

# Package: dirmcmc (via r-universe)

September 17, 2024

**Type** Package  
**Title** Directional Metropolis Hastings Algorithm  
**Version** 1.3.3  
**Date** 2017-02-17  
**Author** Abhirup Mallik <malli066@umn.edu>  
**Maintainer** Abhirup Mallik <malli066@umn.edu>  
**Depends** R (>= 2.10.0)  
**ByteCompile** TRUE  
**Description** Implementation of Directional Metropolis Hastings Algorithm for MCMC.  
**License** GPL-3  
**LazyData** TRUE  
**RoxygenNote** 5.0.1  
**Imports** mcmcse  
**Suggests** knitr  
**VignetteBuilder** knitr  
**NeedsCompilation** no  
**Date/Publication** 2017-02-27 17:47:26  
**Repository** <https://abhirupkgp.r-universe.dev>  
**RemoteUrl** <https://github.com/cran/dirmcmc>  
**RemoteRef** HEAD  
**RemoteSha** c5f25d81bffd707991ca297119523c9bc6179911

## Contents

dirmcmc . . . . .	2
iact . . . . .	2
mcmcdiag . . . . .	3
metropdir . . . . .	5
metropdir.adapt . . . . .	6
msjd . . . . .	8

**Index****10**


---

dirmcmc	<i>dirmcmc: A package implementing Directional Metropolis Hastings for MCMC</i>
---------	---

---

**Description**

dirmcmc package provides functions for simulating from a target distribution with known log un-normalized density and its derivative. The derivative information is needed to construct the DMH kernel, which is a generalization of random walk Metropolis Hastings kernel.

**dirmcmc functions**

`metropdir`: Implements the dmh algorithm to simulate from a given density.

`metropdir.adapt`: Adaptive version of DMH algorithm.

`iact`: Integrated auto correlation times of a univariate chain.

`msjd`: Mean square jump distance of a multivariate chain.

`mcmcdiag`: Some summary of diagnostics of a given chain.

---

iact	<i>Integrated Auto correlation times of a Markov Chain</i>
------	--

---

**Description**

This function calculates the Integrated Auto Correlation Times of a Markov Chain.

**Usage**

```
iact(x)
```

**Arguments**

x                      chain (one dimension)

**Details**

The Integrated Auto Correlation Times of a Markov Chain X is defined as:

$$1 + 2 \sum \Gamma_i$$

, where

$$\Gamma$$

indicates the estimated autocorrelation terms of the chain. These are estimated using the sample correlation matrix from the lagged chain. This measure is intended for one dimensional chains or single component of a multivariate chains.

**Value**

Integrated ACT of the chain.

**Author(s)**

Abhirup Mallik, <malli066@umn.edu>

**See Also**

[msjd](#) for mean squared jumping distance, [mcmcdiag](#) for summary of diagnostic measures of a chain, [multiESS](#) for Multivariate effective sample size.

**Examples**

```
## Not run:
## Banana Target
lupost.banana <- function(x,B){
  -x[1]^2/200 - 1/2*(x[2]+B*x[1]^2-100*B)^2
}
Banana Gradient
gr.banana <- function(x,B){
  g1 <- -x[1]/100 - 2*B*(x[2]+B*x[1]^2-100*B)
  g2 <- -(x[2]+B*x[1]^2-100*B)
  g <- c(g1,g2)
  return(g)
}
out.metdir.banana <- metropdir(obj = lupost.banana, dobj = gr.banana,
initial = c(0,1),lchain = 2000,
sd.prop=1.25,
steplen=0.01,s=1.5,B=0.03)
iact(out.metdir.banana$batch[,1])

## End(Not run)
```

---

mcmcdiag

*mcmcdiag*

---

**Description**

This function calculates all different diagnostics supported in this library and returns in a list

**Usage**

```
mcmcdiag(X)
```

**Arguments**

X Chain (Matrix)

## Details

This function calculates four metrics useful for diagnostics of a Markov chain. The chain input could be univariate or multivariate. The univariate summaries are calculated marginally, for each component for a multivariate chains. Effective sample size is calculated for each component. Integrated auto correlation times is also another componentwise measure calculated for all the components. Multivariate Effective sample size is calculated from mcmcse package. Mean squared jump distance is another multivariate summary measure that is returned.

## Value

list with following elements:

- MEss Multivariate Effective sample size.
- ess vector of effective sample size for each component.
- iact vector of integrated autocorrelation times for each component.
- msjd Mean squared jump distance for the chain.

## Author(s)

Abhirup Mallik, <malli066@umn.edu>

## See Also

[iact](#) for integrated auto correlation times, [msjd](#) for mean squared jump distance of a chain, [multiESS](#) for Multivariate effective sample size.

## Examples

```
## Not run:
## Banana Target
lupost.banana <- function(x,B){
  -x[1]^2/200 - 1/2*(x[2]+B*x[1]^2-100*B)^2
}
Banana Gradient
gr.banana <- function(x,B){
  g1 <- -x[1]/100 - 2*B*(x[2]+B*x[1]^2-100*B)
  g2 <- -(x[2]+B*x[1]^2-100*B)
  g <- c(g1,g2)
  return(g)
}
out.metdir.banana <- metropdir(obj = lupost.banana, dobj = gr.banana,
initial = c(0,1),lchain = 2000,
sd.prop=1.25,
steplen=0.01,s=1.5,B=0.03)
mcmcdiag(out.metdir.banana$batch)

## End(Not run)
```

---

metropdir                      *Directional Metropolis Hastings*

---

**Description**

Implements Markov Chain Monte Carlo for continuous random vectors using Directional Metropolis Hasting Algorithm

**Usage**

```
metropdir(obj, dobj, initial, lchain, sd.prop = 1, steplen = 0, s = 0.95,
          ...)
```

**Arguments**

obj	an R function that evaluates the log unnormalized probability density of the desired equilibrium distribution of the Markov chain. First argument is the state vector of the Markov chain. Other arguments arbitrary and taken from the ... arguments of this function. Should return $-\text{Inf}$ for points of the state space having probability zero under the desired equilibrium distribution.
dobj	an R function that evaluates the derivative of the log unnormalized probability density at the current state of the markov chain.
initial	Initial state of the markov chain. <code>obj(state)</code> must not return $-\text{Inf}$
lchain	length of the chain
sd.prop	scale to use for the proposal
steplen	tuning parameter in mean of proposal
s	tuning parameter in the covariance of proposal
...	any arguments to be passed to obj and dobj.

**Details**

Runs a “Directional Metropolis Hastings” algorithm, with multivariate normal proposal producing a Markov chain with equilibrium distribution having a specified unnormalized density. Distribution must be continuous. Support of the distribution is the support of the density specified by argument obj.

**Value**

Returns the following objects in a list:

- `accept.` acceptance rate.
- `batch.` resulting chain.

**Author(s)**

Abhirup Mallik, <malli066@umn.edu>

**See Also**

[metropdir.adapt](#) for adapting DMH, [iact](#) for integrated auto correlation times, [mcmcdiag](#), [msjd](#) for mean squared jump distance. for summary of diagnostic measures of a chain, [multiESS](#) for Multivariate effective sample size.

**Examples**

```
## Not run:
Sigma <- matrix(c(1,0.2,0.2,1),2,2)
mu <- c(1,1)
Sig.Inv <- solve(Sigma)
Sig.det.sqrt <- sqrt(det(Sigma))
logf <- function(x,mu,Sig.Inv){
  x.center <- as.numeric((x-mu))
  out <- crossprod(x.center,Sig.Inv)
  out <- sum(out*x.center)
  -out/2
}

gr.logf <- function(x,mu,Sig.Inv){
  x.center <- as.numeric((x-mu))
  out <- crossprod(x.center,Sig.Inv)
  -as.numeric(out)
}

set.seed(1234)
system.time(out <- metropdir(obj = logf, dobj = gr.logf, initial = c(1,1),
                             lchain = 1e4,sd.prop=1,steplen=0,s=1, mu = mu,
                             Sig.Inv = Sig.Inv))

#acceptance rate
out$accept
#density plot
plot(density(out$batch[,1]))

## End(Not run)
```

---

metropdir.adapt

*Directional Metropolis Hastings with Adaptation.*


---

**Description**

Implements adaptive version of directional Metropolis Hastings.

**Usage**

```
metropdir.adapt(obj, dobj, initial, lchain, sd.prop = 1, steplen = 0,
  s = 0.95, batchlen = 100, targetacc = 0.234, ...)
```

**Arguments**

obj	an R function that evaluates the log unnormalized probability density of the desired equilibrium distribution of the Markov chain. First argument is the state vector of the Markov chain. Other arguments arbitrary and taken from the . . . arguments of this function. Should return $-\infty$ for points of the state space having probability zero under the desired equilibrium distribution.
dobj	an R function that evaluates the derivative of the log unnormalized probability density at the current state of the markov chain.
initial	Initial state of the markov chain. obj(state) must not return #' $-\infty$
lchain	length of the chain
sd.prop	scale to use for the proposal
steplen	tuning parameter in mean of proposal
s	tuning parameter in the covariance of proposal
batchlen	length of batch used for update. Default is 100.
targetacc	Target acceptance ratio
. . .	any arguments to be passed to obj and dobj.

**Details**

This function is for automatically select a scaling factor for the directional Metropolis Hastings algorithm. This function uses batch wise update of the scale parameter to produce a adaptive chain. The user is required to supply a target acceptance ratio. The adaptive scheme modifies the scale parameter to achieve the target acceptance ratio. It is recommended that to check the output of the adaptation history as returned by the function.

**Author(s)**

Abhirup Mallik, <malli066@umn.edu>

**See Also**

[metropdir](#) for DMH, [iact](#) for integrated auto correlation times, [mcmcdiag](#), [msjd](#) for mean squared jump distance. for summary of diagnostic measures of a chain, [multiESS](#) for Multivariate effective sample size.

**Examples**

```
## Not run:
Sigma <- matrix(c(1,0.2,0.2,1),2,2)
mu <- c(1,1)
Sig.Inv <- solve(Sigma)
Sig.det.sqrt <- sqrt(det(Sigma))
logf <- function(x,mu,Sig.Inv){
  x.center <- as.numeric((x-mu))
  out <- crossprod(x.center,Sig.Inv)
  out <- sum(out*x.center)
  -out/2
}
```

```
  }  
  
  gr.logf <- function(x,mu,Sig.Inv){  
    x.center <- as.numeric((x-mu))  
    out <- crossprod(x.center,Sig.Inv)  
    -as.numeric(out)  
  }  
  set.seed(1234)  
  system.time(out <- metropdir.adapt(obj = logf, dobj = gr.logf, initial = c(1,1),  
                                   lchain = 1e4,sd.prop=1,steplen=0,s=1, mu = mu,  
                                   Sig.Inv = Sig.Inv,targetacc=0.44))  
  
  #acceptance rate  
  out$accept  
  #density plot  
  plot(density(out$batch[,1]))  
  
  ## End(Not run)
```

---

msjd

*Mean Squared Jump Distance of a Markov Chain*

---

## Description

We calculate mean square euclidean jumping distance. The target covariance is unknown and the assumption of elliptical contour might not hold here, hence, we dont implement the variance scaled version. And this version is computationally faster as well.

## Usage

```
msjd(X)
```

## Arguments

X                    chain (Matrix) (in d dim)

## Details

Mean squared jump distance of a markov chain is a measure used to diagnose the mixing of the chain. It is calculated as the mean of the squared euclidean distance between every point and its previous point. Usually, this quantity indicates if the chain is moving enough or getting stuck at some region.

## Value

Mean squared jump distance of the chain.

## Author(s)

Abhirup Mallik, <malli066@umn.edu>



**See Also**

[iact](#) for integrated auto correlation times, [mcmcdiag](#) for summary of diagnostic measures of a chain, [multiESS](#) for Multivariate effective sample size.

**Examples**

```
## Not run:
## Banana Target
lupost.banana <- function(x,B){
  -x[1]^2/200 - 1/2*(x[2]+B*x[1]^2-100*B)^2
}
Banana Gradient
gr.banana <- function(x,B){
  g1 <- -x[1]/100 - 2*B*(x[2]+B*x[1]^2-100*B)
  g2 <- -(x[2]+B*x[1]^2-100*B)
  g <- c(g1,g2)
  return(g)
}
out.metdir.banana <- metropdir(obj = lupost.banana, dobj = gr.banana,
initial = c(0,1),lchain = 2000,
sd.prop=1.25,
steplen=0.01,s=1.5,B=0.03)
msjd(out.metdir.banana$batch)

## End(Not run)
```

# Index

- \* **adaptive**
  - metropdir.adapt, 6
- \* **dmh,**
  - metropdir, 5
  - metropdir.adapt, 6
- \* **dmh.**
  - iact, 2
  - mcmcdiag, 3
  - msjd, 8
- \* **iact,**
  - iact, 2
  - mcmcdiag, 3
  - msjd, 8
- \* **mcmc,**
  - iact, 2
  - mcmcdiag, 3
  - metropdir.adapt, 6
  - msjd, 8
- \* **mcmc**
  - metropdir, 5

dirmcmc, 2  
dirmcmc-package (dirmcmc), 2

iact, 2, 2, 4, 6, 7, 9

mcmcdiag, 2, 3, 3, 6, 7, 9  
metropdir, 2, 5, 7  
metropdir.adapt, 2, 6, 6  
msjd, 2–4, 6, 7, 8  
multiESS, 3, 4, 6, 7, 9