Package 'FIACH'

October 9, 2015

Type Package

Title Retrospective Noise Control for fMRI

Version 0.1.2

Date 2015-09-17

Author Tim Tierney

Maintainer Tim Tierney <fiachmri@gmail.com>

Repository CRAN

Description Useful functions for fMRI preprocessing.

License GPL-2

Depends R (>= 3.2.0)

Imports Rcpp (>= 0.11.4), RNiftyReg (>= 2.0.0),tcltk, tkrplot, utils, stats, graphics, grDevices

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Date/Publication 2015-10-09 10:28:48

R topics documented:

`arrMat` *Array to matrix transformation.*

Description

This function creates a 2D matrix from a 4D/3D image array.

Usage

```
arrMat(input)
```

Arguments

`input`	input must be a 4d or 3d array

Value

returns a matrix(time x No. of voxels)

Examples

```
arr<-array(dim=c(64,64,30,10)) ## 4D array
mat<-arrMat(arr)
dim(mat)
```

---

badData *Identifies spurious observations*

Description

This function identifies spurious observations using a mix of stochastic and deterministic thresholds.

Usage

```
badData(X, meds=NULL, mads=NULL, nMads=1.96, t=0)
```

Arguments

X	X must be a numeric matrix.
meds	Vector containing column center. If NULL will calculate the median.
mads	Vector containing column variability. If NULL will calculate the Median Absolute Deviation.
nMads	The stochastic threshold specified in terms of MADs
t	The deterministic threshold. Specified in percentage distance from median.

Value

Returns a matrix with NaN where data was identified as spurious

Examples

```
x<-rnorm(100)
x[20]<-30
badData(x)
```

basisFunctions *Informed basis set*

Description

This function produces the impulse response for the canonical haemodynamic response function and its derivatives in time and space at a desired sampling rate.

Usage

```
basisFunctions(RT,orth=TRUE)
```

Arguments

RT	RT is the time between samples. It must be specified in seconds (e.g. 1/16).
orth	Boolean describing whether or not the basis should be orthogonalised.

Value

A matrix is returned containing all three impulse responses.

Author(s)

Tim Tierney

Examples

```
RT=1/16
basis<-basisFunctions(RT=RT)
basis<-basis/max(basis)
x<-seq(0,32,RT)
ylim<-c(min(basis)-sd(basis),max(basis)+sd(basis))
plot(x,basis[,1],lwd=7,col="red",type="l",
      main="Informed Basis Set",xlab="Time(seconds)",
      ylab="Intensity(A.U.)",ylim=ylim)

y<-seq(min(ylim),max(ylim),length.out=10)
abline(h=y,col="grey")
legend(x="topright",
       legend=c("Canonical HRF",
                "Temporal Derivative",
                "Dispersion Derivative"),
       col=c("red","blue","green"),lwd=2)
lines(x,basis[,1],lwd=7,col="red")
lines(x,basis[,2],lwd=7,col="blue")
lines(x,basis[,3],lwd=7,col="green")
```

boldContrast *Maximum Theoretical BOLD signal observable*

Description

This function returns a theoretical threshold beyond which BOLD signal changes are likely to be artefactual based on the models presented in Yablonskiy and Haacke(1994).

Usage

```
boldContrast(B0, te, plot = TRUE, random = TRUE, alpha = 0.38, hct = 0.4, cbf.base = 55,
          chi = 1.8e-07, e.base = 0.4, w0 = 267500000,e.act=.1,cbf.act=2*cbf.base)
```

Arguments

B0	Magnetic field strength specified in Tesla.
te	Echo Time specified in ms
plot	Logical indicating whether or not to plot results.
random	Logical indicating if the theoretical model assumes the cylinders embedded in the medium are randomly orientated. If false the cylinders are assumed to be in parallel.
alpha	Power to which cbf.base is raised to estimate Cerebral Blood Volume.
hct	Hematocrit expressed as a real number between 0 and 1.
cbf.base	Cerebral Blood Flow at rest, specified in ml per mg per minute.
chi	Volume susceptibility difference between oxygenated and de-oxygenated blood specified in cgs units.
e.base	1- oxygenation fraction during rest. ranging between 0 and 1
w0	Gyromagnetic ratio of the proton expressed in radians per second per tesla
e.act	1- oxygenation fraction during activation. ranging between 0 and 1. Defaults to .1 to produce a conservative threshold. A value of .2 might produce a more realistic threshold.
cbf.act	Cerebral Blood Flow during activation, specified in ml per mg per minute

Value

A threshold expressed in percent signal change is returned

Author(s)

Tim Tierney

Examples

```
## magnetic field strength of 1.5T and TE of 30ms
## assuming parallel orientation of cylinders.
boldContrast(1.5,30,plot=TRUE,random=FALSE)

## magnetic field strength of 1.5T and TE of
## 30ms assuming random orientation of cylinders
boldContrast(1.5,30,plot=TRUE,random=TRUE)
```

---

colMad *Compute Column Median Absolute Deviations*

Description

This function computes the median absolute deviation of each column in a matrix.

Usage

```
colMad(X)
```

Arguments

X	X must be a numeric matrix.

Value

returns a vector containing the column Median Absolute Deviations.

Examples

```
mat<-matrix(rnorm(100*100),ncol=100)
a<-colMad(mat)
```

---

colMedian *Compute Column Medians*

Description

This function computes the median of each column in a matrix.

Usage

```
colMedian(X)
```

Arguments

X	A must be a numeric matrix.

Value

Returns a vector containing the medians.

Examples

```
mat<-matrix(rnorm(100*100),ncol=100)
a<-colMedian(mat)
```

---

colsd *Compute column standard deviations*

Description

This function computes the standard deviation of each column in a matrix.

Usage

```
colsd(X)
```

Arguments

X	X must be a numeric matrix.

Value

Returns a vector containing the column standard deviations.

Examples

```
mat<-matrix(rnorm(100*100),ncol=100)
a<-colsd(mat)
```

convolve1d *Finite Impulse Response filter*

Description

This function filters data using specified FIR filter coefficients.

Usage

```
convolve1d(x,fir,subtractMed=TRUE)
```

Arguments

x	A numeric matrix where the columns are to be filtered.
fir	The FIR filter coefficients. This should always be a sinc or windowed sinc. Using other fir coefficients will lead to wrong results.
subtractMed	Should the Median be subtracted prior to convolution to prevent Gibb's ringing.

Value

Returns the filtered signal.

Examples

```
## 1.  construct a sinusoid oscillating at .03Hz for 600s with a TR of 2.16s
## 2. add a sinusoid oscillating at .15Hz for 600s with a TR of 2.16s
## 3. make large matrix of signals

test<-sin(2*pi*.03*(seq(0,600,2.16)))
test.noise<-test+sin(2*pi*.15*(seq(0,600,2.16)))
test.mat<-matrix(test.noise,nrow=length(test.noise),ncol=5)

## compute an impulse response. In this case a kaiser window
kais<-kaiserWin(fl=.08,tw=.025,sf=1/2.16,d.sa=70,d.pbr=.1,type="low")
system.time(filt<-convolve1d(test.mat,kais,subtractMed=TRUE))
par(mfrow=c(3,1))
ts.plot(test,ylim=c(-2,2))
ts.plot(test.mat[,1],ylim=c(-2,2))
ts.plot(filt[,1],ylim=c(-2,2))
```

datatypes *Supported Datatypes*

Description

This dataframe contains details of the supported datatypes in FIACH.

Usage

```
data(datatypes)
```

Format

A data frame with 14 observations on the following 3 variables.

`niftyCodes` Strings used to define RNiftyReg datatype output

`niftiCodes` Codes used to define nifti datatypes

`bitpix` number of bits per pixel

Examples

```
data(datatypes)
datatypes
```

dilate *Separable Flat dilation*

Description

This function dilates data with a flat kernel in up to three dimensions.

Usage

```
dilate(input,k)
```

Arguments

`input`	A vector/matrix or 3D array.
`k`	Window width. Note that the window is symmetric around the i^th element with a total of 2*k+1 elements.

Value

returns the dilated array

Examples

```
arr<-array(rnorm(30*30*30), dim=c(30,30,30))
d<-dilate(arr, k=1)
```

---

erode *Separable Flat Erosion*

Description

This function erodes data with a flat kernel in up to three dimensions.

Usage

```
erode(input,k)
```

Arguments

input	A vector/matrix or 3D array.
k	Window width. Note that the window is symmetric around the i^th element with a total of 2*k+1 elements.

Value

returns the eroded array

Examples

```
arr<-array(rnorm(30*30*30), dim=c(30,30,30))
e<-erode(arr, k=1)
```

---

fd *Framewise Displacement*

Description

This function computes the Framewise Displacement of Realignment Parameters. This function assumes that the first 6 columns of the input are the volumetric realignment parameters.

Usage

```
fd(input)
```

Arguments

input	A character vector specifying the path to the rp file or a matrix.

Value

returns a time series containing the fd across time

Examples

```
data(rp)
fd(rp)
```

---

fftN *Zero Padded 1D Fourier transform*

Description

This function is a simple wrapper of Armadillo's fft function. It allows for fast and easy zero padding of a signal.

Usage

```
fftN(X,N=NULL)
```

Arguments

X	X a numeric vector or matrix
N	Length of zero padded signal. If NULL the function will automatically pad sufficiently for a fast transform.

Value

returns the Fourier transform of the signal.

Examples

```
x<-matrix(rnorm(101*1000),nrow = 101,ncol = 1000)
system.time(test1<-fftN(x))
```

`fiach` *FIACH*

Description

This function makes use of The EM algorithm to segment and model noisy areas of the brain. It also utilizes a thresholded spline filter to clean spurious observations in the image.

Usage

```
fiach(input, t, tr, rp=NULL, maxgap = 1, freq = 128,nMads = 1.96)
```

Arguments

`input`	A character vector containing a single string or multiple strings. If multiple strings are provided the data is concatenated and is assumed to be from the same session.The data can be in 4D.nii format or .img and .hdr pairs. If input input is in .img/.hdr pair only the .img file need be specified. These images should be realigned.
`t`	A threshold expressed in percent signal change beyond which the spline filter is used. This threshold is ideally computed using boldContrast.
`tr`	The time between scans. Needed for the high pass filtering.
`rp`	The realignment parameter files that will be appended with additional regressors.
`maxgap`	max number of consecutive time-points you will allow to be filtered before scrubbing (replacement with median) takes place.
`freq`	Desired frequency to high pass filter data at. Specified in seconds. CAUTION!!! THE DEFAULT IS 128 WHICH MAY NOT BE APPROPRIATE FOR ALL DESIGNS.
`nMads`	Number of MADs used for noise calculation.

Value

Nothing is returned to R but the filtered files are written to the directory they came from with the prefix filt_. A directory is also created to store the various diagnostic images and plots produced by this method(median, rTSNR, mask and rTSNR histogram). The regressors to be used in further analysis are in the noise_basis6 file. The global median signal is also outputted in gs.txt file. The framewise displacement is also outputted appended to the noise regressors in fd_noise.txt if movement regressors are supplied.

Author(s)

Tim Tierney

Examples

```
## Not run:
#### create the necessary files ###

dir.create("fiach_example/")
file<-system.file("extdata","motion_ex.nii.gz",package="FIACH")
arr<-readNii(file)
RNiftyReg::writeNifti(arr,"fiach_example/motion_ex.nii.gz",datatype = "short")
data(rp)
write.table(rp[1:13,],"fiach_example/rp.txt",col.names=FALSE,row.names=FALSE)

# running FIACH #
t<-boldContrast(1.5,30)
tr<-2.16
system.time(fiach("fiach_example/motion_ex.nii.gz", t=t,tr=tr, rp="fiach_example/rp.txt"))

## Note that this is a toy example.
## The results are meaningless.
## This is only an example of how
## one would use fiach in a script.

## End(Not run)
```

`gaussKernel` *1 dimensional Gaussian smoothing kernel*

Description

This function produces a Gaussian smoothing kernel that can be used in separable convolution to smooth images.

Usage

```
gaussKernel(fwhm=8,ksize=5,voxsize=1)
```

Arguments

`fwhm`	Full width at half maximum of the kernel.
`ksize`	Size of kernel. Must be Odd
`voxsize`	Size of the voxel in the direction you wish to smooth in. If smoothing a non medical image leave voxsize as 1 unless anisotropic smoothing is required.

Value

Returns a Kernel

Examples

```
ts.plot(gaussKernel(ksize=23,voxsize=1))
```

getDatatype *Determine Datatype*

Description

This function returns the largest datatype from a set of files or images. The return value is a code in RNiftyReg format or the exact nifti code.

Usage

```
getDatatype(input,type="RNiftyReg")
```

Arguments

input	A character vector of file paths, an image or a list of images.
type	A character vector. If equal to "RNiftyReg" the code will be returned in RNiftyReg format. If anything else the nifti code will be returned.

Value

RNiftyReg datatype code or nifti datatype code.

Author(s)

Tim Tierney

Examples

```
file<-system.file("extdata","motion_ex.nii.gz",package="FIACH")
getDatatype(file)
```

gmm *Gaussian Mixture Model*

Description

EM algorithm for mixtures of gaussians

Usage

```
gmm(x, k, imeans = NULL, isd = NULL, ilambda = NULL, print=FALSE, tol=1e-8, maxit = 1000L)
```

Arguments

x	numeric vector containing data
k	integer specifying number of gaussians to be fitted
imeans	optional initial means for mixture model
isd	optional initial standard deviations for mixture model
ilambda	optional initial mixing proportions for mixture model. Must sum to 1.
print	logical indicating whether or not progress is printed to screen.
tol	Convergence criteria for mixture model.
maxit	integer specifying maximum number of iterations allowed

Value

returns a mixture model.

Examples

```
test<-c(rnorm(1000),rnorm(1000,mean = 3,sd = 1))
hist(test)
a<-gmm(test,2)
plot(a)
```

GUI *Graphical User Interface for using the fiach function.*

Description

This function allows for using the fiach function interactively.

Usage

```
GUI()
```

Value

GUI returns nothing.

Examples

```
 ## Not run:
 GUI()

## End(Not run)
```

hampel *Hampel Filter*

Description

This function Hampel Filters each column of a matrix.

Usage

```
hampel(x,k=3,t0=3)
```

Arguments

x	Numeric Matrix where the columns are to be filtered.
k	Window width. Note that the window is symmetric around the i^th element with a total of 2*k+1 elements.
t0	Threshold, expressed in Median Absolute Deviations. Note that setting the threshold to 0 creates a Median filter.

Value

Returns a matrix containing the filtered data

Examples

```
mat<-matrix(rnorm(100*100),ncol=100)
a<-hampel(mat)
```

highBasis *High pass filter basis set*

Description

Generates cosines that can be used to high-pass filter data in a regression framework.

Usage

```
highBasis(N, freq, tr)
```

Arguments

N	Number of time-points to be filtered.
freq	This is the cutoff frequency. It is specified in seconds for comparability with SPM.
tr	This is the time between samples(or the TR for fMRI data.)

Value

A matrix containing the basis set.

Author(s)

Tim Tierney

Examples

```
ts.plot(highBasis(300,128,2.16))
```

---

highPass *SPM high pass filter*

Description

Removes low frequency drifts from data.

Usage

```
highPass(x, freq, tr)
```

Arguments

x	Numeric matrix or vector. If a vector is supplied it will be coerced to a matrix.
freq	This is the cutoff frequency. It is specified in seconds for comparability with SPM.
tr	This is the time between samples(or the TR for fMRI data.)

Value

A high pass filtered matrix will always be returned.

Author(s)

Tim Tierney

Examples

```
t<-seq(1,300)
y<-cos(.01*pi*t)+.1*cos(.2*pi*t)   ## time series with low frequency drift
plot(t,y,type="l",col="red",lwd=5) ## plot it
hp.y<-highPass(y,128,2.16)        ## filter it at 128s (like in SPM) with a tr of 2.16
lines(t,hp.y,col="blue",lwd=5)     ## result has no drift
legend(x="topright",               ## Create the Legend
legend=c("Raw Signal",
         "Highpass Filtered"),
col=c("red","blue"),lwd=2)
```

kaiserWin *Kaiser Window*

Description

Creates a windowed sinc pulse using a kaiser window.

Usage

```
kaiserWin(fh,fl,tw,sf,d.sa,d.pbr,type)
```

Arguments

fh	High pass frequency in Hz.
fl	Low pass frequency in Hz
tw	Transition width in Hz.
sf	Sampling frequency in Hz.
d.sa	Desired Stopband Attenuation specified in dB.
d.pbr	Desired Passband ripple specified in dB.
type	string of either "low","high","band" depending on what you wish to do.

Value

Outputs FIR filter coefficients to desired specifications.

Author(s)

Tim Tierney

Examples

```
kais<-kaiserWin(fl=.08,tw=.025,sf=1/2.16,d.sa=70,d.pbr=.1,type="low")
par(mfrow=c(2,1))
ts.plot(kais,main="Low Pass Kaiser Window")
rawPeriodogram(kais,1/2.16)
```

kmeansMask *Binarize vector, matrix or array*

Description

This function creates a mask for an image using kmeans clustering.

Usage

```
kmeansMask(x)
```

Arguments

x	x can be a vector,matrix or array.

Value

returns a binary vector, matrix or array of zeros and ones.

Examples

```
file<-system.file("extdata","motion_ex.nii.gz",package="FIACH")
arr<-readNii(file)[,,,1]
mask<-kmeansMask(arr)
```

lowBasis *Low pass filter basis set*

Description

Generates cosines that can be used to low-pass filter data in a regression framework.

Usage

```
lowBasis(N, freq, tr)
```

Arguments

N	Number of time-points to be filtered.
freq	This is the cutoff frequency. It is specified in seconds for comparability with SPM.
tr	This is the time between samples(or the TR for fMRI data.)

Value

A matrix containing the basis set.

Author(s)

Tim Tierney

Examples

```
## 300 scans, 10 second cutoff, tr=2.16 seconds
lb<-lowBasis(300,10,2.16)
dim(lb)
plot.ts(lb[,1:10])
```

---

matArr *Matrix to Array Transformation.*

Description

This function creates a 4D/3D image array from a 2D matrix

Usage

```
matArr(mat,dim)
```

Arguments

mat	input must be a 2d matrix
dim	dimensions of array

Value

returns an array of dimension dim

Examples

```
mat<-matrix(nrow=50,ncol=30*30*15)
arr<-matArr(mat,dim=c(30,30,15,50))
dim(arr)
```

naSpline *Interpolate NAs*

Description

This function interpolates NAs in matrices along columns.

Usage

```
naSpline(mat,maxgap=1)
```

Arguments

mat	A must be a numeric matrix.
maxgap	positive integer indicating over what gaps values can be interpolated. Longer gaps will be less accurate.

Value

Returns the interpolated matrix.

Examples

```
x<-rnorm(100)
x[20]<-30
naData<-badData(x)
fixed<-naSpline(naData)
par(mfrow=c(2,1))
ts.plot(x)
ts.plot(fixed)
```

plot.gmm *Plot a Gaussian Mixture Model*

Description

Plots an object of class gmm(Gaussian Mixture Model).

Usage

```
## S3 method for class 'gmm'
plot(x,...)
```

Arguments

x	an object of class gmm
...	Graphical parameters passed to plot command.

Examples

```
test<-c(rnorm(1000),rnorm(1000,mean = 3,sd = 1))
hist(test)
a<-gmm(test,2)
plot(a)
```

---

pseudo *Pseudoinverse.*

Description

This function computes the Pseudoinverse of a design matrix. If a matrix of dependent variables is also supplied the betas will also be returned. Optionally the residuals may be returned as well.

Usage

```
pseudo(x,y=NULL,residuals=FALSE,keepMean=FALSE,includeIntercept=TRUE)
```

Arguments

x	x must be a numeric matrix.
y	y must be a numeric matrix.
residuals	residuals must be either TRUE or FALSE.
keepMean	keepMean must be either TRUE or FALSE. This argument specifies whether residuals should be zero centered or retain their mean.
includeIntercept	includeIntercept must be either TRUE or FALSE. This argument specifies whether a column of ones should be included in x.

Value

Returns a Matrix containing either the pseudoinverse, linear regression coefficients or residuals.

Examples

```
data(trees, package="datasets")

## with intercept
pinv<-pseudo(x=log(trees$Girth),includeIntercept=TRUE)
## without intercept
pinv<-pseudo(x=log(trees$Girth),includeIntercept=FALSE)
##coefficients
coef<-pseudo(x=log(trees$Girth),y=log(trees$Volume))
## residuals
res<-pseudo(x=log(trees$Girth),y=log(trees$Volume),residuals=TRUE)

## standard model
mod<-lm(log(trees$Volume)~log(trees$Girth))
coef2<-coefficients(mod)
res2<-residuals(mod)
## equality

cbind(coef,coef2) ## same but order reversed
cbind(res,res2)   ## same
```

---

quantMask *Compute the Mask Vector*

Description

This function creates a mask vector for an image using a user specified quantile.

Usage

```
quantMask(x,quant=.7)
```

Arguments

x	x can be a 4D/3D array, 2D matrix or 1D vector representing an image.
quant	The quantile level used for masking. Defaults to .7.

Value

returns a binary vector of zeros and ones.

Examples

```
arr<-array(rnorm(10*10*10*10),dim=c(10,10,10,10))
mask<-quantMask(arr,.7)
```

`rawPeriodogram` *Raw Periodogram*

Description

This function computes and plots a raw periodogram. This function may produce inaccurate results on stochastic data.

Usage

```
rawPeriodogram(x,sf=NULL,plot=TRUE,amp=FALSE,phase=FALSE,N=NULL)
```

Arguments

`x`	x a numeric vector
`sf`	Sampling frequency in Hz. If specified the power spectrum will be plotted with Hz instead of normalised frequency on the x-axis.
`plot`	Whether or not to plot the periodogram. If more than one series is used only the first will be plotted.
`amp`	Whether or not to return the amplitude spectrum.
`phase`	Whether or not to return the phase spectrum.
`N`	Length of Fourier transform used to compute spectrum. If NULL N will be chosen for speed and not to prevent scalloping loss.

Value

Returns the periodogram and optionally plots it.

Examples

```
hz.slow<-5
hz.fast<-50
t<-seq(0,1,length.out=300)

sin.sig.slow<-sin(2*pi*t*hz.slow)
sin.sig.fast<-sin(2*pi*t*hz.fast)
sin.sig.combo<-sin.sig.slow+sin.sig.fast

ts.plot(sin.sig.combo)
rawPeriodogram(sin.sig.combo)
rawPeriodogram(sin.sig.combo,300)
```

readNii *Input function for fmri data*

Description

This function reads 4D and 3d nifti and nifti files

Usage

```
readNii(input,fourD=TRUE)
```

Arguments

input	A character vector containing a single string or multiple strings. If multiple strings are provided the data is concatenated and is assumed to be from the same session. The data can be in 4d.nii format or .img and .hdr pairs. If input input is in .img/.hdr pair only the .img file need be specified
fourD	If set to false nii objects are returned for each image.

Value

A 4D array is returned with time being the fourth dimension

Author(s)

Tim Tierney

Examples

```
file<-system.file("extdata","motion_ex.nii.gz",package="FIACH")
func<-readNii(file)
```

rowMad *Compute row Median Absolute Deviations*

Description

This function computes the Median Absolute Deviation of each row in a matrix.

Usage

```
rowMad(X)
```

Arguments

X	X must be a numeric matrix.

Value

Returns a vector containing the row Median Absolute Deviations.

Examples

```
mat<-matrix(rnorm(100*100),ncol=100)
a<-rowMad(mat)
```

---

rowMedian *Compute Row Medians*

Description

This function computes the median of each row in a matrix.

Usage

```
rowMedian(X)
```

Arguments

X	A must be a numeric matrix.

Value

Returns a vector containing the medians.

Examples

```
mat<-matrix(rnorm(100*100),ncol=100)
a<-rowMedian(mat)
```

---

rowsd *Compute row standard deviations*

Description

This function computes the standard deviation of each row in a matrix.

Usage

```
rowsd(X)
```

Arguments

X	X must be a numeric matrix.

Value

Returns a vector containing the row standard deviations.

Examples

```
## Not run:
mat<-matrix(rnorm(100*100),ncol=100)
a<-rowsd(mat)



## End(Not run)
```

---

rp *Realignment parameters*

Description

Realignment Parameters which do no have any obvious extreme movement despite severe image artefact.

Usage

```
data(rp)
```

Format

A data frame with 104 observations on the following 6 variables.

- V1 X Translation parameter
- V2 Y Translation parameter
- V3 Z Translation parameter
- V4 Pitch
- V5 Roll
- V6 Yaw

Examples

```
data(rp)
z<-rp[,3]
dz<- c(0,diff(z))
plot(1:104,dz,ylim=c(-1,1),main="Volume-Volume Displacement",
xlab="Time(scans)",ylab="Displacement(mm)")
lines(1:104,dz,col="blue",lwd=3)
legend("topright",legend="dZ/dt Translation",lwd=3,col="blue")
arrows(54,.5,91,-.184)
text(48,.5,"-.184mm")
```

---

selectR *Interactive File Selector*

Description

This function allows for interactive file selection using regular expressions.

Usage

```
selectR(update=TRUE)
```

Arguments

update	logical indicating whether or not the working directory should be changed to the current selected directory. This means that if this function is called a second time it will initiate in the directory it was in when previously called.

Value

Returns a character vector of file paths. If no files are selected this will still be a character vector but will have length = 0. This function does not work well with network drives. If network drives are found this function initiates at the drive select section.

Examples

```
 ## Not run:
selectR()

## End(Not run)
```

sepConvolve3d *Fast 3d Convolution*

Description

This function convolves data in three different directions with potentially different impulses.

Usage

```
sepConvolve3d(x,kernX, kernY, kernZ)
```

Arguments

x	A 3D/4D array. If a 4D array then each 3d array along the fourth dimension is convolved.
kernX	filter kernel to be used in the x direction
kernY	filter kernel to be used in the y direction
kernZ	filter kernel to be used in the z direction

Value

returns the convolved array

Examples

```
func<-readNii(system.file("extdata","motion_ex.nii.gz",package="FIACH"))

kernx<-gaussKernel(8,21,3.33)
kerny<-gaussKernel(8,21,3.33)
kernz<-gaussKernel(8,21,4)

smooth<-sepConvolve3d(func,kernx,kerny,kernz)
```

sinc *Sinc Pulse*

Description

Creates a sinc with desired properties.

Usage

```
sinc(fh,fl,tw,sf,type,n)
```

Arguments

fh	High pass frequency in Hz.
fl	Low pass frequency in Hz.
tw	Transition width in Hz.
sf	sampling frequency in Hz.
type	string of either "low","high","band" depending on what you wish to do.
n	range over which filter is to be evaluated.

Value

outputs a sinc pulse evaluated over n.

Author(s)

Tim Tierney

Examples

```
par(mfrow=c(2,1))
si<-sinc(fl=.08,tw=0,type="low",n=-40:40,sf=1/2.16)
ts.plot(si,ylab="",main="Sinc Pulse")
s<-rawPeriodogram(si,1/2.16)
```

---

spmHrf *Canonical Haemodynamic Response Function.*

Description

This function produces the impulse response for the canonical haemodynamic response function at a desired sampling rate.

Usage

```
spmHrf(RT, p = c(6, 16, 1, 1, 6, 0, 32))
```

Arguments

RT	RT is the desired sampling frequency. It must be specified in Hz(e.g. 1/16).
p	Default parameters for impulse response estimation identical to SPM's implementation.

Value

comp1	The first component of the list is the impulse response.
comp2	The second component are the parameters used to estimate it.

Author(s)

Tim Tierney

Examples

```
RT<-1/16
can.hrf<-spmHrf(RT)[[1]]
x<-seq(0,32,RT)
plot(x,can.hrf,lwd=7,col="red",type="l",
main="Canonical HRF",xlab="Time(seconds)",
ylab="Intensity(A.U.)")
```

---

spmOrth *Serial Orthogonalisation*

Description

This function recursively orthogonalises variables in a matrix to the first column. It is designed to emulate the SPM approach to orthogonalisation. However instead of computing the results using a pseudo-inverse R's qr functions are used.

Usage

```
spmOrth(a)
```

Arguments

a	matrix to be orthogonalised

Value

The recursively orthogonalised matrix is returned

Author(s)

Tim Tierney

Examples

```
a<-basisFunctions(1/16,orth=FALSE)
ts.plot(a,ylim=c(-.007,.015))
ts.plot(spmOrth(a),ylim=c(-.007,.015))
```

t2Grey *Transverse Relaxation*

Description

This function calculates the T2 or R2 of grey matter at varying field strength

Usage

```
t2Grey(B0,relax)
```

Arguments

B0	Magnetic field strength specified in Tesla.
relax	Logical indicating if R2 or T2 should be returned

Value

Either the T2 in ms or the R2 in Hz is returned

Author(s)

Tim Tierney

Examples

```
t2Grey(1.5,relax=TRUE)
t2Grey(1.5,relax=FALSE)
```

t2sGrey *Effective Transverse Relaxation*

Description

This function returns a theoretical value for T2* based on the models presented in Yablonskiy and Haacke (1994).

Usage

```
t2sGrey(B0, alpha = 0.38, hct = 0.4, cbf.base = 55, chi = 1.8e-07,
        e.base = 0.4, w0 = 267500000,r2=t2Grey(B0,TRUE),relax=TRUE)
```

Arguments

B0	Magnetic field strength specified in Tesla.
alpha	Power to which cbf.base is raised to estimate Cerebral Blood Volume.
hct	Hematocrit expressed as a real number between 0 and 1.
cbf.base	Cerebral Blood Flow at rest, specified in ml per mm per minute.
chi	Volume susceptibility difference between oxygenated and deoxygenated blood specified in cgs units.
e.base	1- oxygenation fraction during rest. ranging between 0 and 1
w0	Gyromagnetic ratio of the proton expressed in radians per second per tesla
r2	Transverse relaxation rate. Specified in Hz
relax	Boolean indicating whether R2* or T2* should be returned

Value

Either R2* in Hz or T2* in ms is returned

Author(s)

Tim Tierney

Examples

```
t2sGrey(3,relax=TRUE)
t2sGrey(3,relax=FALSE)
```

---

viewR *Interactive Medical Image Viewer*

Description

This function allows for interactive viewing of 3D and 4D medical Images.

Usage

```
viewR(data=NULL,xyz=NULL,ret=FALSE)
```

Arguments

data	A 3D/4D array. If the array has an attribute named pixdim viewR will try and adjust the aspect appropriately. If NULL an interactive file selector will open to select the images one would like to view. If file paths to 3D/4D images are supplied they will be read and displayed. Multiple 3D files will be concatenated if the dimensions match. viewR can then be ran in movie mode. Currently this viewR does not display world coordinates. Instead voxel coordinates are used. As such it does not enforce information obtained from nifti headers.

xyz	numeric vector indicating initial voxel coordinates.
ret	Logical indicating if read data should be returned.

Value

If ret is set to TRUE the data will be returned.

Examples

```
 ## Not run:
url <- "http://nifti.nimh.nih.gov/nifti-1/data/filtered_func_data.nii.gz"
urlfile <- tempfile(pattern="filtered_func_data", fileext=".nii.gz")
download.file(url, urlfile, quiet=TRUE)
data<-readNii(urlfile)
viewR(data)
## Try coordinate  47,25,10

## End(Not run)
```

---

zeroNa *Zero non finite values*

Description

This function replaces NA, Inf, -Inf, and NaN values of a matrix, array or vector with zero.

Usage

```
zeroNa(input)
```

Arguments

input	A numeric vector, matrix, or array. If an array it must have less than 5 dimensions.

Value

Returns the same input but with zeros in place of the non-finite values.

Examples

```
 test<-matrix(rnorm(100),10,10)

test[5,5]<-NA
test
zeroNa(test)

test[5,5]<-NaN
test
```

```
zeroNa(test)

test[5,5]<- -Inf
test
zeroNa(test)

test[5,5]<- +Inf
test
zeroNa(test)
```

Index