

Package ‘rbugs’

July 24, 2010

Title Fusing R and OpenBugs

Date 2010-07-07

Version 0.4-4

Author Jun Yan <jyan@stat.uconn.edu> and Marcos Prates
<marcos.prates@uconn.edu>

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 <marcos.prates@uconn.edu>

License GPL (>= 3.0)

Repository CRAN

Date/Publication 2010-07-24 20:43:12

R topics documented:

| | |
|-------------------------|----|
| format4Bugs | 2 |
| genBugsScript | 2 |
| genDataFile | 4 |
| genInitsFile | 5 |
| getBugsOutput | 6 |
| print.rbugs | 7 |
| pumps | 7 |
| rbugs | 8 |
| rbugs2coda | 10 |
| runBugs | 10 |
| schools | 11 |

| | |
|--------------|-----------|
| Index | 14 |
|--------------|-----------|

`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

| | |
|-----------------------|---|
| <code>dataList</code> | A list of data to be formatted. |
| <code>digits</code> | 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 <marcos.prates@uconn.edu>

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,
              workingDir=NULL, bugsWorkingDir=getwd(),
              script, debug = FALSE, useWine=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. |
| workingDir | A character string specifying the directory to store temporary files for running BUGS. |
| bugsWorkingDir | A character string specifying the directory which is recognizable by windows, if using wine. |
| 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. |
| useWine | If TRUE, BUGS is used via wine. |
| 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 <marcos.prates@uconn.edu>

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="/var/scratch/c/tmp/model.txt",
              data.file="/var/scratch/c/tmp/data.txt",
              inits.file=c("/var/scratch/c/tmp/init1.txt",
                          "/var/scratch/c/tmp/init2.txt"),
              workingDir="/var/scratch/c/tmp",
              bugsWorkingDir="c:/tmp",
              script=script.file,
              debug=FALSE, useWine=TRUE)
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 <marcos.prates@uconn.edu>

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 <marcos.prates@uconn.edu>

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

| | |
|-------------------------|---|
| <code>n.chains</code> | The number of chains BUGS had run. |
| <code>workingDir</code> | A character string specifying the name of the directory where the output files are saved. |
| <code>OpenBugs</code> | 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 <marcos.prates@uconn.edu>

`print.rbugs` *Printing a rbugs object*

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` *10 Power Plant Pumps*

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 <- file.path(.path.package("rbugs"), "bugs/model", "pumps.bug")
file.show(pumps.model)
pumps.inits <- file.path(.path.package("rbugs"), "bugs/inits", "pumps.txt")
file.show(pumps.inits)
inits <- list(dget(pumps.inits))
parameters <- c("theta", "alpha", "beta")

## Not run:
## The workingDir must be a directory which can be seen by wine
## The nominalWorkingDir gives the directory name recognizable for
## Windows.
## The full name of the bugs and wine executables need to set.
## In .Renviron, add lines similar to these:
## If using WinBugs
##   BUGS="c:/program files/winbugs14/winbugs14.exe"
##   WINE="/var/scratch/jyan/wine-20040408/wine"
## If using OpenBUGS
##   BUGS="c:/program files/OpenBUGS310/OpenBugs.exe"
##   WINE="/var/scratch/jyan/wine-20040408/wine"

pumps.sim <- rbugs(data = pumps.data, inits, parameters,
                  pumps.model, n.chains = 1, n.iter = 1000,
                  workingDir="/var/scratch/jyan/c/tmp",
                  bugsWorkingDir="c:/tmp/",
                  useWine=TRUE)

## MCMC Analysis
library("coda")
pumps.mcmc <- as.mcmc(pumps.sim$chain1)
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 = Sys.getenv("BUGS"),
workingDir = NULL,
bugsWorkingDir,
useWine = FALSE, wine = Sys.getenv("WINE"),
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 |
| workingDir | a directory to store all the intermediate files |
| bugsWorkingDir | required, the nominal name of the working directory, which is the same as workingDir if useWine is "FALSE". If useWine is "TRUE", it should be a directory which is recognizable by the Windows system though wine. |
| useWine | a logical value with "TRUE" meaning wine is used |
| wine | if useWine==TRUE, the full name of wine |
| 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 <marcos.prates@uconn.edu>

Examples

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

| | |
|------------|----------------------|
| rbugs2coda | <i>Rbugs to coda</i> |
|------------|----------------------|

Description

Convert a rbugs object to a coda format

Usage

```
rbugs2coda(model)
```

Arguments

model an object of class 'rbugs', see [rbugs](#) for details

See Also

[rbugs](#)

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

Description

Create the system command to run BUGS and execute it.

Usage

```
runBugs(bugs=Sys.getenv("BUGS"), script, n.chains, workingDir,  
        useWine=FALSE, wine=Sys.getenv("WINE"),  
        OpenBugs = TRUE, Windows=TRUE, verbose = TRUE)
```

Arguments

| | |
|-------------------------|--|
| <code>bugs</code> | A character string specifying the full name (including path) of the BUGS executable. It must end with <code>.txt</code> to be recognizable by BUGS. |
| <code>script</code> | A character string specifying the full name (including path) of the script to be run. It must end with <code>.txt</code> to be recognizable by BUGS. |
| <code>n.chains</code> | The number of MCMC chains. |
| <code>workingDir</code> | The working directory where the output files from BUGS are stored. |
| <code>useWine</code> | A logical value indicating if wine is to be used. |
| <code>wine</code> | A character string specifying the full name (including path) of the wine executable. |
| <code>OpenBugs</code> | If TRUE, OpenBugs is used. |
| <code>Windows</code> | If TRUE, Windows plataform is used. |
| <code>verbose</code> | If TRUE, print the log file from BUGS. |

Value

None.

Author(s)

Jun Yan <jyan@stat.uconn.edu> and Marcos Prates <marcos.prates@uconn.edu>

See Also

[rbugs](#)

`schools`

8 schools

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 <- file.path(.path.package("rbugs"), "bugs/model", "schools.bug")
file.show(schools.bug)

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

## Unix using WINE
## The workingDir must be a directory which can be seen by wine
## The nominalWorkingDir gives the directory name recognizable for
## Windows.
## The full name of the bugs and wine executables need to set.
## In .Renviron, add lines similar to these:
## If using WinBugs
##   BUGS="c:/program files/winbugs14/winbugs14.exe"
##   WINE="/var/scratch/jyan/wine-20040408/wine"
## If using OpenBUGS
##   BUGS="c:/program files/OpenBUGS310/OpenBugs.exe"
##   WINE="/var/scratch/jyan/wine-20040408/wine"

schools.sim <- rbugs(data=schools.data, inits, parameters,
                    schools.bug, n.chains=3, n.iter=1000,
                    workingDir="/var/scratch/jyan/c/tmp/",
                    bugsWorkingDir="c:/tmp/",
                    useWine=TRUE,
                    wine="/var/scratch/jyan/wine-20040408/wine",
                    debug=TRUE)

## Unix using OpenBUGS
schools.sim <- rbugs(data=schools.data, inits, parameters,
                    schools.bug, n.chains=3, n.iter=1000,
                    bugs="~/scratch/OpenBUGS310/bin/OpenBUGS",
                    bugsWorkingDir="~/scratch/tmp/",
                    OpenBugs=TRUE)

## generate files only

```

```
schools.sim <- rbugs(data=schools.data, inits, parameters,
                   schools.bug, n.chains=3, n.iter=1000,
                   bugsWorkingDir=~ /scratch/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, 10

*Topic **datasets**

 pumps, 7

 schools, 11

*Topic **interface**

 format4Bugs, 1

 genBugsScript, 2

 getBugsOutput, 6

 rbugs, 8

 runBugs, 10

*Topic **print**

 print.rbugs, 7

*Topic **utilities**

 format4Bugs, 1

 genBugsScript, 2

 genDataFile, 4

 genInitsFile, 5

 getBugsOutput, 6

format4Bugs, 1, 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, 11

rbugs2coda, 10

runBugs, 3, 10

schools, 11