

# Package ‘bayesmix’

February 14, 2012

**Version** 0.7-2

**Date** 2011-10-16

**Author** Bettina Gruen

**Maintainer** Bettina Gruen <Bettina.Gruen@jku.at>

**Title** Bayesian Mixture Models with JAGS

**Depends** R (>= 2.10), rjags (>= 2.1.0), coda (>= 0.13)

**Description** Bayesian mixture models of univariate Gaussian distributions using JAGS

**URL** <http://ifas.jku.at/gruen/BayesMix>

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2011-10-17 05:54:35

## R topics documented:

BMMdiag . . . . .	2
BMMmodel . . . . .	3
BMMposteriori . . . . .	4
BMMpriors . . . . .	5
darwin . . . . .	6
fish . . . . .	7
initsFS . . . . .	7
JAGScall . . . . .	8
JAGScontrol . . . . .	9
JAGSrun . . . . .	10
plot.BMMposteriori . . . . .	11
plot.jags . . . . .	12
priors . . . . .	13
randomPermutation . . . . .	14
Sort . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

**Description**

Two different plots are currently provided: a plot of different variables against each other and a plot of the same variable against its values in the other classes.

**Usage**

```
BMMdiag(object, which = 1:2, variables, ask = interactive(), fct1,
         fct2, xlim, ylim, auto.layout = TRUE, caption = NULL,
         main = "", ...)
```

**Arguments**

<code>object</code>	a jags object with model of class <code>BMMmodel</code> .
<code>which</code>	if only one of the plots is required, specify its number.
<code>variables</code>	if <code>variables</code> is missing, the names are taken from the jags object.
<code>ask</code>	prompt user before each page of plots
<code>fct1</code>	string: name of transformation function for variable on x-axis.
<code>fct2</code>	string: name of transformation function for variable on y-axis.
<code>xlim</code>	if no range for <code>xlim</code> is specified, a sensible range is taken.
<code>ylim</code>	if no range for <code>ylim</code> is specified, a sensible range is taken.
<code>auto.layout</code>	logical: if TRUE puts each of the two different plots on one figure.
<code>caption</code>	captions to appear above the plots.
<code>main</code>	title to each plot (in addition to the above 'caption').
<code>...</code>	further graphical parameters (see 'plot.xy' and 'par') may also be supplied as arguments.

**Details**

The plots help determining which variable will induce a unique labelling when taken for ordering of the segments and indicate if the model is overfitted by specifying too many segments.

**Author(s)**

Bettina Gruen

---

BMMmodel	<i>Creates bugs model.</i>
----------	----------------------------

---

### Description

Creates the BUGS-model specification and return the values for the initialization, the prior specification and the observations.

### Usage

```
BMMmodel(y, k, priors, inits = "initsFS", aprioriWeights = 1,
          no.empty.classes = FALSE, restrict = "none", ...)
```

### Arguments

y	a numeric vector.
k	integer indicating the number of segments.
priors	specification of priors by a named list or a BMMpriors object.
inits	specification of initial values by a named list or string indicating the function to be called.
aprioriWeights	specification of prior of the a-priori weights. If aprioriWeights does not have length = k, there is an equal prior for the a-priori weights assumed.
no.empty.classes	logical: should it be prevented that empty classes arise during sampling.
restrict	one of "none", "mu", "tau".
...	further parameters for the function specified in inits.

### Details

By default the function `initsFS` is called for generating initial values. Any other function specified by `inits` is assumed to have at least `x`, `k` and `restrict` as input parameters.

The parameter `restrict` indicates if a location-shift model ("tau"), a scale contaminated model ("mu") or a model where both variables vary over components shall be fitted.

If the logical `no.empty.classes` is TRUE there are observations added to the model that the classes are not empty. This signifies that the likelihood when sampling the class affiliations is changed thus that any data point which is sampled and is the last one in its class stays there.

### Value

If `y` is specified an object of class `BMMmodel` is returned with components:

<code>inits</code>	named list.
<code>data</code>	named list.
<code>bugs</code>	text for .bug-file with prefix missing.

If `y` is missing an object of class `BMMsetup` is returned containing the parameter specifications. When `JAGScall` is called with this object as model argument, `BMMmodel` is called with `y` and the other parameters as input arguments before calling JAGS.

### Author(s)

Bettina Gruen

### See Also

[JAGSrun](#), [initsFS](#)

### Examples

```
data("fish", package = "bayesmix")
model <- BMMmodel(fish, k = 4, priors = list(kind = "independence",
      parameter = "priorsFish", hierarchical = "tau"),
      initialValues = list(S0 = 2))
model
```

---

BMMposteriori

*Plots aposteriori probabilities of data points*

---

### Description

Given a `jags` object with model of class `BMMmodel` the aposteriori probabilities are determined. If `plot=TRUE`, the resulting object of class `BMMposteriori` is plotted.

### Usage

```
BMMposteriori(object, class, caption = NULL, plot = TRUE,
      auto.layout = TRUE, ...)
```

### Arguments

<code>object</code>	a <code>jags</code> object with model of class <code>BMMmodel</code> .
<code>class</code>	a vector of integers indicating for which classes the posterior probabilities shall be plotted. The default is all.
<code>caption</code>	captions to appear above the plots.
<code>plot</code>	logical indicating if a plot shall be made.
<code>auto.layout</code>	logical: if <code>TRUE</code> puts all classes in the posterior probabilities plot on the same figure.
<code>...</code>	further graphical parameters may also be supplied as arguments.

### Details

Given a `jags` object with model of class `BMMmodel` the a posteriori probabilities are calculated for the unique data points with respect to the components specified by `class`.

**Value**

There is a `BMMposteriori` object returned which is a list including the following components

<code>data</code>	vector of unique data points.
<code>post</code>	a matrix including the posteriori probability of the data points for each class.

**Author(s)**

Bettina Gruen

**See Also**

[plot.BMMposteriori](#)

---

<code>BMMpriors</code>	<i>Creates a 'BMMpriors' object</i>
------------------------	-------------------------------------

---

**Description**

This function enables a comfortable creation of `BMMpriors` objects, which can be used for specifying the priors of a `BMMmodel`.

**Usage**

```
BMMpriors(specification, y, eps = 10^-16)
```

**Arguments**

<code>specification</code>	named list including <code>kind</code> , <code>parameter</code> , <code>hierarchical</code> and <code>mod</code> .
<code>y</code>	a numeric vector.
<code>eps</code>	a numeric value indicating the smallest value for flat priors.

**Details**

In `specification` `kind` can be used for specifying if an "independent" or a "conditionallyconjugate" prior shall be used. `parameter` can be a named list of values for the prior variables or a function name (e.g., "priorsUncertain", "priorsFish", "priorsRaftery"). `hierarchical` can be `NULL` or "tau" if a hierarchical prior shall be taken for  $\tau$ . `mod` is a named list which provides the possibility to override the values from `parameter`.

**Value**

There is an object `BMMpriors` returned with components

<code>name</code>	vector indicating which kind of prior is specified and if it is an hierarchical prior and if appropriate with respect to which variable.
<code>var</code>	list of variables in the prior and their specified values.

**Author(s)**

Bettina Gruen

**Examples**

```
data("fish", package = "bayesmix")
priors <- BMMpriors(y = fish)
```

---

darwin

*Differences in heights between plants*

---

**Description**

A numeric vector containing 15 observations of differences in heights between pairs of self-fertilized and cross-fertilized plants grown in the same condition.

**Usage**

```
data(darwin)
```

**Format**

A numeric vector of length 15.

**Details**

Darwin's data set contains two extremely small values. Therefore, this data set can be used for outlier modelling.

**Source**

Abraham, B. and G. Box (1978) Linear models and spurious observations. *Applied Statistics*, **27**, 131–8.

**Examples**

```
data("darwin", package = "bayesmix")
## Estimated sample density
plot(density(darwin[[1]]), ylim = c(0, 0.02), main = "Outlier modelling")
ss <- seq(-100, 100, by = 1)
## Normal density with estimated mean and sd of whole sample
lines(ss, dnorm(ss, mean = mean(darwin[[1]]), sd = sd(darwin[[1]])), col = "red")
## Normal density with estimated mean and sd of sample, where the 2
## extremely small values are removed
lines(ss, dnorm(ss, mean = mean(darwin[-c(1:2),1]),
               sd = sd(darwin[-c(1:2),1])), col = "green")
```

---

fish	<i>Fish length data</i>
------	-------------------------

---

**Description**

A numeric vector containing 256 observations of fish lengths.

**Usage**

```
data(fish)
```

**Format**

A numeric vector of length 256.

**Details**

This data set can be used for modeling unobserved heterogeneity, as it can be assumed that underlying categories present in the data are the age groups to which the fish belong.

**Source**

D. M. Titterington, A. F. M. Smith and U.E. Makov (1985) *Statistical Analysis of Finite Mixture Distributions*. Wiley.

**Examples**

```
data("fish", package = "bayesmix")
ss <- seq(-3, 13, by = 0.01)
hist(fish[[1]], 20, freq = FALSE, main = "Fish data")
lines(ss, dnorm(ss, mean(fish[[1]]), sd(fish[[1]])), col = "red")
```

---

initsFS	<i>create initial values</i>
---------	------------------------------

---

**Description**

Initial values for nodes are created after the suggestion in Sylvia Fruehwirth-Schnatter's book.

**Usage**

```
initsFS(x, k, restrict, initialValues = list())
```

**Arguments**

x	a numeric vector.
k	number of segments.
initialValues	additional initial values specifications.
restrict	one of "none", "mu", "tau".

**Details**

The initial values for  $\mu$  are determined by the quantiles of the data, those for  $\eta$  give equal weight on each segment and those for  $\tau$  are equal for all segments and estimated by the inverse of the IQR of the data divided by 1.34 and squared.

**Value**

A list with initial values for the parameter indicated by the name of the respective list element is returned.

**Author(s)**

Bettina Gruen

---

JAGScall	<i>Calls jags</i>
----------	-------------------

---

**Description**

Using functionality from package rjags the JAGS model is created and posterior draws are sampled.

**Usage**

```
JAGScall(model, y, prefix, control, ...)
```

**Arguments**

model	JAGSmodel object or output from BMMmodel.
y	a numeric vector.
prefix	character: prefix for .bug-file.
control	named list or JAGScontrol object.
...	additional parameters handed over to BMMmodel.

**Value**

Returns a "jags" object.

**Author(s)**

Bettina Gruen

**See Also**[JAGSrun](#)

---

**JAGScontrol***Control parameters for the sampling.*

---

**Description**

The information on the number of burn-in draws and monitored draws is specified. Furthermore, it includes the information which variables shall be monitored and possibly a seed and a random number generator (RNG).

**Usage**

```
JAGScontrol(variables, n.iter = 1000, thin = 1, burn.in = 0, seed,
            rng = c("base::Wichmann-Hill",
                  "base::Marsaglia-Multicarry",
                  "base::Super-Duper",
                  "base::Mersenne-Twister"))
```

**Arguments**

<code>variables</code>	names of variables which shall be monitored.
<code>n.iter</code>	number of monitored draws.
<code>thin</code>	thinning interval.
<code>burn.in</code>	number of discarded burn-in draws.
<code>seed</code>	integer setting the seed for the RNG.
<code>rng</code>	specification of RNG.

**Value**

An object of class `JAGScontrol` is returned which is a list containing the specified information.

**Author(s)**

Bettina Gruen

**See Also**[JAGSrun](#)

**Examples**

```
control <- JAGScontrol(variables = "mu")
control
```

---

 JAGSrun

*MCMC sampling of Bayesian models*


---

**Description**

Calls jags for MCMC sampling.

**Usage**

```
JAGSrun(y, prefix = yname, model = BMMmodel(k = 2),
        control = JAGScontrol(variables = c("mu", "tau", "eta")),
        tmp = TRUE, cleanup = TRUE, ...)
```

**Arguments**

y	a numeric vector.
prefix	character: prefix for .bug-file.
model	object of class JAGSmodel or output from BMMmodel.
control	specification of control by a JAGScontrol object.
tmp	logical: shall the files be written in a temporary directory.
cleanup	logical: shall the created files be removed.
yname	a character string with the actual y argument name.
...	further parameters handed over to BMMmodel where it is used for the function specifying the initial values, e.g., initsFS.

**Details**

This function is a wrapper calling JAGScall.

**Value**

Returns a jags object with components

call	the matched call.
results	results read in from "jags.out" if run was successful or from "jags.dump" if an error occurred.
model	a JAGSmodel object.
variables	vector containing the names of the monitored variables.
data	a numeric vector.

**Author(s)**

Bettina Gruen

**See Also**[JAGScall](#), [BMMmodel](#), [initsFS](#)**Examples**

```

data("fish", package = "bayesmix")
prefix <- "fish"
variables <- c("mu", "tau", "eta")
k <- 3
modelFish <- BMMmodel(k = k, priors = list(kind = "independence",
                                          parameter = "priorsFish", hierarchical = "tau"))
controlFish <- JAGScontrol(variables = c(variables, "S"), n.iter = 100)
z1 <- JAGSrun(fish, prefix, model = modelFish, initialValues = list(S0 = 2),
             control = controlFish, cleanup = TRUE, tmp = FALSE)
zSort <- Sort(z1, "mu")
BMMposteriori(zSort)

data("darwin", package = "bayesmix")
prefix <- "darwin"
k <- 2
modelDarwin <- BMMmodel(k = k, priors = list(kind = "independence",
                                             parameter = "priorsUncertain"),
                      aprioriWeights = c(1, 15),
                      no.empty.classes = TRUE, restrict = "tau")
z2 <- JAGSrun(darwin, prefix, model = modelDarwin, control =
             JAGScontrol(variables = variables, n.iter = 3000,
                          burn.in = 1000), cleanup = TRUE, tmp = FALSE)
plot(z2, variables = "mu")

```

---

plot.BMMposteriori      *Plots a posteriori probabilities of data points*

---

**Description**

Plot method for object of class BMMposteriori, typically called by BMMposteriori.

**Usage**

```

## S3 method for class 'BMMposteriori'
plot(x, caption, main = "", ...)

```

**Arguments**

x	a BMMposteriori object.
caption	captions to appear above the plots.
main	title to each plot-in addition to the above caption.
...	further graphical parameters may also be supplied as arguments.

**Details**

This function is called by BMMposteriori if plot = TRUE.

**Author(s)**

Bettina Gruen

**See Also**

[BMMposteriori](#)

---

plot.jags	<i>Plot jags Object</i>
-----------	-------------------------

---

**Description**

Plots mcmc chains of a jags object.

**Usage**

```
## S3 method for class 'jags'
plot(x, variables = NULL, trace = TRUE, density = TRUE,
     smooth = TRUE, bwf, num, xlim, auto.layout = TRUE,
     ask = interactive(), ...)
```

**Arguments**

x	a jags object.
variables	names of variables which shall be plotted. Default are all names of results except those with a column dimension larger than the number of classes k.
trace	plot trace of each variable.
density	plot density estimate of each variable.
smooth	draw a smooth line through trace plots.
bwf	bandwidth function for density plots.
num	if not all classes of a variable shall be plotted, a subset can be specified.
xlim	if not specified, the range of each variable over all classes is taken as default.
auto.layout	automatically generate output format.
ask	prompt user before each page of plots.
...	further arguments for densityplot.

**Details**

Adapted from `plot.mcmc`.

Currently only implemented for `jags` objects with model of class `BMMmodel`. Otherwise the default plot method for the results of the `jags` object is called (`plot.mcmc`).

**Author(s)**

Bettina Gruen

**See Also**

[plot.mcmc](#), [BMMdiag](#), [BMMposteriori](#)

---

priors

*Creates list of prior specifications*

---

**Description**

Given the data values for the priors are determined.

**Usage**

```
priorsFish(y, eps = 10^-16)
priorsRaftery(y)
priorsUncertain(y, eps = 10^-16)
```

**Arguments**

<code>y</code>	a numeric vector.
<code>eps</code>	a numeric value indicating the smallest value for flat priors.

**Details**

Values for the prior parameter `b0`, `B0`, `nu0` and `S0` are determined.

**Value**

There is a list returned with named components of the prior parameters.

**Author(s)**

Bettina Gruen

---

randomPermutation	<i>Randomly permute segments for MCMC draws</i>
-------------------	---

---

**Description**

Random permutation of segment labels for each draw in order to get a better estimate of the unrestricted likelihood.

**Usage**

```
randomPermutation(x)
```

**Arguments**

x                    a jags object with model of class BMMmodel.

**Details**

The draws are permuted with respect to the different classes k.

**Value**

The input object with permuted results for each draw is returned.

**Warning**

Any variables where there are neither k different chains nor only one chain observed are dropped.

**Author(s)**

Bettina Gruen

---

Sort	<i>Sorts MCMC chains according to certain variables</i>
------	---

---

**Description**

Ascending sorting of results of jags object with model of class BMMmodel with respect to a given variable.

**Usage**

```
Sort(x, by = NULL)
```

**Arguments**

- x a jags object with model of class BMMmodel.
- by variable name according to which the segments shall be ordered.

**Details**

If by is not specified, the first variable in the corresponding vector of the jags object is taken.

**Value**

The input object with results sorted in ascending order according to the variable given in by is returned.

**Warning**

If there arise problems, the original object is returned with a warning.

**Author(s)**

Bettina Gruen

# Index

## \*Topic **datasets**

darwin, [6](#)  
fish, [7](#)

## \*Topic **hplot**

BMMdiag, [2](#)  
BMMposteriori, [4](#)  
plot.BMMposteriori, [11](#)  
plot.jags, [12](#)

## \*Topic **utilities**

BMMmodel, [3](#)  
BMMpriors, [5](#)  
initsFS, [7](#)  
JAGScall, [8](#)  
JAGScontrol, [9](#)  
JAGSrun, [10](#)  
priors, [13](#)  
randomPermutation, [14](#)  
Sort, [14](#)

BMMdiag, [2](#), [13](#)

BMMmodel, [3](#), [11](#)

BMMposteriori, [4](#), [12](#), [13](#)

BMMpriors, [5](#)

darwin, [6](#)

fish, [7](#)

initsFS, [4](#), [7](#), [11](#)

jags (JAGSrun), [10](#)

JAGScall, [8](#), [11](#)

JAGScontrol, [9](#)

JAGSrun, [4](#), [9](#), [10](#)

plot.BMMposteriori, [5](#), [11](#)

plot.jags, [12](#)

plot.mcmc, [13](#)

print.jags (JAGSrun), [10](#)

print.JAGSmodel (BMMmodel), [3](#)

priors, [13](#)

priorsFish (priors), [13](#)

priorsRaftery (priors), [13](#)

priorsUncertain (priors), [13](#)

randomPermutation, [14](#)

Sort, [14](#)