

# Package ‘fslr’

July 31, 2014

**Type** Package

**Title** Wrapper functions for FSL (FMRIB Software Library) from  
Functional MRI. of the Brain (FMRIB)

**Version** 1.1

**Date** 2014-06-17

**Author** John Muschelli <muschellij2@gmail.com>

**Maintainer** John Muschelli <muschellij2@gmail.com>

**Description** Wrapper functions that interface with FSL  
(<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/>), a powerful and commonly-used  
neuroimaging software, using system commands. The goal is to be able to  
interface with FSL completely in R, where you pass R nifti objects and the  
function executes an FSL command and returns an R nifti object if desired.

**Depends** stringr, oro.nifti, matrixStats

**License** GPL-2

**VignetteBuilder** knitr

**Suggests** knitr

**BugReports** <https://github.com/muschellij2/fslr/issues>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-07-31 07:17:59

**R topics documented:**

cal_img	3
checking	3
checkout	4
check_file	4
check_nifti	5
datatype	5
flirt	6
flirt.help	6
fslbet	7
fslbet.help	7
fslbin	8
fslerode	8
fslfill	9
fslhd	10
fslhd.help	11
fslhd.parse	11
fslhelp	12
fslmask	12
fslmaths	13
fslmaths.help	14
fslmerge	14
fslmerge.help	15
fslrange	15
fslsmooth	16
fslstats	17
fslstats.help	17
fslsub2	18
fslthresh	19
fslval	20
fslval.help	20
fslview	21
fslview.help	21
Get FSLDir	22
get.fsl	22
get.fsloutput	22
get.imgext	23
getForms	23
have.fsl	24
melodic	24
melodic.help	25
newnii	25
nii.stub	26
rescale_img	26
tempimg	27
voxdim	27
zero_trans	28

<code>cal_img</code>	3
<code>zscore_img</code> . . . . .	28

<b>Index</b>	<b>30</b>
--------------	-----------

---

<code>cal_img</code>	<i>Set Max/Min for nifti object by range of data</i>
----------------------	--

---

**Description**

Rescales image `cal_max` and `cal_min` slots to be the max and min, respectively, of an object of class `nifti`, with `na.rm=TRUE`. This is so that when images are rendered/written, the values correspond to those in the array (stored in `.Data` slot) are plotted on correct grayscale and no error is given by `writeNIFTI`.

**Usage**

`cal_img(img)`

**Arguments**

`img`                    `nifti` object

**Value**

object of type `nifti`

---

<code>checking</code>	<i>Check if filename is character or nifti object</i>
-----------------------	---

---

**Description**

Check if filename is character or `nifti` object

**Usage**

`checking(file)`

**Arguments**

`file`                    character or `nifti` object

**Value**

character filename or temporary `nii`

---

checkout	<i>Determine if <math>Q</math> and <math>S</math> forms are consistent</i>
----------	--

---

**Description**

This function determines if the determinants of the sform and qform have the same sign

**Usage**

```
checkout(hd)
```

**Arguments**

hd (list) sforms from [getForms](#)

**Value**

logical indicating if sform and qform consistent

---

check_file	<i>Wrapper for getForms with filename</i>
------------	---

---

**Description**

Wrapper for getForms with filename

**Usage**

```
check_file(file)
```

**Arguments**

file (character) filename of image to be checked

**Value**

result of [checkout](#)

---

check_nifti	<i>Check if nifti image or read in</i>
-------------	--

---

**Description**

Simple check to see if input is character or of class nifti

**Usage**

```
check_nifti(x, reorient = FALSE, allow.array = FALSE)
```

**Arguments**

x	character path of image or an object of class nifti, or array
reorient	(logical) passed to <a href="#">readNIFTI</a> if the image is to be re-oriented
allow.array	(logical) Are array types allowed (TRUE) or should there be an error if the object is not character or class nifti.

**Value**

nifti object or array if allow.array=TRUE and x is an array

**See Also**

[readNIFTI](#)

---

datatype	<i>Change Data type for img</i>
----------	---------------------------------

---

**Description**

Tries to figure out the correct datatype for image. Useful for image masks - makes them binary if

**Usage**

```
datatype(img, datatype = NULL, bitpix = NULL, trybyte = TRUE)
```

**Arguments**

img	nifti object (or character of filename)
datatype	(NULL) character of datatype see <a href="#">convert.datatype</a>
bitpix	(NULL) character of bitpix see <a href="#">convert.bitpix</a>
trybyte	(logical) Should you try to make a byte (UINT8) if image in c(0, 1)?

**Value**

object of type nifti

---

flirt *Register using FLIRT*

---

### Description

This function calls `fslirt` to register `infile` to `reffield` and either saves the image or returns an object of class `nifti`, along with the transformation matrix `omat`

### Usage

```
flirt(infile, reffield, omat, dof, outfile = NULL, retimg = FALSE,
      reorient = FALSE, intern = TRUE, opts = "", ...)
```

### Arguments

<code>infile</code>	(character) input filename
<code>reffield</code>	(character) reference image to be registered to
<code>omat</code>	(character) Output matrix name
<code>dof</code>	(numeric) degrees of freedom
<code>outfile</code>	(character) output filename
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to <code>readNIFTI</code> .
<code>intern</code>	(logical) pass to <code>system</code>
<code>opts</code>	(character) additional options to FLIRT
<code>...</code>	additional arguments passed to <code>readNIFTI</code> .

### Value

character or logical depending on `intern`

---

flirt.help *FLIRT help*

---

### Description

This function calls `flirt`'s help

### Usage

```
flirt.help()
```

### Value

Prints help output and returns output as character vector

---

fslbet	<i>Use FSL's Brain Extraction Tool (BET)</i>
--------	--

---

**Description**

This function calls bet to extract a brain from an image, usually for skull stripping.

**Usage**

```
fslbet(infile, outfile = NULL, retimg = FALSE, reorient = FALSE,  
intern = TRUE, opts = "", betcmd = c("bet2", "bet"), ...)
```

**Arguments**

infile	(character) input filename
outfile	(character) output filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
intern	(logical) pass to <a href="#">system</a>
opts	(character) additional options to FLIRT
betcmd	(character) Use bet or bet2 function
...	additional arguments passed to <a href="#">readNIFTI</a> .

**Value**

character or logical depending on intern

---

fslbet.help	<i>Help for FSL BET</i>
-------------	-------------------------

---

**Description**

This function calls bet's help

**Usage**

```
fslbet.help()
```

**Value**

Prints help output and returns output as character vector

---

fslbin *Binarize Image using FSL*

---

### Description

This function calls `fslmaths -bin`. The R functions wraps `fslmaths`

### Usage

```
fslbin(file, outfile = NULL, retimg = FALSE, reorient = FALSE,
       intern = TRUE, opts = "", ...)
```

### Arguments

<code>file</code>	(character) image to be binarized
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
<code>intern</code>	(logical) to be passed to <a href="#">system</a>
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>
<code>...</code>	additional arguments passed to <a href="#">readNIFTI</a> .

### Value

If `retimg` then object of class `nifti`. Otherwise, Result from system command, depends if `intern` is `TRUE` or `FALSE`.

---

fslero *Erode image using FSL*

---

### Description

This function calls `fslmaths -ero` to erode an image with either the default FSL kernel or the kernel specified in `kopts`. The function either saves the image or returns an object of class `nifti`.

### Usage

```
fslero(file, outfile = NULL, retimg = FALSE, reorient = FALSE,
       intern = TRUE, kopts = "", opts = "", ...)
```



**Arguments**

file	(character) image to be eroded
outfile	(character) resultant eroded image name
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
intern	(logical) to be passed to <a href="#">system</a>
kopts	(character) options for kernel
opts	(character) additional options to be passed to <a href="#">fslmaths</a>
...	additional arguments passed to <a href="#">readNIFTI</a> .

**Value**

Result from system command, depends if intern is TRUE or FALSE. If retimg is TRUE, then the image will be returned.

**Examples**

```
if (have.fsl()){
  system.time({
    x = array(rnorm(1e6), dim = c(100, 100, 100))
    img = nifti(x, dim= c(100, 100, 100),
               datatype = convert.datatype()$FLOAT32, cal.min = min(x),
               cal.max = max(x), pixdim = rep(1, 4))
    mask = img > .5
    eroded = fslerode(mask, kopts = "-kernel boxv 5", retimg=TRUE)
  })
}
```

---

fslfill

*Fill image holes*


---

**Description**

This function calls `fslmaths -fillh` to fill in image holes and either saves the image or returns an object of class nifti

**Usage**

```
fslfill(file, outfile = NULL, bin = TRUE, retimg = FALSE,
        reorient = FALSE, intern = TRUE, ...)
```

**Arguments**

file	(character) filename of image to be filled
outfile	(character) name of resultant filled file
bin	(logical) binarize the image before filling
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
intern	(logical) pass to <a href="#">system</a>
...	additional arguments passed to <a href="#">readNIFTI</a> .

**Value**

character or logical depending on intern

**Examples**

```

if (have.fsl()){
system.time({
x = array(rnorm(1e6), dim = c(100, 100, 100))
img = nifti(x, dim= c(100, 100, 100),
datatype = convert.datatype())$FLOAT32, cal.min = min(x),
cal.max = max(x), pixdim = rep(1, 4))
mask = img > .5
eroded = fslerode(mask, kopts = "-kernel boxv 5", retimg=TRUE)
filled = fslfill(eroded, retimg= TRUE)
})
}

```

---

fslhd

*Get Nifti header using FSL*


---

**Description**

This function calls fslhd to obtain a nifti header

**Usage**

```
fslhd(file, opts = "")
```

**Arguments**

file	(character) image filename or character of class nifti
opts	(character) additional options to be passed to fslhd

**Value**

Character of information from fslhd

---

fslhd.help	<i>FSLhd help</i>
------------	-------------------

---

**Description**

This function calls fslhd's help

**Usage**

```
fslhd.help()
```

**Value**

Prints help output and returns output as character vector

---

fslhd.parse	<i>Parse FSL Header</i>
-------------	-------------------------

---

**Description**

This function takes in a FSL header and parses the components

**Usage**

```
fslhd.parse(hd)
```

**Arguments**

hd (character) header from [fslhd](#)

**Value**

data.frame of information from FSL header

---

fslhelp *Wrapper for getting fsl help*

---

### Description

This function takes in the function and returns the help from FSL for that function

### Usage

```
fslhelp(func_name, help.arg = "--help", extra.args = "")
```

### Arguments

func_name	FSL function name
help.arg	Argument to print help, usually "--help"
extra.args	Extra arguments to be passed other than --help

### Value

Prints help output and returns output as character vector

---

fslmask *Mask image using FSL*

---

### Description

This function calls `fslmaths -mas` to mask an image from an image mask and either saves the image or returns an object of class `nifti`

### Usage

```
fslmask(file, mask, outfile = NULL, retimg = FALSE, reorient = FALSE,
intern = TRUE, opts = "", ...)
```

### Arguments

file	(character) image to be masked
mask	(character) mask given for image
outfile	(character) resultant masked image name
retimg	(logical) return image of class <code>nifti</code>
reorient	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to <code>readNIFTI</code> .
intern	(logical) to be passed to <code>system</code>
opts	(character) additional options to be passed to <code>fslmask</code>
...	additional arguments passed to <code>readNIFTI</code> .

**Value**

Result from system command, depends if intern is TRUE or FALSE. if (have.fsl()) system.time(x = array(rnorm(1e6), dim = c(100, 100, 100)) img = nifti(x, dim= c(100, 100, 100), datatype = convert.datatype())\$FLOAT32, cal.min = min(x), cal.max = max(x), pixdim = rep(1, 4)) mask = img > .5 masked = fslmask(img, mask = mask, retimg=TRUE) )

---

fslmaths

*FSL Maths*


---

**Description**

This function calls fslmaths

**Usage**

```
fslmaths(file, outfile = NULL, retimg = FALSE, reorient = FALSE,
intern = TRUE, opts = "", ...)
```

**Arguments**

file	(character) image to be manipulated
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
intern	(logical) to be passed to <a href="#">system</a>
opts	(character) operations to be passed to fslmaths
...	additional arguments passed to <a href="#">readNIFTI</a> .

**Value**

If retimg then object of class nifti. Otherwise, Result from system command, depends if intern is TRUE or FALSE.

---

 fslmaths.help

*FSL Maths Help*


---

**Description**

This function calls fslmaths's help

**Usage**

```
fslmaths.help()
```

**Value**

Prints help output and returns output as character vector

**Examples**

```
if (have.fsl()){
  fslmaths.help()
}
```

---

fslmerge

*Merge images using FSL*


---

**Description**

This function calls fslmerge to merge files on some dimension and either saves the image or returns an object of class nifti

**Usage**

```
fslmerge(infiles, direction = c("x", "y", "z", "t", "a"), outfile = NULL,
         retimg = FALSE, reorient = FALSE, intern = TRUE, ...)
```

**Arguments**

infiles	(character) input filenames
direction	(character) direction to merge over, x, y, z, t (time), a (auto)
outfile	(character) output filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
intern	(logical) pass to <a href="#">system</a>
...	additional arguments passed to <a href="#">readNIFTI</a> .

**Value**

character or logical depending on intern

---

fslmerge.help	<i>FSLMerge help</i>
---------------	----------------------

---

**Description**

This function calls fslmerge's help

**Usage**

```
fslmerge.help()
```

**Value**

Prints help output and returns output as character vector

---

fslrange	<i>Get range of an image</i>
----------	------------------------------

---

**Description**

This function calls fslstats -R to get the range of an image

**Usage**

```
fslrange(file)
```

**Arguments**

file (character) filename of image to be checked

**Value**

numeric vector of length 2

fslsmooth

*Gaussian smooth image using FSL***Description**

This function calls `fslmaths -s` to smooth an image and either saves the image or returns an object of class `nifti`

**Usage**

```
fslsmooth(file, sigma = 10, mask = NULL, outfile = NULL, reting = FALSE,
          reorient = FALSE, intern = TRUE, ...)
```

**Arguments**

<code>file</code>	(character) image to be smoothed
<code>sigma</code>	(numeric) sigma (in mm) of Gaussian kernel for smoothing
<code>mask</code>	(character) optional mask given for image
<code>outfile</code>	(character) resultant smoothed image name (optional) if not give, will be the stub of the filename then <code>_sigma</code>
<code>reting</code>	(logical) return image of class <code>nifti</code>
<code>reorient</code>	(logical) If <code>reting</code> , should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
<code>intern</code>	(logical) to be passed to <a href="#">system</a>
<code>...</code>	additional arguments passed to <a href="#">readNIFTI</a> .

**Value**

Result from system command, depends if `intern` is `TRUE` or `FALSE`.

**Examples**

```
if (have.fsl()){
  system.time({
    x = array(rnorm(1e6), dim = c(100, 100, 100))
    img = nifti(x, dim= c(100, 100, 100),
              datatype = convert.datatype()$FLOAT32, cal.min = min(x),
              cal.max = max(x), pixdim = rep(1, 4))
    s.img = fslsmooth(img, reting=TRUE)
  })
}
```



---

fslstats

*FSL Stats*

---

### Description

This function calls `fslstats`

### Usage

```
fslstats(file, opts = "")
```

### Arguments

`file` (character) filename of image to be checked  
`opts` (character) operation passed to `fslstats`

### Value

Result of `fslstats` command

### Examples

```
if (have.fsl()){  
  system.time({  
    x = array(rnorm(1e6), dim = c(100, 100, 100))  
    img = nifti(x, dim= c(100, 100, 100),  
    datatype = convert.datatype()$FLOAT32, cal.min = min(x),  
    cal.max = max(x), pixdim = rep(1, 4))  
    entropy = fslstats(img, opts='-E')  
  })  
}
```

---

fslstats.help

*FSL Stats Help*

---

### Description

This function calls `fslstats`'s help

### Usage

```
fslstats.help()
```

### Value

Prints help output and returns output as character vector

**Examples**

```
if (have.fsl()){
  fslstats.help()
}
```

fslsub2

*Subsample image by factor of 2***Description**

This function calls `fslmaths -subsamp2` to subsample an image and either saves the image or returns an object of class `nifti`

**Usage**

```
fslsub2(file, outfile = NULL, retimg = FALSE, reorient = FALSE,
        intern = TRUE, ...)
```

**Arguments**

<code>file</code>	(character) filename of image to be subsampled
<code>outfile</code>	(character) name of resultant subsampled file
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to <a href="#">readNIFTI</a> .
<code>intern</code>	(logical) pass to <a href="#">system</a>
<code>...</code>	additional arguments passed to <a href="#">readNIFTI</a> .

**Value**

character or logical depending on `intern`

**Examples**

```
if (have.fsl()){
  system.time({
    x = array(rnorm(1e6), dim = c(100, 100, 100))
    img = nifti(x, dim= c(100, 100, 100),
               datatype = convert.datatype()$FLOAT32, cal.min = min(x),
               cal.max = max(x), pixdim = rep(1, 4))
    subsamp = fslsub2(img, retimg=TRUE)
    print(voxdim(subsamp))
  })
}
```

---

fslthresh	<i>Threshold an image</i>
-----------	---------------------------

---

### Description

This function calls `fslmaths -thr -uthr` to threshold an image and either saves the image or returns an object of class `nifti`

### Usage

```
fslthresh(file, outfile = NULL, thresh = 0, uthresh = NULL,
          retimg = FALSE, reorient = FALSE, intern = TRUE, opts = "", ...)
```

### Arguments

<code>file</code>	(character) filename of image to be thresholded
<code>outfile</code>	(character) name of resultant thresholded file
<code>thresh</code>	(numeric) threshold (anything below set to 0)
<code>uthresh</code>	(numeric) upper threshold (anything above set to 0)
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to <code>readNIFTI</code> .
<code>intern</code>	(logical) pass to <code>system</code>
<code>opts</code>	(character) additional options to be passed to <code>fslmaths</code>
<code>...</code>	additional arguments passed to <code>readNIFTI</code> .

### Value

character or logical depending on `intern`

### Examples

```
if (have.fsl()){
  system.time({
    x = array(rnorm(1e6), dim = c(100, 100, 100))
    img = nifti(x, dim= c(100, 100, 100),
              datatype = convert.datatype()$FLOAT32, cal.min = min(x),
              cal.max = max(x), pixdim = rep(1, 4))
    thresh = fslthresh(img, thresh=0, uthresh = 2, retimg=TRUE)
  })
}
```

---

fslval	<i>Get value from FSL header</i>
--------	----------------------------------

---

**Description**

This function calls `fslval` to obtain a nifti header

**Usage**

```
fslval(file, keyword = "")
```

**Arguments**

file	(character) image filename or character of class nifti
keyword	(character) keyword to be taken from fslhd

**Value**

Character of information from fslhd field specified in keyword

---

fslval.help	<i>fslval help</i>
-------------	--------------------

---

**Description**

This function calls `fslval`'s help

**Usage**

```
fslval.help()
```

**Value**

Prints help output and returns output as character vector

---

fslview	<i>Open image in FSLView</i>
---------	------------------------------

---

**Description**

This function calls `fslview` to view an image in the FSL viewer

**Usage**

```
fslview(file, intern = TRUE, opts = "")
```

**Arguments**

file	(character) filename of image to be thresholded
intern	(logical) pass to <a href="#">system</a>
opts	(character) options for FSLView

**Value**

character or logical depending on intern

---

fslview.help	<i>FSLView help</i>
--------------	---------------------

---

**Description**

This function calls `fslview`'s help

**Usage**

```
fslview.help()
```

**Value**

Prints help output and returns output as character vector

---

Get FSLDir	<i>Get FSL's Directory</i>
------------	----------------------------

---

**Description**

Finds the FSLDIR from system environment or `getOption("fsl.path")` for location of FSL functions and returns it

**Usage**

`fsldir()`

**Value**

Character path

---

get.fsl	<i>Get FSL's Location</i>
---------	---------------------------

---

**Description**

Finds the FSLDIR from system environment or `getOption("fsl.path")` for location of FSL functions

**Usage**

`get.fsl()`

**Value**

NULL if FSL in path, or bash code for setting up FSL DIR

---

get.fsloutput	<i>Determine FSL output type</i>
---------------	----------------------------------

---

**Description**

Finds the FSLOUTPUTTYPE from system environment or `getOption("fsl.outputtype")` for output type (nii.gz, nii, ANALYZE,etc)

**Usage**

`get.fsloutput()`

**Value**

FSLOUTPUTTYPE, such as NIFTI\_GZ. If none found, uses NIFTI\_GZ as default

---

get.imgext	<i>Determine extension of image based on FSLOUTPUTTYPE</i>
------------	--

---

**Description**

Runs `get.fsoutput()` to extract FSLOUTPUTTYPE and then gets corresponding extension (such as .nii.gz)

**Usage**

```
get.imgext()
```

**Value**

Extension for output type

---

getForms	<i>Get Q and S Forms of orientation matrix</i>
----------	--

---

**Description**

This function obtains the s and q forms of an image transformation matrix

**Usage**

```
getForms(file)
```

**Arguments**

file            (character) filename of image to pass to header

**Value**

list with elements of sform and qform and their respective codes

---

have.fsl	<i>Logical check if FSL is accessible</i>
----------	---

---

**Description**

Uses `get.fsl` to check if FSLDIR is accessible or the option `fsl.path` is set and returns logical

**Usage**

```
have.fsl()
```

**Value**

Logical TRUE is FSL is accessible, FALSE if not

**Examples**

```
have.fsl()
```

---

melodic	<i>Run MELODIC ICA</i>
---------	------------------------

---

**Description**

This function calls `melodic`

**Usage**

```
melodic(file, outdir = dirname(file), intern = TRUE, opts = "")
```

**Arguments**

file	(character) image to be run
outdir	(character) output directory. (Default <code>dirname(file)</code> )
intern	(logical) pass to <a href="#">system</a>
opts	(character) options for <code>melodic</code>

**Value**

character or logical depending on `intern`



---

melodic.help	<i>MELODIC help</i>
--------------	---------------------

---

**Description**

This function calls melodic's help

**Usage**

```
melodic.help()
```

**Value**

Prints help output and returns output as character vector

---

newnii	<i>Resets image parameters for a copied nifti object</i>
--------	--

---

**Description**

Resets the slots of a nifti object, usually because an image was loaded, then copied and filled in with new data instead of making a nifti object from scratch. Just a wrapper for smaller functions

**Usage**

```
newnii(img, ...)
```

**Arguments**

img	nifti object (or character of filename)
...	arguments to be passed to <a href="#">datatype</a>

**Value**

object of type nifti

---

nii.stub	<i>Grab nii file stubname</i>
----------	-------------------------------

---

**Description**

Quick helper function to strip of .nii or .nii.gz from filename

**Usage**

```
nii.stub(x)
```

**Arguments**

x	character vector of filenames ending in .nii or .nii.gz
---	---

---

rescale_img	<i>Image Rescaler</i>
-------------	-----------------------

---

**Description**

Rescales an image to be in certain value range. This was created as sometimes DICOM scale and slope parameters may be inconsistent across sites and the data need to be value restricted

**Usage**

```
rescale_img(filename, pngname = NULL, write.nii = TRUE, min.val = -1024,
  max.val = 3071, ROIformat = FALSE, writer = "dcm2nii", ...)
```

**Arguments**

filename	filename of image to be read into R or nifti object
pngname	filename of png of histogram of values of image to be made. For no png - set to NULL (default)
write.nii	logical - should the image be written. filename must be character if this is TRUE (default)
min.val	minimum value of image (default -1024 (for CT)). If no thresholding set to -Inf
max.val	maximum value of image (default 3071 (for CT)). If no thresholding set to Inf
ROIformat	if TRUE, any values $< 0$ will be set to 0
writer	character value to add to description slot of NIFTI header
...	extra methods to be passed to <a href="#">writeNIFTI</a>

**Value**

Object of class nifti

---

tempimg	<i>Create temporary nii.gz file for FSL</i>
---------	---

---

**Description**

Takes in a object of class nifti, writes it to a temp file, appends .nii.gz as [writeNIFTI](#) adds it.

**Usage**

```
tempimg(nim)
```

**Arguments**

nim	object of class nifti
-----	-----------------------

**Value**

filename of output nii.gz

---

voxdim	<i>Gets Voxel Dimensions</i>
--------	------------------------------

---

**Description**

Grabs the pixdim and takes the correct elements

**Usage**

```
voxdim(img)
```

**Arguments**

img	nifti object
-----	--------------

**Value**

Vector of length 3

---

zero_trans	<i>Change intercept to 0 and slope to 1</i>
------------	---

---

**Description**

Forces image scl\_slope to 1 and scl\_inter to be 0 of slots of class nifti. This is so that when images are rendered/written, the values correspond to those in the array (stored in Data slot) and are not scaled.

**Usage**

```
zero_trans(img)
```

**Arguments**

img	nifti object (or character of filename)
-----	---

**Value**

object of type nifti

---

zscore_img	<i>Get Z-score over a margin of an img</i>
------------	--

---

**Description**

Standardizes an image either by the axial, sagittal, or coronal slice

**Usage**

```
zscore_img(img, mask = NULL, margin = 3, remove.na = TRUE,
  remove.nan = TRUE, remove.inf = TRUE, remove.val = 0)
```

**Arguments**

img	character path of image or an object of class nifti
mask	character path of mask or an object of class nifti
margin	Margin of image to z-score over (3-Axial, 2-Sagittal, 1-Coronal)
remove.na	(logical) change NAs to remove.val
remove.nan	(logical) change NaN to remove.val
remove.inf	(logical) change Inf to remove.val
remove.val	(logical) value to put the NA/NaN/Inf

**Value**

Array of object of class nifti

**See Also**

[aperm](#)

**Examples**

```
dim = c(100, 30, 5)
img = array(rnorm(prod(dim), mean=4, sd=4),
            dim=dim)

truth2 = img
for (i in 1:dim(img)[2]) {
  truth2[,i,] = (truth2[,i,] - mean(truth2[,i,]))/sd(truth2[,i,])
}

truth1 = img
for (i in 1:dim(img)[1]) {
  truth1[i,,] = (truth1[i,,] - mean(truth1[i,,]))/sd(truth1[i,,])
}

truth3 = img
for (i in 1:dim(img)[3]) {
  truth3[, ,i] = (truth3[, ,i] - mean(truth3[, ,i]))/sd(truth3[, ,i])
}
try3 = zscore_img(img, margin=3)
stopifnot(all.equal(try3, truth3))
try2 = zscore_img(img, margin=2)
stopifnot(all.equal(try2, truth2))
try1 = zscore_img(img, margin=1)
stopifnot(all.equal(try1, truth1))
```

# Index

aperm, 29

cal\_img, 3  
check\_file, 4  
check\_nifti, 5  
checking, 3  
checkout, 4, 4  
convert.bitpix, 5  
convert.datatype, 5

datatype, 5, 25

flirt, 6  
flirt.help, 6  
fslbet, 7  
fslbet.help, 7  
fslbin, 8  
fslbin.help (fslmaths.help), 14  
fsldir (Get FSLDir), 22  
fslero, 8  
fslero.help (fslmaths.help), 14  
fslfill, 9  
fslfill.help (fslmaths.help), 14  
fslhd, 10, 11  
fslhd.help, 11  
fslhd.parse, 11  
fslhelp, 12  
fslmask, 12  
fslmask.help (fslmaths.help), 14  
fslmaths, 13  
fslmaths.help, 14  
fslmerge, 14  
fslmerge.help, 15  
fslrange, 15  
fslrange.help (fslstats.help), 17  
fslsmooth, 16  
fslsmooth.help (fslmaths.help), 14  
fslstats, 17  
fslstats.help, 17  
fslsub2, 18  
fslsub2.help (fslmaths.help), 14  
fslthresh, 19  
fslthresh.help (fslmaths.help), 14  
fslval, 20  
fslval.help, 20  
fslview, 21  
fslview.help, 21

Get FSLDir, 22  
get.fsl, 22  
get.fsloutput, 22  
get.imgext, 23  
getForms, 4, 23

have.fsl, 24

melodic, 24  
melodic.help, 25

newnii, 25  
nii.stub, 26

readNIFTI, 5–10, 12–14, 16, 18, 19  
rescale\_img, 26

system, 6–10, 12–14, 16, 18, 19, 21, 24

tempimg, 27

voxdim, 27

writeNIFTI, 26, 27

zero\_trans, 28  
zscore\_img, 28