

Package ‘neuRosim’

July 22, 2025

Type Package

Title Simulate fMRI Data

Version 0.2-14

Date 2023-10-18

Depends R (>= 3.1.1), deSolve

Description Generates functional Magnetic Resonance Imaging (fMRI) time series or 4D data. Some high-level functions are created for fast data generation with only a few arguments and a diversity of functions to define activation and noise. For more advanced users it is possible to use the low-level functions and manipulate the arguments. See Welvaert et al. (2011) <[doi:10.18637/jss.v044.i10](https://doi.org/10.18637/jss.v044.i10)>.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2023-10-18 12:50:02 UTC

NeedsCompilation yes

Author Marijke Welvaert [aut],
Joke Durnez [ctb],
Beatrijs Moerkerke [ctb],
Yves Rosseel [ctb],
Karsten Tabelow [ctb, cre],
Geert Verdoolaege [ctb]

Maintainer Karsten Tabelow <karsten.tabelow@wias-berlin.de>

Contents

neuRosim-package	2
balloon	3
canonicalHRF	4
gammaHRF	5
GaussSmoothKernel	6

lowfreqdrift	7
physnoise	8
rrice	9
Sim.3D.GRF	10
simplprepSpatial	11
simplprepTemporal	12
simTSfmri	13
simTSrestingstate	14
simVOLfmri	16
spatialnoise	18
specifydesign	19
specifyregion	20
stimfunction	21
systemnoise	22
tasknoise	23
temporalnoise	24
Index	26

neuRosim-package	<i>Functions to Generate fMRI Data Including Activated Data, Noise Data and Resting State Data</i>
------------------	--

Description

The package allows users to generate fMRI time series or 4D data. Some high-level functions are created for fast data generation with only a few arguments and a diversity of functions to define activation and noise. For more advanced users it is possible to use the low-level functions and manipulate the arguments.

Author(s)

Marijke Welvaert with contributions from Joke Durnez, Beatrijs Moerkerke, Yves Rosseel, Karsten Tabelow, and Geert Verdoolaege

Maintainer: Karsten Tabelow <karsten.tabelow@wias-berlin.de>

References

Welvaert, M., Durnez, J., Moerkerke, B., Verdoolaege, G. and Rosseel, Y. (2011). **neuRosim**: An R Package for Generating fMRI Data. *Journal of Statistical Software*, 44(10), 1–18

Examples

```
## Generate fMRI time series for block design
design <- simplprepTemporal(totaltime=200, onsets=seq(1,200,40),
  durations=20, TR=2, effectsize=1, hrf="double-gamma")
ts <- simTSfmri(design=design, SNR=1, noise="white")
plot(ts, type="l")
```

```
## Generate fMRI slice for block design with activation in 2 regions
design <- simprepTemporal(totaltime=200, onsets=seq(1,200,40),
  durations=20, TR=2, effectsize=1, hrf="double-gamma")
region <- simprepSpatial(regions=2, coord=list(c(32,15),c(57,45)),
  radius=c(10,7), form="sphere")
out <- simVOLfmri(design=design, image=region, dim=c(64,64),
  SNR=1, noise="none")
plot(out[32,15,], type="l")
```

balloon

Balloon model

Description

Generates the BOLD signal based on the Balloon model of Buxton et al. (2004).

Usage

```
balloon(stim, totaltime, acc, par=list(), verbose=TRUE)
```

Arguments

stim	Vector representing the presence/absence (1-0 coding) of a stimulus/activation in seconds.
totaltime	Total duration of stimulus vector in seconds.
acc	Microtime resolution of stimulus vector in seconds.
par	List representing the parameters of the Balloon model. The list should contain the following: kappa Inhibitory gain factor tau1 Inhibitory time constant tauf FWHM of CBF impulse response taum FWHM of CMRO2 impulse response deltat Delay of CBF relative to CMRO2 response n Steady-state flow metabolism relation f1 Normalized CBF response to sustained neural activation tauMTT Transit time through the balloon tau Viscoelastic time constant alpha Steady-state flow-volume relation E0 baseline O2 extraction fraction V0 baseline blood volume a1 weight for deoxyHb change a2 weight for blood volume change
verbose	If TRUE, warnings are displayed.

Details

Based on the provided stimulus boxcar function, a neural activation function is generated that enters the Balloon model to generate a BOLD response. The microtime resolution ensures a high-precision generation of the response. More details can be found in Buxton et al. (2004).

Value

Vector representing the values of the BOLD signal for the given stimulus vector and microtime resolution.

Author(s)

G. Verdoolaeghe, M. Welsaert

References

Buxton, RB, Uludag, K, Dubowitz, DJ and Liu, TT (2004). Modeling the hemodynamic response to brain activation. NeuroImage, 23, S220-S233.

See Also

[canonicalHRF](#), [gammaHRF](#)

Examples

```
s <- rep(rep(0,10), rep(1,10), 5)
T <- 100
it <- 0.1
out <- balloon(s, T, it)
#takes a couple of seconds due to solving of the differential equations
```

canonicalHRF

Double-gamma Haemodynamic response function

Description

Specifies a double-gamma variate haemodynamic response function for the given time vector and parameters.

Usage

```
canonicalHRF(x, param = NULL, verbose = TRUE)
```

Arguments

x	Time vector in seconds.
param	List of parameters of the haemodynamic response function. The list should contain the following: a1 Delay of response relative to onset (default: 6) a2 Delay of undershoot relative to onset (default:12) b1 Dispersion of response (default:0.9) b2 Dispersion of undershoot (default:0.9) c Scale of undershoot (default:0.35)
verbose	If TRUE, warnings are displayed.

Value

Vector representing the values of the function for the given time vector and parameters.

Author(s)

M. Welvaert

References

- [1] Friston, KJ, Fletcher, P, Josephs, O, Holmes, AP, Rugg, MD and Turner, R (1998). Event-related fMRI: Characterising differential responses. *NeuroImage*, 7, 30-40.
- [2] Glover, GH (1999). Deconvolution of impulse response in event-related BOLD fMRI. *NeuroImage*, 9, 416-429.

See Also

[gammaHRF](#), [balloon](#)

Examples

```
t <- 1:100
out <- canonicalHRF(t, verbose=FALSE)
```

gammaHRF

Single Gamma Haemodynamic response function.

Description

Specifies a Gamma variate haemodynamic response function for the given time vector and FWHM.

Usage

```
gammaHRF(x, FWHM = 4, verbose = TRUE)
```

Arguments

x	Time vector in seconds.
FWHM	Full Width Half Maximum of the Gamma variate function.
verbose	If TRUE, warnings are displayed.

Value

Vector representing the values of the function for the given time vector and FWHM.

Author(s)

M. Welsaert

References

Buxton, RB, Uludag, K, Dubowitz, DJ and Liu, TT (2004). Modeling the hemodynamic response to brain activation. *NeuroImage*, 23, S220-S233.

See Also

[canonicalHRF](#), [balloon](#)

Examples

```
t <- 1:100
out <- gammaHRF(t, verbose=FALSE)
```

GaussSmoothKernel	<i>Calculates a discrete Gaussian smoothing kernel (adopted from AnalyzeFMRI)</i>
-------------------	--

Description

Calculates a simple, discrete Gaussian smoothing kernel of a specific size given the covariance matrix of the Gaussian.

Usage

```
GaussSmoothKernel(voxdim=c(1,1,1), ksize=5, sigma=diag(3,3))
```

Arguments

voxdim	The dimensions of each voxel.
ksize	The size (in voxels) of the kernel with which to filter the independent field.
sigma	The covariance matrix of the Gaussian kernel.

Value

An array of dimension (ksize,ksize,ksize) containing the smoothing kernel.

Author(s)

J. L. Marchini

See Also

[Sim.3D.GRF](#)

Examples

```
a <- GaussSmoothKernel(voxdim=c(1,1,1), ksize=5, sigma=diag(1,3))
```

lowfreqdrift

Generate low frequency drift

Description

Generates a low-frequency drift dataset with specified dimensions and frequency.

Usage

```
lowfreqdrift(dim, freq = 128, nscan, TR, template, verbose = TRUE)
```

Arguments

dim	A vector specifying the dimensions of the image.
freq	The frequency of the drift in seconds.
nscan	The number of scans in the dataset.
TR	The repetition time in seconds.
template	An array representing the anatomical structure or mask with dimensions equal to dim.
verbose	Logical indicating if warnings should be printed.

Details

The function generates low-frequency drift based on a basis set of cosine functions. The result is an array with specified dimensions and frequency.

Value

An array containing the drift with dimensions specified in dim.

Author(s)

Y. Rosseel, M. Welvaert

References

Friston et al. (2007). Statistical Parametric Mapping: The analysis of functional brain images. Academic Press.

See Also

[temporalnoise](#), [systemnoise](#), [physnoise](#), [tasknoise](#), [spatialnoise](#)

Examples

```
d <- c(10,10,10)
freq <- 80
nscan <- 100
TR <- 2
out <- lowfreqdrift(d, freq, nscan, TR, verbose=FALSE)
```

physnoise

Generate physiological noise

Description

Generates a physiological noise dataset with specified dimensions and standard deviation. The physiological noise is defined as noise caused by heart beat and respiratory rate.

Usage

```
physnoise(dim, nscan, TR, sigma, freq.heart = 1.17,
  freq.resp = 0.2, template, verbose = TRUE)
```

Arguments

dim	A vector specifying the dimensions of the image.
nscan	The number of scans in the dataset.
TR	The repetition time in seconds.
sigma	The standard deviation of the noise.
freq.heart	The frequency in Hz of the heart beat.
freq.resp	The frequency in Hz of the respiratory rate.
template	An array representing the anatomical structure or mask with dimensions equal to dim.
verbose	Logical indicating if warnings should be printed.

Details

The function generates physiological noise. Heart beat and respiratory rate are defined as sine and cosine functions with specified frequencies. Additional Gaussian noise creates variability over voxels. The result is a noise dataset with specified dimensions and desired standard deviation.

Value

An array containing the noise with dimensions specified in `dim` and `nscan`.

Author(s)

M. Welvaert

See Also

[temporalnoise](#), [lowfreqdrift](#), [systemnoise](#), [tasknoise](#), [spatialnoise](#)

Examples

```
d <- c(10,10,10)
sigma <- 5
nscan <- 100
TR <- 2
out <- physnoise(d, nscan, TR, sigma, verbose=FALSE)
```

rrice

The Rice Distribution

Description

Density and random generation for the Rician distribution

Usage

```
rrice(n, vee, sigma)
```

Arguments

<code>n</code>	number of observations. Must be a positive integer of length 1.
<code>vee</code>	non-centrality parameter of the distribution. Must be a positive integer of length 1.
<code>sigma</code>	scale parameter of the distribution. Must be a positive integer of length 1.

Details

See **VGAM** for more details on the parameters and the formula of the probability density function.

Value

Random deviates for the given number of observations.

Author(s)

T.W. Yee

Examples

```
x <- rrice(n=10,vee=2,sigma=1)
```

Sim.3D.GRF

*Simulate a GRF (adopted from **AnalyzeFMRI**)*

Description

Simulates a Gaussian Random Field with specified dimensions and covariance structure.

Usage

```
Sim.3D.GRF(d, voxdim, sigma, ksize, mask=NULL, type=c("field","max"))
```

Arguments

d	A vector specifying the dimensions of a 3D or 4D array.
voxdim	The dimensions of each voxel.
sigma	The 3D covariance matrix of the field.
ksize	The size (in voxels) of the kernel with which to filter the independent field.
mask	A 3D mask for the field.
type	If type=="field" then the simulated field together with the maximum of the field is returned.If type=="max" then the maximum of the field is returned.

Details

The function works by simulating a Gaussian r.v at each voxel location and the smoothing the field with a discrete filter to obtain a field with the desired covariance structure.

Value

mat	Contains the simulated field if type=="field", else NULL
max	The maximum value of the simulated field

Author(s)

J. L. Marchini

See Also[GaussSmoothKernel](#)**Examples**

```
d <- c(64, 64, 21)
FWHM <- 9
sigma <- diag(FWHM^2, 3) / (8 * log(2))
voxdim <- c(2, 2, 4)
msk <- array(1, dim = d)
field <- Sim.3D.GRF(d = d, voxdim = voxdim, sigma = sigma,
ksize = 9, mask = msk, type = "max")
```

simprepSpatial	<i>Prepare spatial structure of the data</i>
----------------	--

Description

Prepare a list defining the necessary parameters to specify the spatial structure of the activation data.

Usage

```
simprepSpatial(regions, coord, radius = NULL,
form = c("cube", "sphere", "manual"), fading = 0)
```

Arguments

regions	Number of activated regions.
coord	List of coordinates specifying the xyz-coordinates.
radius	If form=cube or sphere, the distance between the center and the edge, if form>manual, the number of voxels in each region.
form	The form of the activated regions.
fading	Decay rate between 0 and 1. 0 means no fading, while 1 results in the fastest decay.

Value

A list with the necessary arguments to be used in [simVOLfmri](#).

Author(s)

M. Welsaert

See Also

[simVOLfmri](#), [simprepTemporal](#), [specifyregion](#)

Examples

```
coord <- list(c(3,3,3),c(6,6,6))
radius <- c(1,2)
out <- simprepSpatial(2, coord, radius, form="cube", fading=0.2)
```

simprepTemporal	<i>Prepare temporal structure of the data</i>
-----------------	---

Description

Prepare a list defining the necessary parameters to specify the temporal structure of the activation data.

Usage

```
simprepTemporal(totaltime, regions = NULL, onsets, durations,
TR, effectsize, accuracy=0.1,
hrf = c("gamma", "double-gamma", "Balloon"),
param = NULL)
```

Arguments

totaltime	Duration of the experiment.
regions	Number of regions. If not specified, it is assumed that all regions have the same design matrix.
onsets	List or vector representing the onsets of the stimulus in seconds.
durations	List or vector representing the durations of the stimulus in seconds.
TR	Repetition time in seconds.
effectsize	List or number representing the effectsize in each condition.
accuracy	Microtime resolution in seconds.
hrf	Haemodynamic response function (double-gamma is default)
param	Vector, matrix or array representing the parameters of the haemodynamic response function.

Value

A list with the necessary arguments to be used in [simVOLfmri](#) or [simTSfmri](#).

Author(s)

M. Welvaert

See Also

[simVOLfmri](#), [simTSfmri](#), [simprepSpatial](#), [specifyregion](#)

Examples

```
ncond <- 2
os <- list(c(20,60),c(15,35))
d <- list(20, 10)
effect <- list(7,10)
total <- 80
TR <- 2
out <- simprepTemporal(total, onsets=os, durations=d, TR=TR,
effectsize=effect, hrf="double-gamma")
```

simTSfmri

Simulate fMRI time series

Description

Simulates an fMRI time series for the specified design and noise type.

Usage

```
simTSfmri(design = list(), base=0, nscan = NULL, TR = NULL, SNR=NULL,
noise = c("none", "white", "temporal", "low-frequency",
"physiological", "task-related", "mixture"), type = c("gaussian", "rician"),
weights, verbose = TRUE, rho = 0.2, freq.low = 128, freq.heart = 1.17,
freq.resp = 0.2, vee=1)
```

Arguments

design	List generated by simprepTemporal specifying the design. If not specified, noise time series are generated.
base	Baseline value of the time series.
nscan	Number of scans.
TR	Repetition time in seconds.
SNR	Signal-to-noise ratio of the time series.
noise	Type of noise (white is default).
type	If noise==white, noise==task-related or noise==mixture, type of system noise (gaussian is default).
weights	If noise==mixture, vector of weights with 5 elements to specify the fraction of the noise components.
verbose	Logical indicating if warnings should be returned.
rho	If noise==temporal or noise==mixture, value of autocorrelation coefficients. The length of the vector indicates the order of the autoregressive model.
freq.low	If noise==low-frequency or noise==mixture, frequency of the low-frequency drift in seconds.

freq.heart	If noise==physiological or noise==mixture, frequency of heart rate in Hz.
freq.resp	If noise==physiological or noise==mixture, frequency of respiratory rate in Hz.
vee	If type=="rician", non-centrality parameter of the distribution.

Value

A vector representing the fMRI time series.

Author(s)

M. Welvaert

See Also

[simVOLfmri](#), [simprepTemporal](#)

Examples

```
design <- simprepTemporal(totaltime=200, onsets=seq(1,200,40),
  durations=20, effectsize=1, TR=2, hrf="double-gamma")
ts <- simTSfmri(design=design, SNR=1, noise="white")
plot(ts, type="l")
```

simTSrestingstate	<i>Simulate fMRI resting state time series</i>
-------------------	--

Description

Synthesizes a single time series x representing resting state activity. The fluctuation frequencies f are limited to a square passband $0.01 \text{ Hz} \leq f \leq 0.1 \text{ Hz}$. TR is the repetition time (needed to compute the passband limits), expressed in seconds. N is the required number of samples (needs not be a power of 2).

Usage

```
simTSrestingstate(nscan, base=0, TR, SNR=NULL, noise = c("none", "white",
  "temporal", "low-frequency", "physiological", "mixture"),
  type = c("gaussian", "rician"), weights, verbose = TRUE, rho = 0.2,
  freq.low = 128, freq.heart = 1.17, freq.resp = 0.2, vee=1)
```

Arguments

nscan	Number of scans.
base	Baseline value of the time series.
TR	Repetition time in seconds.
SNR	Signal-to-noise ratio of the time series.
noise	Type of noise (white is default).
type	If noise==white, noise==mixture, type of system noise (gaussian is default).
weights	If noise==mixture, vector of weights to specify the fraction of the noise components.
verbose	Logical indicating if warnings should be returned.
rho	If noise==temporal or noise==mixture, value of autocorrelation coefficients. The length of the vector corresponds to the order of the autoregressive model.
freq.low	If noise==low-frequency or noise==mixture, frequency of the low-frequency drift in seconds.
freq.heart	If noise==physiological or noise==mixture, frequency of heart rate in Hz.
freq.resp	If noise==physiological or noise==mixture, frequency of respiratory rate in Hz.
vee	If type==rician, non-centrality parameter of the distribution.

Value

A vector representing the resting state time series

Author(s)

J. Durnez, G. Verdoolaege, M. Welvaert

References

- [1] C.G. Fox, Computers & Geoscience, Vol. 13, pp. 369-374, 1987.
- [2] M. Fukunaga, Magnetic Resonance Imaging, Vol. 24, pp. 979-992, 2006.

See Also

[simTSfmri](#)

Examples

```
out <- simTSrestingstate(nscan=50, TR=2, SNR=1, noise="none")
plot(out, type="l")
```

simVOLfmri

*Simulate 3D or 4D fMRI data***Description**

Simulates a 3D or 4D fMRI dataset for the specified design and with activation in the specified regions.

Usage

```
simVOLfmri(design = list(), image = list(), base=0, dim, nscan = NULL,
  TR = NULL, SNR=NULL, noise = c("none", "white", "temporal",
  "spatial", "low-frequency", "physiological", "task-related",
  "mixture"), type = c("gaussian", "rician"),
  spat = c("corr", "gaussRF", "gammaRF"), weights, verbose = TRUE,
  rho.temp = 0.2, rho.spat = 0.75, freq.low = 128,
  freq.heart = 1.17, freq.resp = 0.2, FWHM = 4, gamma.shape = 6,
  gamma.rate = 1, vee=1, template)
```

Arguments

design	List generated by simplprepTemporal specifying the design. If not specified, noise images are generated.
image	List generated by simplprepSpatial specifying the activated regions. If not specified, noise images are generated
base	Baseline of the data. Should be a single number or an array with the same dimensions as in dim.
dim	Dimensions of the image space.
nscan	Number of scans for noise images.
TR	Repetition time for noise images.
SNR	Signal-to-noise ratio.
noise	Type of noise, default is white.
type	If noise==white or noise==mixture, the type of system noise (default is gaussian).
spat	If noise==spatial or noise==mixture, the spatial correlation structure (default is corr).
weights	If noise==mixture, weights vector of six elements.
verbose	Logical indicating if warning should be printed.
rho.temp	If noise==temporal or noise==mixture, value of autocorrelation coefficients. The length of the vector indicates the order of the autoregressive model.
rho.spat	If noise==spatial or noise==mixture, and spat==corr, value of the correlation coefficient.

freq.low	If noise==low-frequency or noise==mixture, frequency of the low-frequency drift in seconds.
freq.heart	If noise==physiological or noise==mixture, frequency of heart rate in Hz.
freq.resp	If noise==physiological or noise==mixture, frequency of respiratory rate in Hz.
FWHM	If noise==spatial or noise==mixture, and spat==gaussRF or spat==gammaRF, value of the FWHM of the Gaussian kernel.
gamma.shape	If noise==spatial or noise==mixture, and spat==gammaRF, value of the shape parameter of the gamma distribution.
gamma.rate	If noise==spatial or noise==mixture, and spat==gammaRF, value of the rate parameter of the gamma distribution.
vee	If type==rician, non-centrality parameter of the rician distribution.
template	An array representing the anatomical structure or mask with dimensions equal to dim.

Value

A 3D or 4D array specifying the values for each voxel in the data.

Author(s)

M. Welvaert

See Also

[simTSfmri](#), [simgrepTemporal](#), [simgrepSpatial](#)

Examples

```
design <- simprepTemporal(totaltime=200, onsets=seq(1,200,40),
  durations=20, effects=1, hrf="double-gamma")
region <- simprepSpatial(regions=2, coord=list(c(32,15),c(57,45)),
  radius=c(10,7), form="sphere", fading=TRUE)
out <- simVOLfmri(design=design, image=region, dim=c(64,64),
  SNR=1, noise="none")
plot(out[32,15,], type="l")
image(1:64, 1:64, out[, ,10], col = grey(0:255/255))
```

spatialnoise

*Generate spatially correlated noise***Description**

Generates a spatially correlated noise dataset with specified dimensions and standard deviation.

Usage

```
spatialnoise(dim, sigma, nscan, method = c("corr", "gammaRF", "gaussRF"),
             type=c("gaussian","rician"), rho = 0.75, FWHM = 4, gamma.shape = 6,
             gamma.rate = 1, vee=1, template, verbose = TRUE)
```

Arguments

dim	A vector specifying the dimensions of the image.
sigma	The standard deviation of the noise.
nscan	The number of scans in the dataset.
method	Method specifying the type of spatial correlation. Default is "corr".
type	Type of distribution if method=="corr". Default is "gaussian"
rho	If method=="corr", the value of the autocorrelation coefficient.
FWHM	If method=="gammaRF" or method=="gaussRF", the full-width-half-maximum of the Gaussian kernel.
gamma.shape	If method=="gammaRF", the shape parameter of the Gamma distribution.
gamma.rate	If method=="gammaRF", the shape parameter of the Gamma distribution.
vee	If method=="corr" and type=="rician", the non-centrality parameter of the rician distribution.
template	An array representing the anatomical structure or mask with dimensions equal to dim.
verbose	Logical indicating if warnings should be printed.

Details

The function generates spatially correlated noise. When method=="corr", AR(1) voxelwise correlations are introduced. If method=="gaussRF" or method=="gammaRF", respectively a Gaussian Random Field or a Gamma Random Field is created. The result is a noise array with specified dimensions and desired standard deviation. The generation of the random fields is based on the function Sim.3D.GRF from J.L. Marchini in the package **AnalyzeFMRI**.

Value

An array containing the noise with dimensions specified in dim and nscan.

Author(s)

J. Durnez, B. Moerkerke, M. Waelvaert

See Also

[temporalnoise](#), [lowfreqdrift](#), [physnoise](#), [tasknoise](#), [systemnoise](#), [Sim.3D.GRF](#)

Examples

```
d <- c(10,10,10)
sigma <- 5
nscan <- 100
rhospat <- 0.7
out <- spatialnoise(d, sigma, nscan, method="corr", rho=rhospat, verbose=FALSE)
```

specifydesign	<i>Generate design matrix.</i>
---------------	--------------------------------

Description

Generates a design matrix to be used as a model for the simulated activation.

Usage

```
specifydesign(onsets, durations, totaltime, TR, effectsize, accuracy=0.1,
conv = c("none", "gamma", "double-gamma", "Balloon"),
cond.names = NULL, param = NULL)
```

Arguments

onsets	List or vector representing the onsets in seconds.
durations	List or vector representing the durations in seconds.
totaltime	Duration of the experiment in seconds.
TR	Repetition time in seconds.
effectsize	List or number representing the effectsize in each condition.
accuracy	Microtime resolution in seconds.
conv	Should the design matrix be convoluted, default is none.
cond.names	Optional names for the conditions.
param	Parameters of the haemodynamic response function. See gammaHRF and canonicalHRF for more details.

Value

A matrix specifying the design.

Author(s)

M. Welsaert

See Also[specifyregion](#), [gammaHRF](#), [canonicalHRF](#), [balloon](#)**Examples**

```

os <- list(c(20,60),c(15,35))
d <- list(20, 10)
total <- 80
TR <- 2
out <- specifydesign(os, d, total, TR, effectsize=list(2,5), conv="double-gamma")

```

specifyregion

*Generate activation image***Description**

Generates an image with activated regions for specified dimensions. The regions are defined by their center and radius or can be entered manually.

Usage

```

specifyregion(dim, coord, radius = NULL,
form = c("cube", "sphere", "manual"),
fading = 0)

```

Arguments

dim	Dimensions of the image space.
coord	Coordinates of the activated region, if form=="cube" or form=="sphere", the coordinates represent the center of the region, if form=="manual", the coordinates should be in matrix form, where the rows represent the voxels and the columns the x-y-z coordinates.
radius	If form=="cube" or form=="sphere", the distance in voxels from the center of the region to the edge.
form	The form of the activated region. Default is "cube".
fading	Decay rate between 0 and 1. 0 means no fading, while 1 results in the fastest decay.

Value

An array representing the activation image with specified regions.

Author(s)

M. Welvaert

See Also[specifyregion](#), [gammaHRF](#), [canonicalHRF](#), [balloon](#)**Examples**

```
d <- c(10,10,10)
coord <- c(3,3,3)
radius <- 1
out <- specifyregion(d, coord, radius, form="sphere")
```

stimfunction

*Generate a stimulus boxcar function.***Description**

Generates a stimulus boxcar vector for the specified time duration and microtime resolution based on the user-defined onsets and durations.

Usage

```
stimfunction(totaltime, onsets, durations, accuracy)
```

Arguments

totaltime	Total time of the design in seconds.
onsets	Vector representing the onsets of the stimulus in seconds.
durations	Vector representing the durations of the stimulus in seconds.
accuracy	Microtime resolution in seconds.

Details

If duration is a single number, it is assumed that all stimulus onsets have the same duration.

Value

A vector in microtime resolution specifying the stimulus boxcar function in 1-0 coding.

Author(s)

M. Welvaert

See Also[specifydesign](#)**Examples**

```
total <- 100
os <- c(1, 21, 41, 61, 81)
d <- 10
out <- stimfunction(total, os, d, 0.1)
```

systemnoise	<i>Generate system noise</i>
-------------	------------------------------

Description

Generates a system noise dataset with specified dimensions and standard deviation. The noise can be either Gaussian or Rician distributed.

Usage

```
systemnoise(dim, nscan, type=c("gaussian","rician"), sigma, vee, template,
verbose = TRUE)
```

Arguments

dim	A vector specifying the dimensions of the image.
nscan	The number of scans in the dataset.
type	Distribution of system noise. Default is gaussian.
sigma	The standard deviation of the noise.
vee	If type=="rician", the non-centrality parameter of the distribution .
template	An array representing the anatomical structure or mask with dimensions equal to dim.
verbose	Logical indicating if warnings should be printed.

Value

An array containing the noise with dimensions specified in dim and nscan.

Author(s)

M. Welvaert

See Also

[temporalnoise](#), [lowfreqdrift](#), [physnoise](#), [tasknoise](#), [spatialnoise](#)

Examples

```
d <- c(10,10,10)
sigma <- 5
nscan <- 100
out <- systemnoise(d, nscan, type="rician", sigma, verbose=FALSE)
```

tasknoise

*Generate task-related noise***Description**

Generates a Gaussian noise dataset with specified dimensions and standard deviation only when a task is performed or activation is present.

Usage

```
tasknoise(act.image, sigma, type=c("gaussian","rician"), vee=1)
```

Arguments

act.image	Array defining where and when activation is present.
sigma	Standard deviation of the noise.
type	Distribution of task-related noise. Default is gaussian.
vee	If type=="rician", the non-centrality parameter of the distribution.

Details

The function generates random Gaussian noise for those voxels in the dataset that show activation. The result is a noise array with specified dimensions and desired standard deviation.

Value

An array containing the noise.

Author(s)

M. Welsaert

See Also

[temporalnoise](#), [lowfreqdrift](#), [physnoise](#), [systemnoise](#), [spatialnoise](#)

Examples

```
d <- c(10,10,10)
sigma <- 5
nscan <- 100
act <- array(rep(0, prod(d)*nscan), dim=c(d,nscan))
act[2:4,2:4,2:4,c(20:30,40:50,60:70)] <- 1
out <- tasknoise(act, sigma)
```

temporalnoise	<i>Generate temporally correlated noise</i>
---------------	---

Description

Generates an autoregressive noise dataset with specified dimensions and standard deviation.

Usage

```
temporalnoise(dim, nscan, sigma, rho = 0.2, template, verbose = TRUE)
```

Arguments

dim	A vector specifying the dimensions of a 2D or 3D array.
nscan	The number of scans in the dataset.
sigma	The standard deviation of the noise.
rho	The autocorrelation coefficients. The length of the vector determines the order of the autoregressive model.
template	An array representing the anatomical structure or mask with dimensions equal to dim.
verbose	Logical indicating if warnings should be printed.

Value

An array containing the noise with dimensions specified in dim.

Author(s)

J. Durnez, B. Moerkerke, M. Welvaert

See Also

[systemnoise](#), [lowfreqdrift](#), [physnoise](#), [tasknoise](#), [spatialnoise](#)

Examples

```
d <- c(10,10,10)
sigma <- 5
nscan <- 100
rho <- c(0.3,-0.7)
out <- temporalnoise(d, nscan, sigma, rho, verbose=FALSE)
```

Index

* **activation**

- balloon, [3](#)
- canonicalHRF, [4](#)
- gammaHRF, [5](#)

* **high-level**

- simprepSpatial, [11](#)
- simprepTemporal, [12](#)
- simTSfmri, [13](#)
- simTSrestingstate, [14](#)
- simVOLfmri, [16](#)

* **low-level**

- balloon, [3](#)
- canonicalHRF, [4](#)
- gammaHRF, [5](#)
- GaussSmoothKernel, [6](#)
- lowfreqdrift, [7](#)
- physnoise, [8](#)
- rrice, [9](#)
- Sim. 3D.GRF, [10](#)
- spatialnoise, [18](#)
- specifydesign, [19](#)
- specifyregion, [20](#)
- stimfunction, [21](#)
- systemnoise, [22](#)
- tasknoise, [23](#)
- temporalnoise, [24](#)

* **noise**

- GaussSmoothKernel, [6](#)
- lowfreqdrift, [7](#)
- physnoise, [8](#)
- rrice, [9](#)
- Sim. 3D.GRF, [10](#)
- spatialnoise, [18](#)
- systemnoise, [22](#)
- tasknoise, [23](#)
- temporalnoise, [24](#)

* **package**

- neuRosim-package, [2](#)

balloon, [3](#), [5](#), [6](#), [20](#), [21](#)

canonicalHRF, [4](#), [4](#), [6](#), [19–21](#)

gammaHRF, [4](#), [5](#), [5](#), [19–21](#)

GaussSmoothKernel, [6](#), [11](#)

lowfreqdrift, [7](#), [9](#), [19](#), [22–24](#)

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

neuRosim-package, [2](#)

physnoise, [8](#), [8](#), [19](#), [22–24](#)

rrice, [9](#)

Sim. 3D.GRF, [7](#), [10](#), [19](#)

simprepSpatial, [11](#), [12](#), [16](#), [17](#)

simprepTemporal, [11](#), [12](#), [13](#), [14](#), [16](#), [17](#)

simTSfmri, [12](#), [13](#), [15](#), [17](#)

simTSrestingstate, [14](#)

simVOLfmri, [11](#), [12](#), [14](#), [16](#)

spatialnoise, [8](#), [9](#), [18](#), [22–24](#)

specifydesign, [19](#), [22](#)

specifyregion, [11](#), [12](#), [20](#), [20](#), [21](#)

stimfunction, [21](#)

systemnoise, [8](#), [9](#), [19](#), [22](#), [23](#), [24](#)

tasknoise, [8](#), [9](#), [19](#), [22](#), [23](#), [24](#)

temporalnoise, [8](#), [9](#), [19](#), [22](#), [23](#), [24](#)