmas = 1,
  mix_name = "Mixtures",
  colour = TRUE,
  colour_by_cov = FALSE,
  cov_name = NULL,
  ggargs = NULL,
  ...
)
```

Arguments

`x`	An object created via the function `cosimmr_load`
`tracers`	The choice of tracers to plot. If there are more than two tracers, it is recommended to plot every pair of tracers to determine whether the mixtures lie in the mixing polygon defined by the sources
`title`	A title for the graph
`xlab`	The x-axis label. By default this is assumed to be delta-13C but can be made richer if required. See examples below.
`ylab`	The y-axis label. By default this is assumed to be delta-15N in per mil but can be changed as with the x-axis label
`sigmas`	The number of standard deviations to plot on the source values. Defaults to 1.
`mix_name`	A optional string containing the name of the mixture objects, e.g. Geese.
`colour`	If TRUE (default) creates a plot. If not, puts the plot in black and white

`colour_by_cov`	if TRUE this allows users to colour the mixtures on the isospace plot by a specified covariate. Defaults to FALSE
`cov_name`	The name of the covariate the user wishes to colour the mixture points on the plot by
`ggargs`	Extra arguments to be included in the ggplot (e.g. axis limits)
`...`	Not used

Details

It is desirable to have the vast majority of the mixture observations to be inside the convex hull defined by the food sources. When there are more than two tracers (as in one of the examples below) it is recommended to plot all the different pairs of the food sources. See the vignette for further details of richer plots.

Value

isospace plot

Author(s)

Emma Govan <emmagovan@gmail.com>, Andrew Parnell

See Also

See `plot.cosimmr_output` for plotting the output of a simmr run. See `cosimmr_ffvb` for running a cosimmr object once the iso-space is deemed acceptable.

Examples

```
# A simple example with 10 observations, 4 food sources and 2 tracers
data(geese_data_day1)
cosimmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ c(1,2,3,2,3,1,2,3,1),
    source_names = source_names,
    source_means = source_means,
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)

# Plot
plot(cosimmr_1)

### A more complicated example with 30 obs, 3 tracers and 4 sources
data(simmr_data_2)
cosimmr_3 <- with(
```

```
    simmr_data_2,
    cosimmr_load(
      formula = mixtures ~ 1,
      source_names = source_names,
      source_means = source_means,
      source_sds = source_sds,
      correction_means = correction_means,
      correction_sds = correction_sds,
      concentration_means = concentration_means
    )
  )

# Plot 3 times - first default d13C vs d15N
plot(cosimmr_3)
# Now plot d15N vs d34S
plot(cosimmr_3, tracers = c(2, 3))
# and finally d13C vs d34S
plot(cosimmr_3, tracers = c(1, 3))
```

`plot.cosimmr_output` *Plot different features of an object created from* `cosimmr_ffvb`.

Description

This function allows for 4 different types of plots of the simmr output created from `cosimmr_ffvb`. The types are: plot of beta values

Usage

```
## S3 method for class 'cosimmr_output'
plot(
  x,
  type = c("isospace", "beta_histogram", "beta_boxplot", "prop_histogram",
    "prop_density", "covariates_plot"),
  obs = 1,
  cov_name = NULL,
  binwidth = 0.05,
  alpha = 0.5,
  title = NULL,
  n_output = 3600,
  source = NULL,
  one_plot = FALSE,
  n_pred = 1000,
  ...
)
```

Arguments

`x`	An object of class `cosimmr_output` created via `cosimmr_ffvb`.
`type`	The type of plot required. Can be one or more of 'isospace', 'beta_histogram', 'beta_boxplot', 'prob_histogram', 'prob_density', 'covariates_plot'
`obs`	The observation number you wish to plot
`cov_name`	The name of the covariate you wish to plot (for beta and covariates plot)
`binwidth`	The width of the bins for the histogram. Defaults to 0.05
`alpha`	The degree of transparency of the plots. Not relevant for matrix plots
`title`	The title of the plot.
`n_output`	The number of theta samples you wish to plot with. Defaults to 3600
`source`	The number or name of the source you wish to plot over for 'covariates_plot', defaults to NULL which means all sources are used
`one_plot`	Whether to plot line covariates plot on one plot. Defaults to FALSE
`n_pred`	Number of points to use when plotting line covariates plot. Defaults to 1000.
`...`	Currently not used

Details

The matrix plot should form a necessary part of any SIMM analysis since it allows the user to judge which sources are identifiable by the model. Further detail about these plots is provided in the vignette.

Value

one or more of 'isospace', 'beta_histogram', 'beta_boxplot', 'prop_histogram', 'prop_density', or 'covariates_plot'

Author(s)

Emma Govan <emmagovan@gmail.com>, Andrew Parnell

See Also

See `cosimmr_ffvb` for creating objects suitable for this function, and many more examples. See also `cosimmr_load` for creating simmr objects, `plot.cosimmr_input` for creating isospace plots.

Examples

```
# A simple example with 10 observations, 2 tracers and 4 sources

# The data
data(geese_data_day1)

# Load into simmr
simmr_1 <- with(
```

```
    geese_data_day1,
    cosimmr_load(
      formula = mixtures ~ 1,
      source_names = source_names,
      source_means = source_means,
      source_sds = source_sds,
      correction_means = correction_means,
      correction_sds = correction_sds,
      concentration_means = concentration_means
    )
  )
  # Plot
  plot(simmr_1)


  # FFVB run
  simmr_1_out <- cosimmr_ffvb(simmr_1)

  plot(simmr_1_out, type = c("isospace", "beta_hist"))
```

`plot.cosimmr_pred_out` *Plot different features of an object created from* `cosimmr_ffvb`.

Description

This function allows for 4 different types of plots of the simmr output created from `cosimmr_ffvb`. The types are: plot of beta values

Usage

```
## S3 method for class 'cosimmr_pred_out'
plot(
  x,
  type = c("beta_histogram", "beta_boxplot", "prop_obs", "prop_density"),
  obs = 1,
  cov_name = NULL,
  binwidth = 0.05,
  alpha = 0.5,
  title = NULL,
  n_output = 3600,
  ...
)
```

Arguments

`x`	An object of class `cosimmr_output` created via `cosimmr_ffvb`.
`type`	The type of plot required. Can be one or more of 'isospace', 'beta_histogram', 'beta_boxplot', 'prob_histogram', 'prob_density', 'covariates_plot'

`obs`	The observation you wish to plot
`cov_name`	The name of the covariate you wish to plot (for beta and covariate plots)
`binwidth`	The width of the bins for the histogram. Defaults to 0.05
`alpha`	The degree of transparency of the plots. Not relevant for matrix plots
`title`	The title of the plot.
`n_output`	The number of theta samples you wish to plot with. Defaults to 3600
`...`	Currently not used

Details

The matrix plot should form a necessary part of any SIMM analysis since it allows the user to judge which sources are identifiable by the model. Further detail about these plots is provided in the vignette.

Value

one or more of 'isospace', 'beta_histogram', 'beta_boxplot', 'prop_histogram', 'prop_density', or 'covariates_plot'

Author(s)

Emma Govan <emmagovan@gmail.com», Andrew Parnell

See Also

See `cosimmr_ffvb` for creating objects suitable for this function, and many more examples. See also `cosimmr_load` for creating simmr objects, `plot.cosimmr_input` for creating isospace plots.

Examples

```
# A simple example with 10 observations, 2 tracers and 4 sources

# The data
data(geese_data_day1)

# Load into simmr
cosimmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ 1,
    source_names = source_names,
    source_means = source_means,
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)
```

```
# Plot
plot(cosimmr_1)


# FFVB run
cosimmr_1_out <- cosimmr_ffvb(cosimmr_1)

plot(cosimmr_1_out, type = c("isospace", "beta_hist"))
```

`posterior_predictive` *Plot the posterior predictive distribution for a cosimmr run*

Description

This function takes the output from `cosimmr_ffvb` and plots the posterior predictive distribution to enable visualisation of model fit. The simulated posterior predicted values are returned as part of the object and can be saved for external use

Usage

```
posterior_predictive(
  cosimmr_out,
  prob = 0.5,
  plot_ppc = TRUE,
  n_samples = 3600,
  sort_data = TRUE
)
```

Arguments

`cosimmr_out`	A run of the cosimmr model from `cosimmr_ffvb`.
`prob`	The probability interval for the posterior predictives. The default is 0.5 (i.e. 50pc intervals)
`plot_ppc`	Whether to create a bayesplot of the posterior predictive or not.
`n_samples`	The number of samples you wish to generate for y_pred. Defaults to 3600.
`sort_data`	Whether to order the data from lowest to highest predicted mean or not. Defaults to TRUE.

Value

plot of posterior predictives and simulated values

#' @author Emma Govan <emmagovan@gmail.com> Andrew Parnell

See Also

`cosimmr_ffvb` for creating objects suitable for this function

Examples

```
data(geese_data_day1)
cosimmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ c(1,2,3,2,1,2,3,2,1),
    source_names = source_names,
    source_means = source_means,
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)

# Plot
plot(cosimmr_1)

# Print
cosimmr_1

# FFVB run
cosimmr_1_out <- cosimmr_ffvb(cosimmr_1)

# Prior predictive
post_pred <- posterior_predictive(cosimmr_1_out)
```

`predict.cosimmr_output` *Predicts proportion of each source in a mixture, based on values provided for covariates*

Description

Predicts proportion of each source in a mixture, based on values provided for covariates

Usage

```
## S3 method for class 'cosimmr_output'
predict(object, x_pred, n_output = 3600, ...)
```

Arguments

`object`	An object of class `cosimmr_output` created via the function `cosimmr_ffvb`
`x_pred`	A data.frame of covariate values that the user wishes to predict source proportions for, provided in the same order that the original covariance matrix was. Important for this to be a data.frame otherwise numeric values can be set as characters and this causes incorrect calculations.

`n_output`	the number of posterior samples to generate. Defaults to 3600.
`...`	Other arguments (not used)

Value

object of class 'cosimmr_pred_out'

Author(s)

Emma Govan <emmagovan@gmail.com> Andrew Parnell

References

Andrew C. Parnell, Donald L. Phillips, Stuart Bearhop, Brice X. Semmens, Eric J. Ward, Jonathan W. Moore, Andrew L. Jackson, Jonathan Grey, David J. Kelly, and Richard Inger. Bayesian stable isotope mixing models. Environmetrics, 24(6):387–399, 2013.

See Also

`cosimmr_load` for creating objects suitable for this function, and `plot.cosimmr_output` for plotting output.

Examples

```
## See the package vignette for a detailed run through of these 4 examples

# Data set 1: 10 obs on 2 isos, 4 sources, with tefs and concdep
data(geese_data_day1)
cov_1 = c(1,2,3,2,3,1,1,1,2)
simmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ cov_1,
    source_names = source_names,
    source_means = source_means,
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)

# Plot
plot(simmr_1)

# Print
simmr_1

# FFVB run
simmr_1_out <- cosimmr_ffvb(simmr_1)

# Print it
```

```
print(simmr_1_out)


# Plot
plot(simmr_1_out, type = "isospace")
plot(simmr_1_out, type = "beta_histogram", cov_name = "cov_1")

x_pred = data.frame(cov_1 = c(1,5))

pred_array<-predict(simmr_1_out, x_pred)
```

`print.cosimmr_input` *Print simmr input object*

Description

Print simmr input object

Usage

```
## S3 method for class 'cosimmr_input'
print(x, ...)
```

Arguments

`x`	An object of class `cosimmr_input`
`...`	Other arguments (not supported) #' @author Emma Govan <emmagovan@gmail.com> Andrew Parnell

Value

A neat presentation of your simmr object.

See Also

`cosimmr_load` for creating objects suitable for this function

`print.cosimmr_output` *Print a simmr output object*

Description

Print a simmr output object

Usage

```
## S3 method for class 'cosimmr_output'
print(x, ...)
```

Arguments

`x`	An object of class `cosimmr_output`
`...`	Other arguments (not supported)

Value

Returns a neat summary of the object

See Also

`cosimmr_ffvb` for creating `cosimmr_output` objects

`prior_viz` *Plot the prior distribution for a cosimmr run*

Description

This function takes the output from `cosimmr_ffvb` and plots the prior distribution to enable visual inspection. This can be used by itself or together with `posterior_predictive` to visually evaluate the influence of the prior on the posterior distribution.

Usage

```
prior_viz(
  cosimmr_out,
  plot = TRUE,
  include_posterior = TRUE,
  n_sims = 10000,
  scales = "free"
)
```

Arguments

`cosimmr_out`	A run of the cosimmr model from `cosimmr_ffvb`
`plot`	Whether to create a density plot of the prior or not. The simulated prior values are returned as part of the object
`include_posterior`	Whether to include the posterior distribution on top of the priors. Defaults to TRUE. The posterior returned is of the mean value of covariates
`n_sims`	The number of simulations from the prior distribution
`scales`	The type of scale from `facet_wrap` allowing for `fixed`, `free`, `free_x`, `free_y`

Value

A list containing `plot`: the ggplot object (useful if requires customisation), and `sim`: the simulated prior values which can be compared with the posterior densities

#' @author Emma Govan <emmagovan@gmail.com> Andrew Parnell

See Also

`cosimmr_ffvb` for creating objects suitable for this function

Examples

```
data(geese_data_day1)
cosimmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ 1,
    source_names = source_names,
    source_means = source_means,
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)

# Plot
plot(cosimmr_1)

# Print
cosimmr_1

# FFVB run
cosimmr_1_out <- cosimmr_ffvb(cosimmr_1)

# Prior predictive
prior <- prior_viz(cosimmr_1_out)
head(prior$p_prior_sim)
summary(prior$p_prior_sim)
```

`simmr_data_1` *A simple fake stable isotope mixing data set*

Description

A simple fake data set with 10 observations on 2 isotopes, with 4 sources, and with corrections/trophic enrichment factors (TEFs or TDFs), and concentration dependence means

Usage

```
simmr_data_1
```

Format

A list with the following elements

- **mixtures** A two column matrix containing delta 13C and delta 15N values respectively
- **source_names** A character vector of the food source names
- **tracer_names** A character vector of the tracer names (d13C and d15N)
- **source_means** A matrix of source mean values for the tracers in the same order as `mixtures` above
- **source_sds** A matrix of source sd values for the tracers in the same order as `mixtures` above
- **correction_means** A matrix of TEFs mean values for the tracers in the same order as `mixtures` above
- **correction_sds** A matrix of TEFs sd values for the tracers in the same order as `mixtures` above
- **concentration_means** A matrix of concentration dependence mean values for the tracers in the same order as `mixtures` above

---

`simmr_data_2` *A 3-isotope fake stable isotope mixing data set*

Description

A fake data set with 30 observations on 3 isotopes, with 4 sources, and with corrections/trophic enrichment factors (TEFs or TDFs), and concentration dependence means

Usage

```
simmr_data_2
```

Format

A list with the following elements

mixtures A three column matrix containing delta 13C, delta 15N, and delta 34S values respectively

source_names A character vector of the food source names

tracer_names A character vector of the tracer names (d13C, d15N, d34S)

source_means A matrix of source mean values for the tracers in the same order as `mixtures` above

source_sds A matrix of source sd values for the tracers in the same order as `mixtures` above

correction_means A matrix of TEFs mean values for the tracers in the same order as `mixtures` above

correction_sds A matrix of TEFs sd values for the tracers in the same order as `mixtures` above

concentration_means A matrix of concentration dependence mean values for the tracers in the same order as `mixtures` above

---

`square_data` *An artificial data set used to indicate effect of priors*

Description

A fake box data set identified by Fry (2014) as a failing of SIMMs See the link for more interpretation of these data and the output

Usage

```
square_data
```

Format

A list with the following elements

mixtures A two column matrix containing delta 13C and delta 15N values respectively

source_names A character vector of the food source names

tracer_names A character vector of the tracer names (d13C, d15N)

source_means A matrix of source mean values for the tracers in the same order as `mixtures` above

source_sds A matrix of source sd values for the tracers in the same order as `mixtures` above

correction_means A matrix of TEFs mean values for the tracers in the same order as `mixtures` above

correction_sds A matrix of TEFs sd values for the tracers in the same order as `mixtures` above

concentration_means A matrix of concentration dependence mean values for the tracers in the same order as `mixtures` above

Source

<doi:10.3354/meps10535>

`summary.cosimmr_output`

Summarises the output created with `cosimmr_ffvb`

Description

Produces textual summaries and convergence diagnostics for an object created with `cosimmr_ffvb`. The different options are: 'quantiles' which produces credible intervals for the parameters, 'statistics' which produces means and standard deviations, and 'correlations' which produces correlations between the parameters.

Usage

```
## S3 method for class 'cosimmr_output'
summary(
  object,
  type = c("quantiles", "statistics", "correlations"),
  obs = 1,
  ...
)
```

Arguments

`object`	An object of class `cosimmr_output` produced by the function `cosimmr_ffvb`
`type`	The type of output required. At least none of quantiles', 'statistics', or 'correlations'.
`obs`	The observation to generate a summary for. Defaults to 1.
`...`	Not used

Details

The quantile output allows easy calculation of 95 per cent credible intervals of the posterior dietary proportions. The correlations allow the user to judge which sources are non-identifiable.

Value

A list containing the following components:

`quantiles`	The quantiles of each parameter from the posterior distribution
`statistics`	The means and standard deviations of each parameter
`correlations`	The posterior correlations between the parameters

Note that this object is reported silently so will be discarded unless the function is called with an object as in the example below.

Author(s)

Emma Govan <emmagovan@gmail.com> Andrew Parnell

See Also

See `cosimmr_ffvb`for creating objects suitable for this function, and many more examples. See also `cosimmr_load` for creating cosimmr objects, `plot.cosimmr_input` for creating isospace plots, `plot.cosimmr_output` for plotting output.

Examples

```
# A simple example with 10 observations, 2 tracers and 4 sources

# The data
data(geese_data_day1)
cosimmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ 1,
    source_names = source_names,
    source_means = source_means,
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)

# Plot
plot(cosimmr_1)


# FFVB run
cosimmr_1_out <- cosimmr_ffvb(cosimmr_1)

# Summarise
summary(cosimmr_1_out) # This outputs all the summaries
summary(cosimmr_1_out, type = "quantiles") # Just the diagnostics
# Store the output in an
ans <- summary(cosimmr_1_out,
  type = c("quantiles", "statistics")
)
```

`summary.cosimmr_pred_out` *Summarises the output created with* `cosimmr_ffvb`

Description

Produces textual summaries and convergence diagnostics for an object created with `cosimmr_ffvb`. The different options are: 'quantiles' which produces credible intervals for the parameters, 'statistics' which produces means and standard deviations, and 'correlations' which produces correlations between the parameters.

Usage

```
## S3 method for class 'cosimmr_pred_out'
summary(
  object,
  type = c("quantiles", "statistics", "correlations"),
  obs = 1,
  ...
)
```

Arguments

`object`	An object of class `cosimmr_pred_output` produced by the function `predict.cosimmr_output`
`type`	The type of output required. At least none of quantiles', 'statistics', or 'correlations'.
`obs`	The observation to generate a summary for. Defaults to 1.
`...`	Not used

Details

The quantile output allows easy calculation of 95 per cent credible intervals of the posterior dietary proportions. The correlations allow the user to judge which sources are non-identifiable.

Value

A list containing the following components:

`quantiles`	The quantiles of each parameter from the posterior distribution
`statistics`	The means and standard deviations of each parameter
`correlations`	The posterior correlations between the parameters

Note that this object is reported silently so will be discarded unless the function is called with an object as in the example below.

Author(s)

Emma Govan <emmagovan@gmail.com> Andrew Parnell

See Also

See `cosimmr_ffvb`for creating objects suitable for this function, and many more examples. See also `cosimmr_load` for creating cosimmr objects, `plot.cosimmr_input` for creating isospace plots, `plot.cosimmr_output` for plotting output.

Examples

```
# A simple example with 10 observations, 2 tracers and 4 sources

# The data
data(geese_data_day1)
cosimmr_1 <- with(
  geese_data_day1,
  cosimmr_load(
    formula = mixtures ~ c(1,2,3,3,2,3,1,2,1),
    source_names = source_names,
    source_means = source_means,
    source_sds = source_sds,
    correction_means = correction_means,
    correction_sds = correction_sds,
    concentration_means = concentration_means
  )
)

# Plot
plot(cosimmr_1)


# FFVB run
cosimmr_1_out <- cosimmr_ffvb(cosimmr_1)

# Summarise
summary(cosimmr_1_out) # This outputs all the summaries
summary(cosimmr_1_out, type = "quantiles") # Just the diagnostics
# Store the output in ans
ans <- summary(cosimmr_1_out,
  type = c("quantiles", "statistics")
)
```

Index