

MCMCvis: Tools to Visualize, Manipulate, and Summarize MCMC Output

Casey Youngflesh¹

¹ Stony Brook University

DOI: [10.21105/joss.00586](https://doi.org/10.21105/joss.00586)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 16 February 2018

Published: 18 February 2018

Licence

Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

MCMCvis is an R (R Core Team 2018) package to used to visualize, manipulate, and summarize MCMC output. MCMC output may be derived from Bayesian model output fit with JAGS (Plummer 2003), Stan (Carpenter et al. 2017), or other MCMC samplers. 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 and ‘publication-ready’ output.

The package contains five functions:

- MCMCsummary - summarize MCMC output for particular parameters of interest
- MCMCpstr - summarize MCMC output for particular parameters of interest while preserving 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

Installation

The released version of MCMCvis can be installed from CRAN with:

```
install.packages('MCMCvis')
```

Or the latest, development version from Github with:

```
install.packages('devtools')  
devtools::install_github('caseyyoungflesh/MCMCvis', build_vignettes = TRUE)
```

Vignette

The vignette for this package can be run using:

```
vignette('MCMCvis')
```

Examples

Summarize

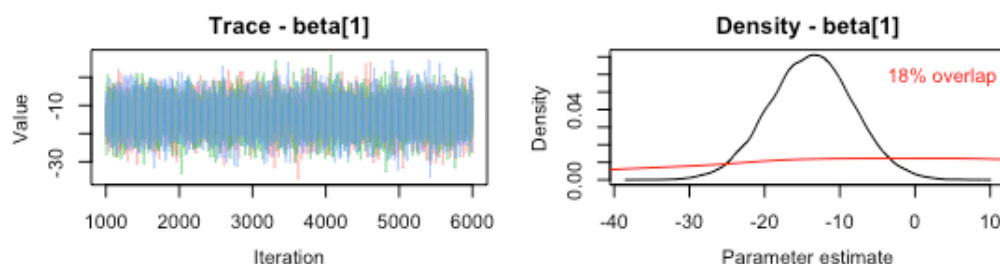
Output summary information from MCMC output. Rhat values and number of effective samples (if specified) can be output to assess convergence of chains. Parameters of interest can be specified in order to only display the desired information.

```
data(MCMC_data)
MCMCsummary(MCMC_data, params = 'beta', n.eff = TRUE)
#>      mean   sd  2.5%   50%  97.5% Rhat n.eff
#> beta[1] -13.83 5.53 -24.67 -13.78 -2.96    1 18000
#> beta[2]  -5.60 0.14  -5.88  -5.60 -5.32    1 17868
#> beta[3] -16.82 1.71 -20.15 -16.82 -13.45    1 18279
#> beta[4] -19.55 2.61 -24.59 -19.55 -14.41    1 17714
#> beta[5]   8.68 5.26  -1.66   8.69 19.04    1 18435
#> beta[6]   2.86 7.68 -12.04   2.83 17.93    1 17754
#> beta[7]   2.06 7.78 -13.11   2.01 17.27    1 18000
#> beta[8] -15.95 3.62 -23.08 -15.93  -8.93    1 17904
#> beta[9]   8.44 4.64  -0.69   8.44 17.58    1 17949
#> beta[10] 16.69 4.36   8.24  16.66 25.40    1 18993
```

Evaluate

Assess convergence by checking posterior chains. Prior posterior overlap (PPO) can be easily assessed by supplying an argument to the function. The degree of overlap is plotted and calculated to determine how informative the data was for the posterior distribution, in relation to the prior distribution used in the model.

```
PR <- rnorm(15000, 0, 32)
MCMCtrace(MCMC_data, params = 'beta\\[1\\]', ISB = FALSE, priors = PR)
```



Manipulate

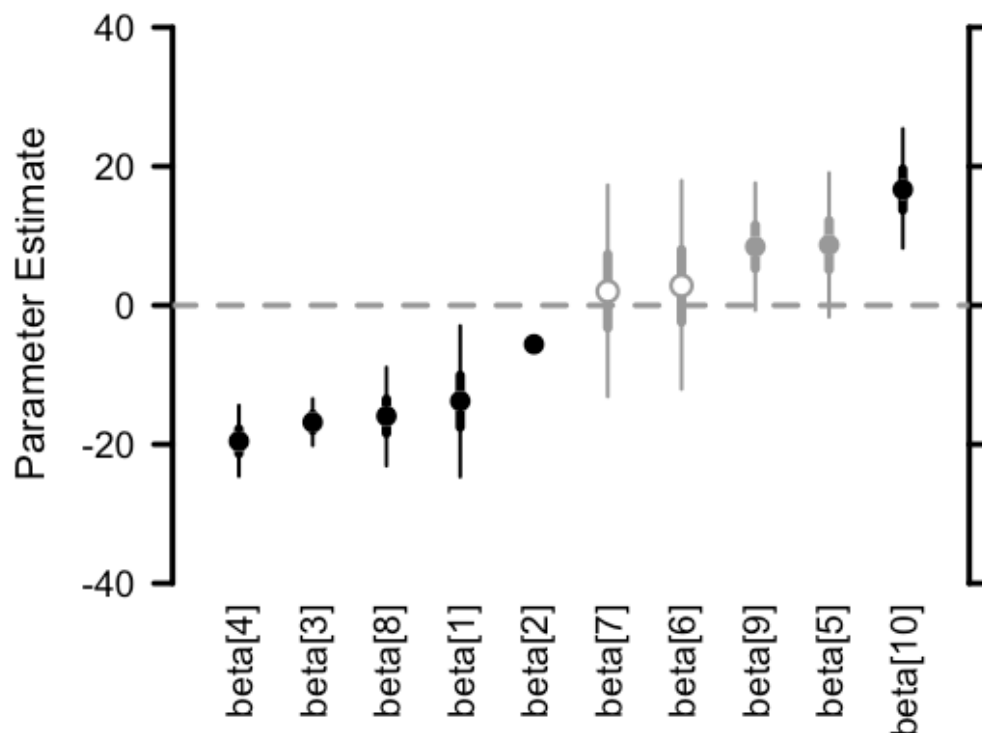
Extract posterior chains for parameters of interest. In this way, large MCMC objects can be subset to smaller, more manageable, objects. Output can be in matrix format, or 'mcmc.list' format.

```
just_betas_mcmc_obj <- MCMCchains(MCMC_data, params = 'beta', mcmc.list = TRUE)
```

Visualize

Easily visualize posterior output for parameters of interest using caterpillar plots. A number of plotting options exist to make sure the desired plot is created.

```
MCMCplot(MCMC_data,  
  params = 'beta',  
  rank = TRUE,  
  horiz = FALSE)
```



References

Carpenter, Bob, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. “Stan: A Probabilistic Programming Language.” *Journal of Statistical Software, Articles* 76 (1):1–32. <https://doi.org/10.18637/jss.v076.i01>.

Plummer, Martyn. 2003. “JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling.”

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org>.