

# Package: adaptsmoFMRI (via r-universe)

September 7, 2024

**Type** Package

**Title** Adaptive Smoothing of FMRI Data

**Version** 1.2

**Date** 2022-09-24

**Description** Adaptive smoothing functions for estimating the blood oxygenation level dependent (BOLD) effect by using functional Magnetic Resonance Imaging (fMRI) data, based on adaptive Gauss Markov random fields, for real as well as simulated data. The implemented models make use of efficient Markov Chain Monte Carlo methods. Implemented methods are based on the research developed by A. Brezger, L. Fahrmeir, A. Hennerfeind (2007) <<https://www.jstor.org/stable/4626770>>.

**License** GPL-2

**Depends** R(>= 4.2.0)

**Collate** 'adaptive-gmrf-2covar.R' 'adaptive-gmrf.R'  
'plot-weights-real.R' 'plot-weights-sim.R'  
'sim-adaptive-gmrf-2covar.R' 'sim-adaptive-gmrf.R'  
'sim-fmri-2covar.R' 'sim-fmri.R'

**Imports** methods, stats, spatstat, spatstat.geom, Matrix, coda, mvtnorm, MCMCpack, parallel

**RoxygenNote** 7.2.1

**NeedsCompilation** no

**Author** Maximilian Hughes [aut, cre, ctb]

**Maintainer** Maximilian Hughes <hughesgm@me.com>

**Repository** CRAN

**Date/Publication** 2022-09-25 10:30:05 UTC

## Contents

adaptsmoFMRI-package . . . . .	2
adaptiveGMRF . . . . .	3

adaptiveGMRF2COVAR . . . . .	4
data_simfmri . . . . .	6
data_simfmri2COVAR . . . . .	6
sim.adaptiveGMRF . . . . .	7
sim.adaptiveGMRF2COVAR . . . . .	8
sim.fmri . . . . .	10
sim.fmri2COVAR . . . . .	11
sim.weightsplot . . . . .	12
weightsplot . . . . .	13
<b>Index</b>	<b>14</b>

---

adaptsmoFMRI-package    *Adaptive Smoothing of FMRI Data*

---

## Description

Adaptive smoothing functions for estimating the blood oxygenation level dependent (BOLD) effect by using functional Magnetic Resonance Imaging (fMRI) data, based on adaptive Gauss Markov random fields, for real as well as simulated data. The implemented models make use of efficient Markov Chain Monte Carlo methods.

## Details

Package: adaptsmoFMRI  
 Type: Package  
 Version: 1.1  
 Date: 2022-08-16  
 License: GPL-2

## Author(s)

Maximilian Hughes Maintainer: <hughesgm@me.com>

## References

- A. Brezger, L. Fahrmeir, A. Hennerfeind *Adaptive Gaussian Markov random fields with applications in human brain mapping*, Applied Statistics, **56**, Part 3, pp. 327-345, 2007
- V. J. Schmid, A. Thornton, B. Witcher *Working with the DICOM and NIFTI Data Standards in R*, Journal of Statistical Software, **Volume 44**, Issue 6. 2011

adaptiveGMRF

*Adaptive GMRF Model (Real Data)***Description**

This function estimates the effects of functional MR Images (fMRI), with the method of efficient Markov Chain Monte Carlo (MCMC) simulation. The Metropolis Hastings (MH) algorithm is used for the non-approximate case and the Gibbs sampler for the approximate case.

**Usage**

```
adaptiveGMRF(data, hrf, approximate = FALSE, K = 500, a =
  0.001, b = 0.001, c = 0.001, d = 0.001, nu = 1, filter
  = NULL, block = 1, burnin = 1, thin = 1)
```

**Arguments**

data	fMRI-data, needs to be an array of dimension (dx x dy x T).
hrf	haemodynamic response function, needs to be a vector of length T.
approximate	logical, if TRUE then the approximate case is choosen. Default is FALSE.
K	scalar, length of the MCMC path, hence iteration steps.
a	scalar, shape hyperparameter of the inverse-gamma distribution of the variance parameter ( $\sigma_i^2$ ).
b	scalar, scale hyperparameter of the inverse gamma distribution of the variance parameter ( $\sigma_i^2$ ).
c	scalar, shape hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
d	scalar, scale hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
filter	scalar, a value between 0 and 1 defining to which extent the fMRI-data should be filtered. The corresponding formular is $\max(\text{fMRI}) * \text{filter}$ .
nu	scalar, shape and scale hyperparameter of the gamma distribution of the interaction weights ( $w_{ij}$ ).
block	scalar, when approximate==TRUE then a block of weights is updated at a time.
burnin	scalar, defining the first iteration steps which should be omitted from MCMC path.
thin	scalar, only every thin step of MCMC path is saved to output.

**Value**

dx	scalar, number of pixels in x-direction.
dy	scalar, number of pixels in y-direction.
I	scalar, number of pixels.

coord	matrix, coordinates of pixels.
NEI	scalar, number of weights.
nei	matrix, locations of weights in precision matrix.
mask	matrix, masked out pixels.
beta.out	matrix, MCMC path of covariates.
w.out	matrix, MCMC path of weights.
sigma.out	matrix, MCMC path of variance parameters.
tau.out	matrix, MCMC path of hyper parameters.

**Note**

This function is solely for one covariate and real data sets.

**Author(s)**

Maximilian Hughes

**Examples**

```
# See example function for simulated data (one covariate).
```

---

adaptiveGMRF2COVAR      *Adaptive GMRF Model (Real Data)*

---

**Description**

This function estimates the effects of functional MR Images (fMRI), with the method of efficient Markov Chain Monte Carlo (MCMC) simulation. The Metropolis Hastings (MH) algorithm is used for the non-approximate case and the Gibbs sampler for the approximate case.

**Usage**

```
adaptiveGMRF2COVAR(data, hrf, approximate = FALSE, K =
  500, a = 0.001, b = 0.001, c = 0.001, d = 0.001, nu =
  1, filter = NULL, block = 1, burnin = 1, thin = 1)
```

**Arguments**

data	fMRI-data, needs to be an array of dimension (dx x dy x T).
hrf	haemodynamic response function, needs to be a vector of length T.
approximate	logical, if TRUE then the approximate case is chosen. Def#' ault is FALSE.
K	scalar, length of the MCMC path, hence iteration steps.
a	scalar, shape hyperparameter of the inverse-gamma distribution of the variance parameter ( $\sigma_i^2$ ).

b	scalar, scale hyperparameter of the inverse gamma distribution of the variance parameter ( $\sigma_i^2$ ).
c	scalar, shape hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
d	scalar, scale hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
filter	scalar, a value between 0 and 1 defining to which extent the fMRI-data should be filtered. The corresponding formular is $\max(\text{fmri}) * \text{filter}$ .
nu	scalar, shape and scale hyperparameter of the gamma distribution of the interaction weights ( $w_{ij}$ ).
block	scalar, when <code>approximate==TRUE</code> then a block of weights is updated at a time.
burnin	scalar, defining the first iteration steps which should be omitted from MCMC path.
thin	scalar, only every thin step of MCMC path is saved to output.

**Value**

dx	scalar, number of pixels in x-direction.
dy	scalar, number of pixels in y-direction.
I	scalar, number of pixels.
coord	matrix, coordinates of pixels.
NEI	scalar, number of weights.
nei	matrix, locations of weights in precision matrix.
mask	matrix, masked out pixels.
beta.out	matrix, MCMC path of covariates.
w.out	matrix, MCMC path of weights.
sigma.out	matrix, MCMC path of variance parameters.
tau.out	matrix, MCMC path of hyper parameters.

**Note**

This function is solely for two covariates and real data sets.

**Author(s)**

Maximilian Hughes

**Examples**

```
# See example function for simulated data (one covariate).
```

---

data_simfmri	<i>Simulated FMRI Data Set (for one covariate)</i>
--------------	--

---

**Description**

The data set contains a synthetic spatiotemporal data set, hence a simulated data set. The data set is an array of dimension 20 x 20 x 210.

**Value**

No return value, called for side effects

**Note**

This data set was created while inserting one covariate.

---

data_simfmri2COVAR	<i>Simulated FMRI Data Set (for two covariates)</i>
--------------------	---

---

**Description**

The data set contains a synthetic spatiotemporal data set, hence a simulated data set. The data set is an array of dimension 20 x 20 x 180.

**Value**

No return value, called for side effects

**Note**

This data set was created while inserting two covariates.

---

sim.adaptiveGMRF      *Adaptive GMRF Model for Simulated Data*

---

### Description

This function estimates the effects of a synthetic spatiotemporal data set resembling functional MR Images (fMRI), with the method of efficient Markov Chain Monte Carlo (MCMC) simulation. The Metropolis Hastings (MH) algorithm is used for the non-approximate case and the Gibbs sampler for the approximate case.

### Usage

```
sim.adaptiveGMRF(data, hrf, approximate = FALSE, K = 500,
  a = 1, b = 1, c = 1, d = 1, nu = 1, block = 1, burnin =
  1, thin = 1)
```

### Arguments

data	simulated fMRI-data, needs to be an array of dimension (20 x 20 x T).
hrf	haemodynamic response function, needs to be a vector of length T.
approximate	logical, if TRUE then the approximate case is chosen. Default is FALSE.
K	scalar, length of the MCMC path, hence iteration steps.
a	scalar, shape hyperparameter of the inverse-gamma distribution of the variance parameter ( $\sigma_i^2$ ).
b	scalar, scale hyperparameter of the inverse gamma distribution of the variance parameter ( $\sigma_i^2$ ).
c	scalar, shape hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
d	scalar, scale hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
nu	scalar, shape and scale hyperparameter of the gamma distribution of the interaction weights ( $w_{ij}$ ).
block	scalar, when approximate==TRUE then a block of weights is updated at a time.
burnin	scalar, defining the first iteration steps which should be omitted from MCMC path.
thin	scalar, only every thin step of MCMC path is saved to output.

### Value

dx	scalar, number of pixels in x-direction.
dy	scalar, number of pixels in y-direction.
I	scalar, number of pixels.
iter	scalar, number of MCMC iterations.

coord	matrix, coordinates of pixels.
nei	matrix, locations of weights in precision matrix.
NEI	scalar, number of weights.
beta.out	matrix, MCMC path of covariates.
w.out	matrix, MCMC path of weights.
sigma.out	matrix, MCMC path of variance parameters.
tau.out	matrix, MCMC path of hyper parameters.

**Note**

This function is solely for one covariate.

**Author(s)**

Maximilian Hughes

**Examples**

```
# non-transformed hr-function
T <- 210
seq.length <- T*3
index <- seq(3, T*3, by = 3)
hrf <- rep(c(-0.5, 0.5), each=30, times=ceiling(T/30*1.5))
hrf <- as.matrix(hrf[index])
# get simulated data
data("sim_fmri")
data <- data_simfmri
# execute function
set.seed(111222)
K <- 2
a <- b <- c <- d <- nu <- 1
test.sim.adaptive <- sim.adaptiveGMRF(data, hrf, approximate=TRUE, K,
                                     a, b, c, d, nu)
```

---

sim.adaptiveGMRF2COVAR

*Adaptive GMRF Model for Simulated Data*

---

**Description**

This function estimates the effects of a synthetic spatiotemporal data set resembling functional MR Images (fMRI), with the method of efficient Markov Chain Monte Carlo (MCMC) simulation. The Metropolis Hastings (MH) algorithm is used for the non-approximate case and the Gibbs sampler for the approximate case.

**Usage**

```
sim.adaptiveGMRF2COVAR(data, hrf, approximate = FALSE, K
= 500, a = 1, b = 1, c = 1, d = 1, nu = 1, block = 1,
burnin = 1, thin = 1)
```

**Arguments**

data	simulated fMRI-data, needs to be an array of dimension (20 x 20 x T).
hrf	haemodynamic response function, needs to be a vector of length T.
approximate	logical, if TRUE then the approximate case is chosen. Default is FALSE.
K	scalar, length of the MCMC path, hence iteration steps.
a	scalar, shape hyperparameter of the inverse-gamma distribution of the variance parameter ( $\sigma_i^2$ ).
b	scalar, scale hyperparameter of the inverse gamma distribution of the variance parameter ( $\sigma_i^2$ ).
c	scalar, shape hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
d	scalar, scale hyperparameter of the inverse gamma distribution of the precision parameter ( $\tau$ ).
nu	scalar, shape and scale hyperparameter of the gamma distribution of the interaction weights ( $w_{ij}$ ).
block	scalar, when approximate==TRUE then a block of weights is updated at a time.
burnin	scalar, defining the first iteration steps which should be omitted from MCMC path.
thin	scalar, only every thin step of MCMC path is saved to output.

**Value**

dx	scalar, number of pixels in x-direction.
dy	scalar, number of pixels in y-direction.
I	scalar, number of pixels.
iter	scalar, number of MCMC iterations.
coord	matrix, coordinates of pixels.
nei	matrix, locations of weights in precision matrix.
NEI	scalar, number of weights.
beta.out	matrix, MCMC path of covariates.
w.out	matrix, MCMC path of weights.
sigma.out	matrix, MCMC path of variance parameters.
tauk.out	matrix, MCMC path of hyper parameters.

**Note**

This function is solely for two covariates.

**Author(s)**

Maximilian Hughes

**Examples**

```
# See example function for simulated data (one covariate).
```

---

sim.fmri	<i>Simulate FMRI Data</i>
----------	---------------------------

---

**Description**

This function returns the synthetic spatiotemporal data set resembling functional MR Images (fMRI) data.

**Usage**

```
sim.fmri(hrf, beta)
```

**Arguments**

hrf	haemodynamic response function, needs to be a vector of length T.
beta	scalar, defines the height of the activated area, in form of a cylinder.

**Details**

The returned data is simulated on a 20 x 20 grid.

**Value**

fmri	matrix, simulated fmri data.
------	------------------------------

**Note**

This function is solely for one covariate.

**Author(s)**

Maximilian Hughes

## Examples

```
# non-transformed hr-function
T <- 210
seq.length <- T*3
index <- seq(3, T*3, by = 3)
hrf <- rep(c(-0.5, 0.5), each=30, times=ceiling(T/30*1.5))
hrf <- as.matrix(hrf[index])
# define height of activation area
beta <- 3
# use function to obtain fmri data
data <- sim.fmri(hrf, beta)$fmri
```

---

sim.fmri2COVAR	<i>Simulate FMRI Data</i>
----------------	---------------------------

---

## Description

This function returns the synthetic spatiotemporal data set resembling functional MR Images (fMRI) data.

## Usage

```
sim.fmri2COVAR(hrf, beta.Var1, beta.Var2)
```

## Arguments

hrf	haemodynamic response function, needs to be a vector of length T.
beta.Var1	scalar, defines the height of the activated area, in form of a cylinder of the first grid.
beta.Var2	scalar, defines the height of the activated area, in form of a cylinder of the second grid.

## Details

The returned data is simulated on a 20 x 20 grid.

## Value

fmri	matrix, simulated fmri data.
------	------------------------------

## Note

This function is solely for two covariates.

## Author(s)

Maximilian Hughes

**Examples**

```

# non-transformed hr-function
T <- 180
seq.length <- T*3
index <- seq(3, T*3, by = 3)
vis <- rep(c(-0.5, 0.5), each=30, times=ceiling(T/30*1.5))
vis <- as.matrix(vis[index])
aud <- rep(c(-0.5, 0.5), each=45, times=ceiling(T/30*1.5))
aud <- as.matrix(aud[index])
hrf <- cbind(vis,aud)
# define height of activation area
beta.Var1 <- beta.Var2 <- 3
# use function to obtain fmri data
data <- sim.fmri2COVAR(hrf, beta.Var1, beta.Var2)$fmri

```

---

sim.weightsplot

*Plot Function for Weights (Simulated Data)*


---

**Description**

This function plots the weights interacting between estimated effects for each pixel.

**Usage**

```
sim.weightsplot(weights, nei, nx, ny, thresh = 0.05, ...)
```

**Arguments**

weights	matrix, containing MCMC-output the of posterior estimates of weights.
nei	matrix, locations of weights in precision matrix.
nx	scalar, number of pixels in x-direction.
ny	scalar, number of pixels in y-direction.
thresh	scalar, defining the threshold to which the median of the weights smaller than this threshold should be plotted.
...	graphical parameters for image can also be passed on as arguments to this function.

**Value**

No return value, called for side effects

**Note**

This function is solely for MCMC-outputs on simulated data.

**Author(s)**

Maximilian Hughes

---

`weightsplot`*Plot Function for Weights (Real Data)*

---

**Description**

This function plots the weights interacting between estimated effects for each pixel.

**Usage**

```
weightsplot(weights, nei, nx, ny, coord, thresh = 0.1,  
            ...)
```

**Arguments**

<code>weights</code>	matrix, containing MCMC-output the of posterior estimates of weights.
<code>nei</code>	matrix, locations of weights in precision matrix.
<code>nx</code>	scalar, number of pixels in x-direction.
<code>ny</code>	scalar, number of pixels in y-direction.
<code>coord</code>	matrix, coordinates of pixels.
<code>thresh</code>	scalar, defining the threshold to which the median of the weights smaller than this threshold should be plotted.
<code>...</code>	graphical parameters for image can also be passed on as arguments to this function.

**Value**

No return value, called for side effects

**Note**

This function is solely for MCMC-outputs on real data.

**Author(s)**

Maximilian Hughes

# Index

## \* package

adaptsmoFMRI-package, [2](#)

adaptiveGMRF, [3](#)

adaptiveGMRF2COVAR, [4](#)

adaptsmoFMRI (adaptsmoFMRI-package), [2](#)

adaptsmoFMRI-package, [2](#)

data\_simfmri, [6](#)

data\_simfmri2COVAR, [6](#)

sim.adaptiveGMRF, [7](#)

sim.adaptiveGMRF2COVAR, [8](#)

sim.fmri, [10](#)

sim.fmri2COVAR, [11](#)

sim.weightsplot, [12](#)

weightsplot, [13](#)