

Package ‘rbugs’

February 20, 2015

Title Fusing R and OpenBugs and Beyond

Date 2011-02-19

Version 0.5-9

Author Jun Yan <jyan@stat.uconn.edu> and Marcos Prates
<marcosop@est.ufmg.br>

Description Functions to prepare files needed for running BUGS in batch-mode, and running BUGS from R. Support for Linux and Windows systems with OpenBugs is emphasized.

Suggests coda (>= 0.13-5)

Maintainer Marcos Prates <marcosop@est.ufmg.br>

License GPL (>= 3.0)

Repository CRAN

Date/Publication 2013-04-09 07:12:16

NeedsCompilation no

R topics documented:

format4Bugs	2
genBugsScript	2
genDataFile	4
genInitsFile	5
getBugsOutput	6
print.rbugs	7
pumps	7
rbugs	8
rbugs2coda	9
rio	10
runBugs	12
schools	13

Index	15
--------------	-----------

`format4Bugs`*Format a Data List as BUGS Data Input*

Description

Transform a data list into a format which can be read in as data by BUGS.

Usage

```
format4Bugs(dataList, digits = 5)
```

Arguments

`dataList` A list of data to be formatted.
`digits` The desired number of significant digits.

Value

A character vector of data readable by BUGS.

Author(s)

Jun Yan <jyan@stat.uiowa.edu> and Marcos Prates <marcosop@est.ufmg.br>

Examples

```
dat <- list(a = runif(1), b=rnorm(2), c=matrix(rexp(4), 2, 2))
format4Bugs(dat)
```

`genBugsScript`*Generating Script File for BUGS*

Description

Generating a script file to run BUGS batch-mode.

Usage

```
genBugsScript(paramSet, n.chains, n.iter, n.burnin, n.thin, dic,
              model.file, data.file, inits.files,
              bugsWorkingDir=getwd(),
              script, debug = FALSE,
              OpenBugs=TRUE, Windows=TRUE, seed=NULL)
```

Arguments

paramSet	A character vector of the names of the parameter to be monitored.
n.chains	The number of chains to be monitored.
n.iter	The number of iterations of each chain.
n.burnin	The length of the burn-in.
n.thin	Thinning rate.
dic	If TRUE, dic will be monitored
model.file	A character string of the name of file which contains the model specification for BUGS. It should end with .txt to be recognizable by BUGS.
data.file	A character string of the name of the data file. It should end with .txt to be recognizable by BUGS.
inits.files	A vector of file names of initial values. They should end with .txt to be recognizable by BUGS.
bugsWorkingDir	A character string specifying the directory to store temporary files for running BUGS.
script	A character string naming the files to print the script to. It must end with .txt to be recognizable by BUGS.
debug	A logical value indicating whether or not closing the BUGS window after running the script.
OpenBugs	If TRUE, OpenBugs is used.
Windows	If TRUE, Windows plataform is used.
seed	An integer of random number seed.

Details

This function only write a script to 'script', which uses the file names of 'model.file', 'data.file', and 'inits.files', without actually reading them. These file names should be correct when using [runBugs](#).

workingDir and bugsWorkingDir establish the translation between a directory native to the operating system and a directory recognizable by wine. They are the same if useWine == FALSE.

Value

None.

Author(s)

Jun Yan <jyan@stat.uconn.edu> and Marcos Prates <marcosop@est.ufmg.br>

See Also

[genDataFile](#), [genInitsFile](#), [format4Bugs](#)

Examples

```
## no tested examples for mac-os.
## Not run:
script.file <- paste(tempfile("script"), ".txt", sep="")
genBugsScript(paramSet=c("alpha", "theta"), n.chains=2, n.keep=1500,
              n.burnin=500, n.thin=1,
              model.file=~ /tmp/model.txt",
              data.file=~ /tmp/data.txt",
              inits.file=c("~/tmp/init1.txt",
                           "~/tmp/init2.txt"),
              bugsWorkingDir=" ~/tmp/",
              script=script.file)
file.show(script.file)
unlink(script.file)

## End(Not run)
```

genDataFile

Generating the Data File for BUGS

Description

Generating the data file which can be used in the script file for running BUGS batch-mode.

Usage

```
genDataFile(dataList, dataFile)
```

Arguments

dataList	A list of data that are needed by BUGS. Its elements must be numeric vector or matrices.
dataFile	A character string naming the file to print to. It must end with .txt to be recognizable by BUGS.

Value

None.

Author(s)

Jun Yan <jyan@stat.uconn.edu> and Marcos Prates <marcosop@est.ufmg.br>

See Also

[genInitsFile](#), [genBugsScript](#), [format4Bugs](#)

Examples

```
dat <- list(a = runif(1), b=rnorm(2), c=matrix(rexp(4), 2, 2))
genDataFile(dat, "foo.txt")
file.show("foo.txt")
unlink("foo.txt")
```

genInitsFile*Generating Initial Value Files for BUGS*

Description

Generating the files of initial values for MCMC which can be used in the script for running BUGS in batch-mode.

Usage

```
genInitsFile(n.chains, inits, initsFileStem)
```

Arguments

<code>n.chains</code>	The number of chains to run.
<code>inits</code>	A list of list of initial values or a function which returns a list of initial values. See details.
<code>initsFileStem</code>	A character string naming the files to print to. See details.

Details

if `inits` is a list, it should have length `n.chains`, and each element of the list should be a list which contains one set of initial values. If `inits` is a function, it will be run `n.chains` times to generate a list of list of initial values.

The `initsFileStem` is the stem of the file names. The resulted file names end with `.txt`. For example, if `n.chains` is 3 and `initsFileStem` is "init", then the file names will be `init1.txt`, `init2.txt`, and `init3.txt`.

Value

None.

Author(s)

Jun Yan <jyan@stat.uiowa.edu> and Marcos Prates <marcosop@est.ufmg.br>

See Also

[genDataFile](#), [genBugsScript](#)

Examples

```
## when inits is a list:
inits <- list(list(alpha=1, beta=2), list(alpha=4, beta=4))
genInitsFile(2, inits, "foo.init")
file.show("foo.init1.txt")
file.show("foo.init2.txt")
## when inits is a function:
inits <- function() list(alpha=rnorm(2, sd=100),
                        beta=rgamma(1, 0.1, 0.001))
genInitsFile(2, inits, "foo.init")
file.show("foo.init1.txt")
file.show("foo.init2.txt")
unlink("foo.init1.txt")
unlink("foo.init2.txt")
```

getBugsOutput

Collect the MCMC samples from BUGS

Description

Collect the MCMC samples from BUGS.

Usage

```
getBugsOutput(n.chains, workingDir, OpenBugs = TRUE)
```

Arguments

n.chains	The number of chains BUGS had run.
workingDir	A character string specifying the name of the directory where the output files are saved.
OpenBugs	If TRUE, OpenBugs is used.

Details

This function assumes that under `workingDir`, there is a coda index file `'codaIndex.txt'`, and the coda output for `n.chains` chains are named as `coda1.txt`, `coda2.txt`, ...

Value

A list of matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively.

Author(s)

Jun Yan <jyan@stat.uconn.edu> and Marcos Prates <marcosop@est.ufmg.br>

print.rbugs	<i>Printing a rbugs object</i>
-------------	--------------------------------

Description

Printing a rbugs object

Usage

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

Arguments

x	an object of class 'rbugs', see rbugs for details
...	further arguments to print

See Also

[rbugs](#)

pumps	<i>10 Power Plant Pumps</i>
-------	-----------------------------

Description

10 Power Plant Pumps.

Usage

```
data(pumps)
```

Format

A data frame with 10 observations on the following 2 variables.

t The length of the operation time.

x The number of failures.

Examples

```

data(pumps)
pumps.data <- list(t = pumps$t, x = pumps$x, N = nrow(pumps))
pumps.model <- system.file("bugs/model/pumps.bug", package="rbugs")
file.show(pumps.model)
pumps.inits <- system.file("bugs/inits/pumps.txt", package="rbugs")
file.show(pumps.inits)
inits <- list(dget(pumps.inits))
parameters <- c("theta", "alpha", "beta")

## Not run:
pumps.sim <- rbugs(data = pumps.data, inits, parameters,
                  pumps.model, n.chains = 1, n.iter = 1000,
                  workingDir=~"/tmp/",
                  useWine=TRUE)

## MCMC Analysis
library("coda")
pumps.mcmc <- rbugs2coda(pumps.sim)
summary(pumps.mcmc)
effectiveSize(pumps.mcmc)

## End(Not run)

```

rbugs

Run BUGS from R in a BATCH

Description

Generate files (data, init, script) that are necessary to run BUGS, call BUGS through the OS, and collect the MCMC output.

Usage

```

rbugs(data, inits, paramSet, model,
       n.chains = 1, n.iter = 2000, n.burnin = floor(n.iter/2),
       n.thin = max(1, floor(n.chains * (n.iter - n.burnin)/1000)),
       dic = FALSE,
       debug = FALSE,
       bugs = system("which OpenBUGS", TRUE),
       bugsWorkingDir,
       OpenBugs = TRUE,
       cleanBugsWorkingDir = FALSE,
       genFilesOnly = FALSE,
       verbose = FALSE, seed = NULL)

```


Arguments

data	a list of data object to be used by BUGS
inits	a list of list of initial values, or a function which returns a list of initial values
paramSet	a vector of the names of the parameters to be monitored
model	the file name of the model description
n.chains	the number of chains to be monitored
n.iter	the number of iteration of each chain
n.burnin	the length of the burn-in
n.thin	thinning rate
dic	if TRUE, dic will be monitored
debug	a logical value indicating whether or not closing the BUGS window
bugs	the full name (including the path) of the BUGS executable
bugsWorkingDir	a directory to store all the intermediate files
OpenBugs	if TRUE, OpenBugs is used
cleanBugsWorkingDir	if TRUE, the generated files will be removed from the bugsWorkingDir
genFilesOnly	If TRUE, the script, data, inits, and model files will be generated but not run.
verbose	if TRUE, print the log file from BUGS
seed	an interger of random number seed

Value

A list with the inforamtion of the MCMC sample and each one of the nChains .

Author(s)

Jun Yan <jyan@stat.uconn.edu> and Marcos Prates <marcosop@est.ufmg.br>

Examples

```
##see examples in \code{\link{pumps}} and \code{\link{schools}}
```

rbugs2coda

Rbugs to coda

Description

Convert a rbugs object to a coda format

Usage

```
rbugs2coda(model, burnin = NULL, thin = NULL)
```

Arguments

model	an object of class 'rbugs', see rbugs for details
burnin	a burn-in period for the chains
thin	the thinning interval between consecutives observations

See Also

[rbugs](#)

rio

Dengue Fever in the State of Rio de Janeiro

Description

Shape file of the state of Rio de Janeiro

Usage

```
data(rio)
```

Format

A shape file with 92 observations on the following 9 variables.

GeoCode Rio de Janeiro counties numeric codes

Name Rio de Janeiro counties names

Lat Rio de Janeiro counties latitude

Long Rio de Janeiro counties longitude

Pop Rio de Janeiro counties population

Income Rio de Janeiro counties percentage of homes living with less than one minimum salary

Urban Rio de Janeiro counties percentage of urban area

Dengue Counts of dengue fever cases by county in the state of Rio de Janeiro

E Expected counts of dengue fever cases by county in the state of Rio de Janeiro, conditioned on the total number of cases the stratified by population rate

Examples

```
data(rio)

## Not run:
library(spdep)
rio.bug <- system.file("bugs/model/dengue.bug", package="rbugs")

## Get the neighbor structure for Rio
rj.nb=poly2nb(rio)
```

```
## Number of cases by county
Y = rio@data$Dengue

## Get covariates
X <- as.matrix(rio@data[,c("Urban","Income")])

## Get expected values
E <- rio@data$E

## Number of covariates
P <- ncol(X)

## Total counties
N <- length(rj.nb)

## Number of neighbors of each county
num <- sapply(rj.nb, length)

## Adjacency neighbor list of each county
adj <- unlist(rj.nb)

## Total sum of the number of neighbors in the map
sumNumNeigh <- length(adj)

## Set data file
data.rio <- list (N=N, P=P, Y=Y, adj=adj, X=X, E=E, num=num, sumNumNeigh=sumNumNeigh)

## Generate the initial values for spatial vector (u), random noise (v) and covariates effects (beta)
u.aux <- rep(0,N)
v.aux <- rep(0,N)
beta.aux <- rep(0,P)

## Generate the list with the necessary information in the init file
inits <- list( list(beta = beta.aux, tau.u = 1, tau.v = 1, u=u.aux, v=v.aux, Int=0.0))

## Set the parameters that will be saved and returned by BUGS
parameters <- c("beta", "tau.u", "tau.v", "RR", "Int")

## no tested examples for mac-os.

rio.sim <- rbugs(data.rio, inits, parameters,
               rio.bug, n.chains=2, n.iter=60000,
               n.burnin = 10000, n.thin = 10,
               bugs="/usr/bin/OpenBUGS",
               bugsWorkingDir="/home/marcos/tmp")

## MCMC analysis
library("coda")
rio.mcmc <- rbugs2coda(rio.sim)
summary(rio.mcmc)
effectiveSize(rio.mcmc)
gelman.diag(rio.mcmc)
```

```
## End(Not run)
```

runBugs	<i>Execute a BUGS Script from R</i>
---------	-------------------------------------

Description

Create the system command to run BUGS and execute it.

Usage

```
runBugs(bugs=system("which OpenBUGS", TRUE), script, n.chains, workingDir,
        OpenBugs = TRUE, Windows=TRUE, verbose = TRUE)
```

Arguments

bugs	A character string specifying the full name (including path) of the BUGS executable. It must end with .txt to be recognizable by BUGS.
script	A character string specifying the full name (including path) of the script to be run. It must end with .txt to be recognizable by BUGS.
n.chains	The number of MCMC chains.
workingDir	The working directory where the output files from BUGS are stored.
OpenBugs	If TRUE, OpenBugs is used.
Windows	If TRUE, Windows platform is used.
verbose	If TRUE, print the log file from BUGS.

Value

None.

Author(s)

Jun Yan <jyan@stat.uconn.edu> and Marcos Prates <marcosop@est.ufmg.br>

See Also

[rbugs](#)

schools	<i>8 schools</i>
---------	------------------

Description

8 schools analysis from section 5.5 of "Bayesian Data Analysis" by Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin.

Usage

```
data(schools)
```

Format

A data frame with 8 observations on the following 3 variables.

school a factor with levels A B C D E F G H

estimate a numeric vector

sd a numeric vector

Source

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B. (2003): *Bayesian Data Analysis*, 2nd edition, CRC Press.

Examples

```
data(schools)
J <- nrow(schools)
y <- schools$estimate
y <- rnorm(length(y))
sigma.y <- schools$sd
schools.data <- list("J", "y", "sigma.y")
## schools.data <- list(J=J, y=y, sigma.y=sigma.y)
inits <- function() {list (theta=rnorm(J,0,100),
                          mu.theta=rnorm(1,0,100),
                          sigma.theta=runif(1,0,100))}
parameters <- c("theta", "mu.theta", "sigma.theta")
schools.bug <- system.file("bugs/model/schools.bug", package="rbugs")
file.show(schools.bug)

## Not run:
## no tested examples for mac-os.

schools.sim <- rbugs(data=schools.data, inits, parameters,
                    schools.bug, n.chains=3, n.iter=1000,
                    bugs="/usr/bin/OpenBUGS",
                    bugsWorkingDir="~/tmp/")
```

```
## generate files only
schools.sim <- rbugs(data=schools.data, inits, parameters,
                    schools.bug, n.chains=3, n.iter=1000,
                    bugsWorkingDir=~"/tmp/",
                    OpenBugs=TRUE, genFilesOnly=TRUE)

## MCMC analysis
library("coda")
schools.mcmc <- rbugs2coda(schools.sim)
summary(schools.mcmc)
effectiveSize(schools.mcmc)
gelman.diag(schools.mcmc)

## End(Not run)
```

Index

*Topic **coda**

 rbugs2coda, 9

*Topic **datasets**

 pumps, 7

 rio, 10

 schools, 13

*Topic **interface**

 format4Bugs, 2

 genBugsScript, 2

 getBugsOutput, 6

 rbugs, 8

 runBugs, 12

*Topic **print**

 print.rbugs, 7

*Topic **utilities**

 format4Bugs, 2

 genBugsScript, 2

 genDataFile, 4

 genInitsFile, 5

 getBugsOutput, 6

format4Bugs, 2, 3, 4

genBugsScript, 2, 4, 5

genDataFile, 3, 4, 5

genInitsFile, 3, 4, 5

getBugsOutput, 6

print, 7

print.rbugs, 7

pumps, 7

rbugs, 7, 8, 10, 12

rbugs2coda, 9

rio, 10

runBugs, 3, 12

schools, 13