

# Package ‘MCMCvis’

May 2, 2018

**Type** Package

**Title** Tools to Visualize, Manipulate, and Summarize MCMC Output

**Version** 0.10.2

**Maintainer** Casey Youngflesh <caseyyoungflesh@gmail.com>

**Description** Performs key functions for MCMC analysis using minimal code - visualizes, manipulates, and summarizes MCMC output. Functions support simple and straightforward subsetting of model parameters within the calls, and produce presentable and 'publication-ready' output. MCMC output may be derived from Bayesian model output fit with JAGS, Stan, or other MCMC samplers.

**License** GPL-3

**URL** <http://github.com/caseyyoungflesh/MCMCvis>

**BugReports** <http://github.com/caseyyoungflesh/MCMCvis/issues>

**LazyData** TRUE

**Imports** coda, rstan, grDevices, graphics, stats, overlapping

**Depends** R (>= 3.3.0)

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Casey Youngflesh [aut, cre] (<<https://orcid.org/0000-0001-6343-3311>>)

**Repository** CRAN

**Date/Publication** 2018-05-02 15:20:23 UTC

## R topics documented:

MCMCchains	2
MCMCplot	3
MCMCpstr	6
MCMCsummary	7

MCMCtrace . . . . .	9
MCMCvis . . . . .	11
MCMC_data . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

MCMCchains	<i>Extract posterior chains from MCMC output</i>
------------	--

---

## Description

Extract posterior chains from MCMC output for specific parameters of interest.

## Usage

```
MCMCchains(object, params = "all", excl = NULL, ISB = TRUE,
            mcmc.list = FALSE)
```

## Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters of interest. Default 'all' returns chains for all parameters.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored - input from params and excl are otherwise matched exactly. If FALSE, square brackets are not ignored - input from params and excl are matched using regular expression format. This allows partial names to be used when specifying parameters of interest.
mcmc.list	Logical specifying whether to return an mcmc.list. If TRUE, an mcmc.list object is returned, rather than a matrix.

## Details

Function returns matrix with one chain per column for specified parameters. Multiple input chains for each parameter are combined to one posterior chain. Parameters are arranged in columns alphabetically.

object argument can be a stanfit object (rstan package), an mcmc.list object (coda package), an R2jags model object (R2jags package), a jagsUI model object (jagsUI package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

**Examples**

```

#Load data
data(MCMC_data)

#Extract MCMC chains
ex <- MCMCchains(MCMC_data)
apply(ex, 2, mean)

#Extract MCMC chains for just 'beta' parameters
ex2 <- MCMCchains(MCMC_data, params = 'beta')
apply(ex2, 2, mean)

#Just 'beta[1]', 'beta[4]', and 'alpha[3]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\\
ex3 <- MCMCchains(MCMC_data, params = c('beta\\[1\\]', 'beta\\[4\\]', 'alpha\\[3\\]'), ISB = FALSE)
apply(ex3, 2, sd)

```

MCMCplot

*Caterpillar plots of posterior distributions from MCMC output***Description**

Visualize posterior distributions from MCMC output for specific parameters of interest using caterpillar plots. Color of median dot represents the overlap of the posterior distribution with 0 (or other specified value).

**Usage**

```

MCMCplot(object, params = "all", excl = NULL, ISB = TRUE, ref = 0,
  ref_ovl = FALSE, col = "black", rank = FALSE, horiz = TRUE, xlim,
  ylim, xlab, ylab, main, labels, labels_sz = 1.2, med_sz = 1.5,
  thick_sz = 5, thin_sz = 2, ax_sz = 3, axis_text_sz = 1.3,
  tick_text_sz = 1.2, main_text_sz = 1.2, tick_pos, mar = c(5.1, 4.1, 4.1,
  2.1))

```

**Arguments**

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters to be plotted. Default 'all' plots posteriors for all parameters. See VALUE below.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.

ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the <code>params</code> and <code>excl</code> arguments. If TRUE, square brackets are ignored - input from <code>params</code> and <code>excl</code> are otherwise matched exactly. If FALSE, square brackets are not ignored - input from <code>params</code> and <code>excl</code> are matched using <code>grep</code> , which can take arguments in regular expression format. This allows partial names to be used when specifying parameters of interest.
ref	Value indicating where vertical reference line should be created and what value to use a reference for caterpillar median coloration. Default is <code>ref = 0</code> . Argument NULL will plot no reference line.
ref_ovl	Logical specifying whether the style/color of plotted median dots and CI should be changed based on whether the 50 % and 95 % credible intervals overlap the reference line. See DETAILS for more information.
col	Character string specifying which color to render estimates on plot. When <code>ref_ovl = TRUE</code> , this argument has no effect and colors plotted will be based on the credible intervals and reference line.
rank	Logical specifying whether output should be ranked. If TRUE posteriors will be ranked in decreasing order (based on specified measure of centrality) from top down.
horiz	Logical specifying orientation of plot. If TRUE posteriors will be plotted running horizontally (parallel to the x-axis). If FALSE posteriors will be plotted running vertically (perpendicular to the x-axis).
xlim	Numerical vector of length 2, indicating range of x-axis. Only applicable if <code>horiz = TRUE</code> .
ylim	Numerical vector of length 2, indicating range of y-axis. Only applicable if <code>horiz = FALSE</code> .
xlab	Character string labeling x-axis. Only applicable if <code>horiz = TRUE</code> . Default label is 'Parameter Estimate'. Option NULL will return plot with no label on x-axis.
ylab	Character string labeling y-axis. Only applicable if <code>horiz = FALSE</code> . Default label is 'Parameter Estimate'. Option NULL will return plot with no label on y-axis.
main	Character string indicating title of plot.
labels	Character string (or vector of character strings if plotting > 1 parameter) labeling parameter estimates along y-axis (if <code>horiz = FALSE</code> ) or x-axis (if <code>horiz = TRUE</code> ). Default option will use parameter names from object. Option NULL will return plot with no labels on axis.
labels_sz	Number specifying size of text for parameter labels on axis.
med_sz	Number specifying size of points represents posterior medians.
thick_sz	Number specifying thickness of 50 percent CI line (thicker line).
thin_sz	Number specifying thickness of 95 percent CI line (thinner line).
ax_sz	Number specifying thickness of axis and ticks.

<code>axis_text_sz</code>	Number specifying size of text for axis label.
<code>tick_text_sz</code>	Number specifying size of text for tick labels on axis.
<code>main_text_sz</code>	Number specifying size of text for main title.
<code>tick_pos</code>	Numeric vector specifying where ticks on axis should be placed.
<code>mar</code>	Numerical vector of length 4 specifying plot margins - (BOTTOM, LEFT, TOP, RIGHT). Changes to the margin should be made within the function rather than using the <code>par</code> call. Default is <code>c(5.1, 4.1, 4.1, 2.1)</code> - the R plot default.

### Details

Points represent posterior medians. Parameters where 50% credible intervals overlap 0 (or other specified value) are indicated by 'open' circles. Parameters where 50 percent credible intervals DO NOT overlap 0 AND 95 percent credible intervals DO overlap 0 (or other specified value) are indicated by 'closed' gray circles. Parameters where 95 percent credible intervals DO NOT overlap 0 (or other specified value) are indicated by 'closed' black circles. Thick lines represent 50 percent credible intervals while thin lines represent 95 % credible intervals. `ref_ov1 = TRUE` can be used to enable this feature.

object argument can be a `stanfit` object (rstan package), an `mcmc.list` object (coda package), an `R2jags` model object (R2jags package), a `jagsUI` model object (jagsUI package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

### Notes

When specifying `rank = TRUE` and specifying labels for labels, labels will be applied to parameters before they are ranked.

Thanks to Cinner et al. 2016, whose Fig. 1 inspired this plot.

### References

Cinner, J. E., C. Huchery, M. A. MacNeil, N. A. J. Graham, T. R. McClanahan, J. Maina, E. Maire, J. N. Kittinger, C. C. Hicks, C. Mora, E. H. Allison, S. D'Agata, A. Hoey, D. A. Feary, L. Crowder, I. D. Williams, M. Kulbicki, L. Vigliola, L. Wantiez, G. Edgar, R. D. Stuart-Smith, S. A. Sandin, A. L. Green, M. J. Hardt, M. Beger, A. Friedlander, S. J. Campbell, K. E. Holmes, S. K. Wilson, E. Brokovich, A. J. Brooks, J. J. Cruz-Motta, D. J. Booth, P. Chabanet, C. Gough, M. Tupper, S. C. A. Ferse, U. R. Sumaila, and D. Mouillot. 2016. Bright spots among the world's coral reefs. *Nature* 535:416-419.

### Examples

```
#Load data
data(MCMC_data)

#Plot MCMC output
MCMCplot(MCMC_data, labels = NULL)
```

```

#Just 'beta' parameters
MCMCplot(MCMC_data, params = 'beta')

#Just 'beta[1]', 'beta[4]', and 'alpha[3]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\\
MCMCplot(MCMC_data, params = c('beta\\[1\\]', 'beta\\[4\\]', 'alpha\\[3\\]'), ISB = FALSE)

#Rank parameters by posterior mean
MCMCplot(MCMC_data, params = 'beta', rank = TRUE)

#Create vertical plot
MCMCplot(MCMC_data, params = 'beta', horiz = FALSE)

```

---

MCMCpstr	<i>Summary function for MCMC output that preserves parameter structure</i>
----------	--

---

## Description

Extract summary information from MCMC output (specific function specified) for specific parameters of interest while preserving original parameter structure (i.e., scalar, vector, matrix, array). Function outputs a list with calculated values for each specified parameter.

## Usage

```
MCMCpstr(object, params = "all", excl = NULL, ISB = TRUE, digits = 2,
         func = mean)
```

## Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters to be returned in summary output. Default 'all' returns all parameters in summary output.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored - input from params and excl are otherwise matched exactly. If FALSE, square brackets are not ignored - input from params and excl are matched using grep, which can take arguments in regular expression format. This allows partial names to be used when specifying parameters of interest.
digits	Number of digits to include for posterior summary. Values will be rounded to the specified number of digits (except for Rhat which is always rounded to 2 digits). Default is digits = 2.

func            Function to be performed on MCMC output. Output of specified function must be of length 1.

### Details

object argument can be a stanfit object (rstan package), an mcmc.list object (coda package), an R2jags model object (R2jags package), a jagsUI model object (jagsUI package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

### Examples

```
#Load data
data(MCMC_data)

MCMCpstr(MCMC_data, func = function(x) quantile(x, probs = 0.01))
```

---

MCMCsummary

*Summary function for MCMC output*


---

### Description

Extract summary information from MCMC output (mean, median, quantiles, Gelman-Rubin convergence statistic, number of effective samples, and specified custom metrics) for specific parameters of interest.

### Usage

```
MCMCsummary(object, params = "all", excl = NULL, ISB = TRUE, digits = 2,
             Rhat = TRUE, n.eff = FALSE, func = NULL, func_name = NULL)
```

### Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters to be returned in summary output. Default 'all' returns all parameters in summary output.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored - input from params and excl are otherwise matched exactly. If FALSE, square brackets are not ignored - input from params and excl are matched using grep, which can take arguments in regular expression format. This allows partial names to be used when specifying parameters of interest.

<code>digits</code>	Number of digits to include for posterior summary. Values will be rounded to the specified number of digits (except for <code>Rhat</code> which is always rounded to 2 digits). Default is <code>digits = 2</code> .
<code>Rhat</code>	Logical specifying whether to calculate and display the Gelman-Rubin convergence statistic ( <code>Rhat</code> ). Values near 1 suggest convergence (Brooks and Gelman 1998). <code>Rhat = FALSE</code> will prevent display of this column in summary output. Specifying <code>Rhat = FALSE</code> , will increase function speed, particularly with very large <code>'mcmc.list'</code> objects.
<code>n.eff</code>	Logical specifying whether to calculate and display the number of effective samples for each parameter. Kruschke (2014) recommends <code>n.eff &gt; 10,000</code> for reasonably stable posterior estimates from M-H/Gibbs sampling derived output. <code>n.eff = FALSE</code> will prevent display of this column in summary output. Specifying <code>n.eff = FALSE</code> , will increase function speed, particularly with very large <code>'mcmc.list'</code> objects.
<code>func</code>	Function to be performed on MCMC output. If a function is specified, it will be evaluated on posteriors for each specified parameter and returned as a column in the summary output (or multiple columns if the function returns more than one value).
<code>func_name</code>	Character string (or vector of character strings) specifying labels for output from <code>func</code> argument. If <code>func_name</code> is not specified, columns with <code>func</code> argument will be labeled <code>'func'</code> .

### Value

Function returns summary information (including parameter posterior mean, posterior sd, 2.5% quantile, median, 97.5% quantile, Gelman-Rubin convergence statistic (`Rhat`), number of effective samples, and other specified metrics) for specified parameters.

### Details

object argument can be a `stanfit` object (`rstan` package), an `mcmc.list` object (`coda` package), an `R2jags` model object (`R2jags` package), a `jagsUI` model object (`jagsUI` package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

### Notes

For `mcmc.list` objects, Gelman-Rubin convergence statistic (`Rhat`) is calculated using the `gelman.diag` function in the `coda` package.

For `mcmc.list` objects, the number of effective samples is calculated using the `effectiveSize` function in the `coda` package.

### References

Brooks, S. P., and A. Gelman. 1998. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics* 7:434.

Kruschke, J. 2014. Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan. Academic Press.

### Examples

```
#Load data
data(MCMC_data)

#Summary information for MCMC output
MCMCsummary(MCMC_data)

#Just 'beta' parameters
MCMCsummary(MCMC_data, params = 'beta')

#Just 'beta[1]', 'beta[4]', and 'alpha[3]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\\
MCMCsummary(MCMC_data, params = c('beta\\[1\\]', 'beta\\[4\\]', 'alpha\\[3\\]'), ISB = FALSE)
```

---

MCMCtrace

*Trace and density plots from MCMC output*


---

### Description

Trace and density plots of MCMC chains for specific parameters of interest. Print plots to pdf by default.

### Usage

```
MCMCtrace(object, params = "all", excl = NULL, ISB = TRUE, iter = 5000,
  gvals = NULL, priors = NULL, post_zm = TRUE, PPO_out = FALSE,
  pdf = TRUE, open_pdf = TRUE, filename, wd = getwd(), type = "both",
  ind = FALSE)
```

### Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters of interest. Default 'all' returns chains for all parameters.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored - input from params and excl are otherwise matched exactly. If FALSE, square brackets are not ignored - input from params and excl are matched using grep, which can take arguments in regular expression format. This allows partial names to be used when specifying parameters of interest.

<code>iter</code>	Number of iterations to plot for trace and density plots. The default value is 5000, meaning the last 5000 iterations of the chain will be plotted.
<code>gvals</code>	Vector containing generating values if simulated data was used to fit model. These values will be plotted as vertical lines on the density plots to compare posterior distributions with the true parameter values used to generate the data. No line will be apparent if the generating value is outside the plotted range of the posterior distribution.
<code>priors</code>	Matrix containing random draws from prior distributions corresponding to parameters of interest. If specified, priors are plotted along with posterior density plots. Percent overlap between prior and posterior (PPO) is also calculated and displayed on each plot. Each column of the matrix represents a prior for a different parameter. Parameters are plotted alphabetically - priors should be sorted accordingly. If <code>priors</code> contains only one prior and more than one parameter is specified for the <code>params</code> argument, this prior will be used for all parameters. The number of draws for each prior should equal the number of iterations specified by <code>iter</code> (or total draws if less than <code>iter</code> ) times the number of chains, though the function will automatically adjust if more or fewer iterations are specified. See DETAILS below.
<code>post_zm</code>	Logical - if <code>post_zm = FALSE</code> x- and y-limits of density plots are scaled so that both the prior and posterior can be visualized on a single density plot (rather than zoomed on the posterior).
<code>PPO_out</code>	Logical - if <code>PPO_out = TRUE</code> percent overlap between prior and posterior (PPO) will be output to a dataframe.
<code>pdf</code>	Logical - if <code>pdf = TRUE</code> plots will be exported to a pdf.
<code>open_pdf</code>	Logical - if <code>open_pdf = TRUE</code> pdf will open in viewer after being generated.
<code>filename</code>	Name of pdf file to be printed. Default is 'MCMCtrace'.
<code>wd</code>	Working directory for pdf output. Default is current directory.
<code>type</code>	Type of plot to be output. 'both' outputs both trace and density plots, 'trace' outputs only trace plots, and 'density' outputs only density plots.
<code>ind</code>	Logical - if <code>ind = TRUE</code> , separate density lines will be plotted for each chain. If <code>ind= FALSE</code> , one density line will be plotted for all chains.

### Details

object argument can be a `stanfit` object (`rstan` package), an `mcmc.list` object (`coda` package), an `R2jags` model object (`R2jags` package), a `jagsUI` model object (`jagsUI` package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

Matrices for the `priors` argument can be generated using commands such as `rnorm`, `rgamma`, `runif`, etc. Distributions not supported by base R can be generated by using the appropriate packages. It is important to note that some discrepancies between MCMC samplers and R may exist regarding the parameterization of distributions - one example of this is the use of precision in JAGS but standard deviation in R for the 'second parameter' of the normal distribution. If the number of draws for each prior distribution is greater than the total number used for the density plot (`iter` times the number

of chains), the function will use a subset of the prior draws. If the number of draws for each prior distribution is less than the total number used for the density plot, the function will resample (with replacement) from the prior to obtain the appropriate number of draws.

### Examples

```
#Load data
data(MCMC_data)

#Traceplots for all 'beta' parameters - a pdf of the traceplots is generated by default
MCMCtrace(MCMC_data, params = 'beta', pdf = FALSE)

#Traceplots (individual density lines for each chain) just for 'beta[1]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\\
MCMCtrace(MCMC_data, params = 'beta\\[1\\]', ISB = FALSE, ind = TRUE, pdf = FALSE)

#Plot prior on top of posterior and calculate prior/posterior overlap (PPO) just for 'beta[1]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\\
PR <- rnorm(15000, 0, 32)
MCMCtrace(MCMC_data, params = 'beta\\[1\\]', ISB = FALSE, priors = PR, pdf = FALSE)
```

---

MCMCvis

*The 'MCMCvis' package*


---

### Description

'MCMCvis' is an R package used to visualize, manipulate, and summarize MCMC output. MCMC output may be derived from Bayesian model output fit with JAGS, Stan, or other MCMC samplers.

### Details

The following functions are currently available:

- MCMCsummary - summarize MCMC output for particular parameters of interest
- MCMCpstr - summarize MCMC output for particular parameters of interest while preserving original parameter structure
- MCMCtrace - create trace and density plots of MCMC chains for particular parameters of interest
- MCMCchains - easily extract posterior chains from MCMC output for particular parameters of interest
- MCMCplot - create caterpillar plots from MCMC output for particular parameters of interest)

Example data can be loaded using `data(MCMC_data)`.

'MCMCvis' was designed to perform key functions for MCMC analysis using minimal code, in order to free up time/brainpower for interpretation of analysis results. Functions support simple and straightforward subsetting of model parameters within the calls, and produce presentable, 'publication-ready' output.

The vignette can be run using `vignette('MCMCvis')` if vignette is built when installing package.

**Author(s)**

Casey Youngflesh <caseyyoungflesh@gmail.com>

---

MCMC\_data

*Simulated MCMC output data*

---

**Description**

Sample MCMC output containing 30 parameters - alpha[1], ... , alpha[10], beta[1], ... , beta[10], gamma[1], ... , gamma[10].

**Usage**

MCMC\_data

**Format**

mcmc.list object with 3 chains for each parameter, 6000 iterations for each chain

# Index

## \*Topic **datasets**

MCMC\_data, [12](#)

MCMC\_data, [12](#)

MCMCchains, [2](#)

MCMCplot, [3](#)

MCMCpstr, [6](#)

MCMCsummary, [7](#)

MCMCtrace, [9](#)

MCMCvis, [11](#)

MCMCvis-package (MCMCvis), [11](#)