

Package ‘fslr’

September 10, 2025

Type Package

Title Wrapper Functions for 'FSL' ('FMRIB' Software Library) from
Functional MRI of the Brain ('FMRIB')

Version 2.27.0

Maintainer John Muschelli <muschelli2@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 objects of class 'nifti', implemented by
package 'oro.nifti', and the function executes an 'FSL' command and returns an R
object of class 'nifti' if desired.

Imports methods, R.utils, graphics, grDevices, stats, utils

Depends oro.nifti (>= 0.5.0), neurobase (>= 1.32.0), R (>= 3.2.0)

License GPL-3

VignetteBuilder knitr

Suggests knitr, rmarkdown, covr

BugReports <https://github.com/muschelli2/fslr/issues>

SystemRequirements FSL

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author John Muschelli [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6469-1750>>)

Repository CRAN

Date/Publication 2025-09-09 22:00:02 UTC

Contents

applytopup	6
aux.file-methods	7
bitpix-methods	7
cal.max-methods	8
cal.min-methods	8
checkout	9
check_file	9
datatype-methods	10
data_type-methods	10
descrip-methods	11
dim_-methods	11
download_fsl	12
dtifit	12
eddy	13
eddy_correct	15
enforce_form	15
face_removal_mask	16
fast	17
fast.help	18
flirt	19
flirt.help	20
flirt_apply	20
fnirt	21
fnirt.help	22
fnirt_with_affine	22
fnirt_with_affine_apply	23
fslabs.help	25
fslacos.help	25
fsladd.help	26
fsland	26
fslasin.help	27
fslatan.help	27
fslbet.help	28
fslbin.help	29
fslbinv.help	29
fslchfiletype	30
fslchfiletype.help	31
fslcmd	31
fslcog	32
fslcos.help	33
fslcpgeom	34
fslcpgeom.help	35
fsldir	35
fsldiv.help	36
fsledge.help	36
fslentropy	37

fslepi_reg	37
fslerode.help	39
fslexp.help	39
fslfill.help	40
fslfill2	40
fslgetorient	41
fslhd	42
fslhd.help	42
fslhd.parse	43
fslhelp	43
fslindex.help	44
fsllog.help	45
fslmask.help	45
fslmaths.help	46
fslmax	46
fslmean	47
fslmerge.help	47
fslmul.help	48
fslnan.help	48
fslnanm.help	49
fslor	50
fslorient	51
fslorient.help	51
fslorienter	52
fslrand.help	52
fslrandn.help	53
fslrange	53
fslrecip.help	54
fslrem.help	55
fslreorient2std	55
fslreorient2std.help	56
fslrobustfov	57
fslrobustfov.help	58
fslroi	58
fslsd	59
fslsin	60
fslsin.help	61
fslslicetimer	61
fslsmooth.help	62
fslsmooth_in_mask	63
fslsplit	64
fslsplit.help	65
fslsqr.help	65
fslsqrt.help	66
fslstats	66
fslstats.help	67
fslsub.help	67
fslsub2.help	68

fslsum	69
fslswapdim.help	69
fsltan.help	70
fslthresh.help	70
fslval	71
fslval.help	71
fslview	72
fslview.help	73
fslvol	73
fslvolume	74
fslxor	74
fsl_abs	75
fsl_acos	76
fsl_add	77
fsl_anat	78
fsl_anat.help	79
fsl_applywarp	80
fsl_applywarp.help	81
fsl_asin	81
fsl_atan	82
fsl_atlas_dir	83
fsl_avscale	83
fsl_bet	84
fsl_biascorrect	85
fsl_bin	86
fsl_binv	87
fsl_bin_tab	88
fsl_cluster	88
fsl_cos	90
fsl_data_dir	91
fsl_deface	92
fsl_dice	93
fsl_dilate	93
fsl_div	95
fsl_edge	96
fsl_erode	97
fsl_exp	98
fsl_fill	99
fsl_index	100
fsl_log	101
fsl_mask	102
fsl_maths	103
fsl_merge	104
fsl_mul	105
fsl_nan	106
fsl_nanm	107
fsl_rand	108
fsl_randn	109

fsl_recip	110
fsl_rem	111
fsl_resample	112
fsl_smooth	113
fsl_smoothest	114
fsl_sqr	115
fsl_sqrt	116
fsl_std_dir	117
fsl_sub	117
fsl_sub2	118
fsl_swapdim	119
fsl_tan	120
fsl_thresh	121
fsl_tsplot	123
fsl_version	124
get.fsl	125
get.fsloutput	126
get.imgext	126
getForms	127
get_quickshear_mask	127
have.fsl	128
intent_code-methods	129
intent_name-methods	129
intent_p1-methods	130
intent_p2-methods	130
intent_p3-methods	130
invert_xfm	131
magic-methods	132
mcflirt	132
mcflirt.help	133
melodic	133
melodic.help	134
mid_sagittal_align	135
mni_fname	136
mni_img	137
mrdefacer	137
parse_avscale	138
pixdim-methods	139
probtrackx	139
qform,character-method	142
qform_code-methods	142
readrpi	143
read_xfm	143
reverse_rpi_orient	144
rpi_orient	145
run_first_all	146
run_first_all.help	147
scl_inter-methods	147

scl_slope-methods	148
sform_code-methods	148
sizeof_hdr-methods	148
slice_code-methods	149
slice_duration-methods	149
slice_end-methods	150
slice_start-methods	150
susan	151
susan.help	152
toffset-methods	153
topup	153
vox_offset-methods	155
xfibres	155

Index	157
--------------	------------

applytopup	<i>applytopup - calling FSL applytopup</i>
------------	--

Description

A tool for applying and correcting estimated susceptibility induced distortions

Usage

```
applytopup(
  infile,
  datain,
  index,
  topup_files,
  out = NULL,
  method = c("lsr", "jac"),
  interp = c("spline", "trilinear"),
  verbose = TRUE
)
```

```
apply_topup(...)
```

```
fsl_applytopup(...)
```

Arguments

infile	list of names of input image (to be corrected)
datain	name of text file with PE directions/times
index	list of indices into –datain of the input image (to be corrected)
topup_files	name of field/movements (from topup)
out	basename for output (warped) image

method	Use jacobian modulation (jac) or least-squares resampling (lsr), default=lsr.
interp	Image interpolation model, trilinear or spline. Default spline
verbose	Print diagnostic information while running
...	arguments passed to topup if using fsl_topup

aux.file-methods	<i>Extract Image aux.file attribute</i>
------------------	---

Description

aux_file method for character types

Usage

```
## S4 method for signature 'character'
aux.file(object)
```

Arguments

object is a filename to pass to [fslval](#)

bitpix-methods	<i>Extract Image bitpix attribute</i>
----------------	---------------------------------------

Description

bitpix method for character types

Usage

```
## S4 method for signature 'character'
bitpix(object)
```

Arguments

object is a filename to pass to [fslval](#)

cal.max-methods	<i>Extract Image cal.max attribute</i>
-----------------	--

Description

cal_max method for character types

Usage

```
## S4 method for signature 'character'  
cal.max(object)
```

Arguments

object is a filename to pass to [fslval](#)

cal.min-methods	<i>Extract Image cal.min attribute</i>
-----------------	--

Description

cal_min method for character types

Usage

```
## S4 method for signature 'character'  
cal.min(object)
```

Arguments

object is a filename to pass to [fslval](#)

checkout	<i>Determine if Q and S 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

Examples

```
if (have.fsl()){  
  mnifile = file.path(fslidir(), "data", "standard",  
    "MNI152_T1_2mm.nii.gz")  
  forms = getForms(mnifile)  
  checkout(forms)  
}
```

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

Description

Checking the q/s-forms for a header

Usage

```
check_file(file, ...)
```

Arguments

file (character) filename of image to be checked
... options passed to [checking](#)

Value

result of [checkout](#)

Examples

```
library(fslr)
if (have.fsl()){
  mnifile = mni_fname("2")
  check_file(mnifile)
}
```

datatype-methods	<i>Extract Image datatype attribute</i>
------------------	---

Description

datatype method for character types

Usage

```
## S4 method for signature 'character'
datatype(object)
```

Arguments

object is a filename to pass to [fslval](#)

data_type-methods	<i>Extract Image data_type attribute</i>
-------------------	--

Description

data_type method for character types

Usage

```
## S4 method for signature 'character'
data_type(object)
```

Arguments

object is a filename to pass to [fslval](#)

descrip-methods	<i>Extract Image descrip attribute</i>
-----------------	--

Description

descrip method for character types

Usage

```
## S4 method for signature 'character'  
descrip(object)
```

Arguments

object is a filename to pass to [fslval](#)

dim_-methods	<i>Extract Image dim_ attribute</i>
--------------	-------------------------------------

Description

dim_ method for class character

Usage

```
## S4 method for signature 'character'  
dim_(object)
```

Arguments

object is a filename to pass to [fslval](#)

download_fsl	<i>Download FSL</i>
--------------	---------------------

Description

Download FSL Tarball

Usage

```
download_fsl(
  os = c("macosx", "redhat5", "redhat6", "centos5", "centos6", "debian", "ubuntu"),
  outdir = tempdir(),
  overwrite = TRUE,
  ...
)
```

Arguments

os	Operating system
outdir	Output directory for tarball
overwrite	If file.path(outdir, tarball_name) exists, should it be overwritten?
...	Arguments to pass to download.file

Value

Filename of destination file

dtifit	<i>DTI Fitting Procedure from FSL</i>
--------	---------------------------------------

Description

Calls dtifit from FSL

Usage

```
dtifit(
  infile,
  bvecs,
  bvals,
  mask = NULL,
  outprefix = NULL,
  opts = "",
  bet.opts = "",
  verbose = TRUE,
```

```

    sse = FALSE,
    save_tensor = FALSE,
    grad_image = NULL
)

```

Arguments

infile	Input filename
bvecs	b-vectors: matrix of 3 columns or filename of ASCII text file
bvals	b-values: vector of same length as number of rows of b-vectors or filename of ASCII text file
mask	Mask filename
outprefix	Output prefix
opts	Additional options for dtifit
bet.opts	Options for fslbet if mask is not supplied
verbose	print diagnostic messages
sse	Save sum of squared errors
save_tensor	Save tensor file out
grad_image	Gradient Nonlinearity Tensor file

Value

Vector of character filenames of output. See Note

Note

On successful completion of the command, the following files will be output, which are: mask - the mask used in the analysis outprefix_V1 - 1st eigenvector outprefix_V2 - 2nd eigenvector outprefix_V3 - 3rd eigenvector outprefix_L1 - 1st eigenvalue outprefix_L2 - 2nd eigenvalue outprefix_L3 - 3rd eigenvalue outprefix_MD - mean diffusivity outprefix_FA - fractional anisotropy outprefix_MO - mode of the anisotropy (oblate ~ -1; isotropic ~ 0; prolate ~ 1) outprefix_S0 - raw T2 signal with no diffusion weighting optional output If sse = TRUE, then the additional file will be present: outprefix_sse - Sum of squared error If save_tensor = TRUE, then the additional file will be present: outprefix_tensor - tensor as a 4D file in this order: Dxx,Dxy,Dxz,Dyy,Dyz,Dzz

eddy

Eddy Current Correction

Description

This function calls eddy from FSL for DTI Processing

Usage

```

eddy(
  infile,
  mask,
  acq_file,
  index_file,
  bvecs,
  bvals,
  topup = NULL,
  outfile = NULL,
  retimg = TRUE,
  opts = "",
  verbose = TRUE,
  eddy_cmd = c("eddy", "eddy_openmp", "eddy_cuda"),
  ...
)

```

Arguments

<code>infile</code>	input filename of 4D image.
<code>mask</code>	Mask filename (or class nifti)
<code>acq_file</code>	A text-file describing the acquisition parameters for the different images in <code>infile</code> . The format of this file is identical to that used by <code>topup</code> (though the parameter is called <code>--datain</code> there).
<code>index_file</code>	A text-file that determines the relationship between on the one hand the images in <code>infile</code> and on the other hand the acquisition parameters in <code>acq_file</code> .
<code>bvecs</code>	A text file with normalised vectors describing the direction of the diffusion weighting.
<code>bvals</code>	A text file with b-values describing the "amount of" diffusion weighting
<code>topup</code>	This should only be specified if you have previously run 'topup' on your data and should be the same name that you gave as an argument to the <code>--out</code> parameter when you ran <code>topup</code> , aka the base name for output files from <code>topup</code> .
<code>outfile</code>	Output file basename
<code>retimg</code>	(logical) return image of class nifti
<code>opts</code>	Additional options to pass to arguments passed to eddy
<code>verbose</code>	print diagnostic messages
<code>eddy_cmd</code>	The version of <code>eddy</code> to run.
<code>...</code>	Not currently used

Value

Result from system command currently

eddy_correct	<i>Eddy Current Correction</i>
--------------	--------------------------------

Description

This function calls eddy_correct from FSL for DTI Processing

Usage

```
eddy_correct(infile, outfile = NULL, retimg = TRUE, reference_no = 0, ...)
```

Arguments

infile	input filename of 4D image.
outfile	Output filename
retimg	(logical) return image of class nifti
reference_no	Set the volume number for the reference volume that will be used as a target to register all other volumes to. (default=0, i.e. the first volume)
...	Additional arguments passed to fslcmd

Value

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

enforce_form	<i>Enforce Either Qform or Sform is set</i>
--------------	---

Description

Enforce Either Qform or Sform is set

Usage

```
enforce_form(file, ...)
```

Arguments

file	(character) image filename or character of class nifti
...	additional arguments to pass to getForms

Value

A character filename

Examples

```
if (have_fsl()) {
  res = enforce_form(mni_fname())
}
```

face_removal_mask	Face Removal Mask
-------------------	-------------------

Description

Face Removal Mask

Usage

```
face_removal_mask(
  file,
  template = mni_fname(mm = "1"),
  face_mask = mni_face_fname(mm = "1"),
  outfile = NULL,
  dof = 12,
  cost = "mutualinfo",
  retimg = FALSE
)

deface_image(file, ...)
```

Arguments

file	input image
template	Template image to register input image to. Set to NULL (recommended) if want to use from https://github.com/poldracklab/pydeface . Alternatively, use mni_fname .
face_mask	Mask of image, in same space as template. Set to NULL (recommended) if want to use from https://github.com/poldracklab/pydeface . Alternatively, use mni_face_fname .
outfile	Output file name
dof	(numeric) degrees of freedom (default 6 - rigid body)
cost	Cost function passed to flirt
retimg	(logical) return image of class nifti
...	not used

Value

An image or filename depending on retimg

Examples

```
if (have_fsl()) {
  file = "~/Downloads/sample_T1_input.nii.gz"
  if (file.exists(file)) {
    mask = face_removal_mask(file = file,
                             template = NULL, face_mask = NULL)
    image = fslmask(file, mask)
  }
}
```

fast

FSL FAST

Description

This function calls fast from FSL

Usage

```
fast(
  file,
  outfile = NULL,
  bias_correct = TRUE,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  type = c("T1", "T2", "PD"),
  out_type = c("seg", "mixeltype", "pve_0", "pve_1", "pve_2", "pveseg"),
  verbose = TRUE,
  all_images = FALSE,
  ...
)
```

```
fast_all(..., all_images = TRUE)
```

```
fast_nobias_all(..., bias_correct = FALSE, all_images = FALSE)
```

```
fsl_fast(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

```
fslfast(...)
```

```
fsl_fast_nobias(
  ...,
  bias_correct = FALSE,
  outfile = tempfile(fileext = ".nii.gz"),
```

```

    retimg = FALSE
)

fast_nobias(..., bias_correct = FALSE)

fslfast_nobias(..., bias_correct = FALSE)

```

Arguments

file	(character) image to be manipulated
outfile	(character) resultant image name (optional)
bias_correct	(logical) if FALSE, then "--nobias" is passed to FAST. Additional options can be sent using opts, but this is the most commonly one changed.
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fast
type	type of image T1, T2, or PD.
out_type	(character) Suffix to grab from outfile. For example, output filename is paste0(outfile, "_", out_type)
verbose	(logical) print out command before running
all_images	If retimg
...	additional arguments passed to readnii .

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fast.help

FAST help

Description

This function calls fast's help

Usage

```
fast.help()
```

Value

Prints help output and returns output as character vector

Examples

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

flirt

Register using FLIRT

Description

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

Usage

```
flirt(
  infile,
  reffile,
  omat = NULL,
  dof = 6,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

<code>infile</code>	(character) input filename
<code>reffile</code>	(character) reference image to be registered to
<code>omat</code>	(character) Output matrix name
<code>dof</code>	(numeric) degrees of freedom (default 6 - rigid body)
<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 readnii .
<code>intern</code>	(logical) pass to system
<code>opts</code>	(character) additional options to FLIRT
<code>verbose</code>	(logical) print out command before running
<code>...</code>	additional arguments passed to readnii .

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

Examples

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

flirt_apply

Apply Warp from FLIRT

Description

This function applies a matrix from [flirt](#) to other images

Usage

```
flirt_apply(
  infile,
  reffile,
  initmat,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

infile	(character) input filename
reffile	(character) reference image to be registered to
initmat	(character) Matrix of transformation
outfile	(character) output filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
opts	(character) additional options to FLIRT
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

character or logical depending on intern

fnirt	<i>Register using FNIRT</i>
-------	-----------------------------

Description

This function calls `fnirt` to register `infile` to `reffile` and either saves the image or returns an object of class `nifti`

Usage

```
fnirt(  
  infile,  
  reffile,  
  outfile = NULL,  
  retimg = TRUE,  
  reorient = FALSE,  
  intern = FALSE,  
  opts = "",  
  verbose = TRUE,  
  ...  
)
```

Arguments

infile	(character) input filename
reffile	(character) reference image to be registered to
outfile	(character) output filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
opts	(character) additional options to FLIRT
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

character or logical depending on intern

fnirt.help	<i>FNIRT help</i>
------------	-------------------

Description

This function calls fnirt's help

Usage

```
fnirt.help()
```

Value

Prints help output and returns output as character vector

fnirt_with_affine	<i>Register using FNIRT, but doing Affine Registration as well</i>
-------------------	--

Description

This function calls fnirt to register infile to reffile and either saves the image or returns an object of class nifti, but does the affine registration first

Usage

```
fnirt_with_affine(
  infile,
  reffile,
  flirt.omat = NULL,
  flirt.outfile = NULL,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  flirt.opts = "",
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

infile	(character) input filename
reffile	(character) reference image to be registered to
flirt.omat	(character) Filename of output affine matrix
flirt.outfile	(character) Filename of output affine-registered image
outfile	(character) output filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
flirt.opts	(character) additional options to FLIRT
opts	(character) additional options to FNIRT
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

character or logical depending on intern

fnirt_with_affine_apply

Applies FLIRT then FNIRT transformations

Description

Applies an affine transformation with FLIRT then the warp image with FNIRT

Usage

```
fnirt_with_affine_apply(
  infile,
  reffile,
  flirt.omat = NULL,
  flirt.outfile = NULL,
  fnirt.warpfile = NULL,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  flirt.opts = "",
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

<code>infile</code>	(character) input filename
<code>reffile</code>	(character) reference image to be registered to
<code>flirt.omat</code>	(character) Filename of output affine matrix
<code>flirt.outfile</code>	(character) Filename of output affine-registered image
<code>fnirt.warpfile</code>	(character) Filename of warp image from fnirt
<code>outfile</code>	(character) output filename
<code>retimg</code>	(logical) return image of class nifti
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) pass to system
<code>flirt.opts</code>	(character) additional options to FLIRT
<code>opts</code>	(character) additional options to FNIRT
<code>verbose</code>	(logical) print out command before running
<code>...</code>	additional arguments passed to readnii .

Value

character or logical depending on `intern`

See Also

[fnirt_with_affine](#)

`fslabs.help`*fslabs Help*

Description

This function calls `fslmaths`'s help, as `fslabs` is a wrapper for `fslmaths`

Usage

```
fslabs.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

`fslacos.help`*fslacos Help*

Description

This function calls `fslmaths`'s help, as `fslacos` is a wrapper for `fslmaths`

Usage

```
fslacos.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fsladd.help

fsladd Help

Description

This function calls `fslmaths`'s help, as `fsladd` is a wrapper for `fslmaths`

Usage

```
fsladd.help(...)
```

Arguments

... passed to `fslmaths.help`

Value

Prints help output and returns output as character vector

Examples

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

fsland

Logical AND with Images using FSL

Description

This function multiplies two images using `fslmul`) after binarizing the images (using `fslbin`

Usage

```
fsland(file, file2, ...)
```

```
fsl_and(file, file2, ...)
```

Arguments

file (character) input image
file2 (character) image to be multiplied
... additional arguments passed to `fslmul`.

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fslasin.help	<i>fslasin Help</i>
--------------	---------------------

Description

This function calls `fslmaths`'s `help`, as `fslasin` is a wrapper for `fslmaths`

Usage

```
fslasin.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslatan.help	<i>fslatan Help</i>
--------------	---------------------

Description

This function calls `fslmaths`'s `help`, as `fslatan` is a wrapper for `fslmaths`

Usage

```
fslatan.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslbet.help

Help for FSL BET

Description

This function calls bet's help

Usage

```
fslbet.help(betcmd = c("bet2", "bet"))
```

Arguments

betcmd (character) Get help for bet or bet2 function

Value

Prints help output and returns output as character vector

Examples

```
if (have.fsl()){  
  fslbet.help()  
  fslbet.help("bet")  
}
```

`fslbin.help`*fslbin Help*

Description

This function calls `fslmaths`'s help, as `fslbin` is a wrapper for `fslmaths`

Usage

```
fslbin.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

`fslbinv.help`*fslbinv Help*

Description

This function calls `fslmaths`'s help, as `fslbinv` is a wrapper for `fslmaths`

Usage

```
fslbinv.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslchfiletype	<i>FSL Change file type</i>
---------------	-----------------------------

Description

This function calls `fslchfiletype`

Usage

```
fslchfiletype(
  file,
  filetype = "NIFTI_GZ",
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

<code>file</code>	(character) image to be manipulated
<code>filetype</code>	filetype to change image to
<code>outfile</code>	Output filename. If NULL, will overwrite input file
<code>retimg</code>	(logical) return image of class nifti
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>verbose</code>	(logical) print out command before running
<code>...</code>	additional arguments passed to readnii .

Value

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

fslchfiletype.help	<i>fslchfiletype help</i>
--------------------	---------------------------

Description

This function calls fslchfiletype's help

Usage

```
fslchfiletype.help()
```

Value

Prints help output and returns output as character vector

Examples

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

fslcmd	<i>FSL Command Wrapper</i>
--------	----------------------------

Description

This function calls fsl command passed to func

Usage

```
fslcmd(
  func,
  file,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  verbose = TRUE,
  samefile = FALSE,
  opts_after_outfile = FALSE,
  frontopts = "",
  no.outfile = FALSE,
  trim_front = FALSE,
  run = TRUE,
  ...
)
```

Arguments

func	(character) FSL function
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 readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to func
verbose	(logical) print out command before running
samefile	(logical) is the output the same file?
opts_after_outfile	(logical) should opts come after the outfile in the FSL command?
frontopts	(character) options/character to put in before filename
no.outfile	(logical) is there an output file in the arguments of the FSL function?
trim_front	trim the whitespace from the front of the command.
run	(logical) Should the command just be printed (if FALSE)?
...	additional arguments passed to readnii .

Value

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

fslcog	<i>Image Center of Gravity (FSL)</i>
--------	--------------------------------------

Description

Find Center of Gravity of Image from FSL

Usage

```
fslcog(img, mm = TRUE, verbose = TRUE, ts = FALSE)
```

Arguments

img	Object of class nifti, or path of file
mm	Logical if the center of gravity (COG) would be in mm (default TRUE) or voxels (FALSE)
verbose	(logical) print out command before running
ts	(logical) is the series a timeseries (4D), invoking -t option

Value

Vector of length 3 unless ts option invoked

Note

FSL uses a 0-based indexing system, which will give you a different answer compared to cog, but `fslcog(img, mm = FALSE) + 1` should be relatively close to `cog(img)`

Examples

```
if (have.fsl()){
  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))
  fslcog(img)
}
```

fslcos.help

fslcos Help

Description

This function calls `fslmaths`'s help, as `fslcos` is a wrapper for `fslmaths`

Usage

```
fslcos.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslcpgeom

*FSL Copy Geometry***Description**

This function calls `fslcpgeom`

Usage

```
fslcpgeom(
  file,
  file_with_header,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

<code>file</code>	(character) image to be manipulated
<code>file_with_header</code>	image with header to be copied over
<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 readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>
<code>verbose</code>	(logical) print out command before running
<code>...</code>	additional arguments passed to readnii .

Value

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

`fslcpgeom.help`*fslcpgeom help*

Description

This function calls fslcpgeom's help

Usage

```
fslcpgeom.help()
```

Value

Prints help output and returns output as character vector

Examples

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

`fsldir`*Get FSL's Directory*

Description

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

Usage

```
fsldir()
```

```
fsl_dir()
```

Value

Character path

fsldiv.help

fsldiv Help

Description

This function calls fslmaths's help, as fsldiv is a wrapper for fslmaths

Usage

```
fsldiv.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fsledge.help

fsledge Help

Description

This function calls fslmaths's help, as fsledge is a wrapper for fslmaths

Usage

```
fsledge.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslentropy	<i>Image Mean Entropy</i>
------------	---------------------------

Description

Estimates Mean Entropy of Image from FSL

Usage

```
fslentropy(img, nonzero = FALSE, verbose = TRUE, ts = FALSE)
```

Arguments

img	Object of class nifti, or path of file
nonzero	(logical) Should the statistic be taken over non-zero voxels
verbose	(logical) print out command before running
ts	(logical) is the series a timeseries (4D), invoking -t option

Value

Vector of unless ts option invoked, then matrix

Note

This uses option -e or -E in [fslstats](#)

fslepi_reg	<i>Register EPI images to Structural image</i>
------------	--

Description

This function calls epi_reg, designed to register EPI images (typically functional or diffusion) to structural (e.g. T1-weighted) image.

Usage

```
fslepi_reg(
  epi,
  t1,
  t1_brain,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  fmap = NULL,
```

```

    fmap_mag = NULL,
    fmap_mag_brain = NULL,
    echo_spacing = NA,
    phase_enc_dir = c("x", "y", "z", "-x", "-y", "-z"),
    weight = NULL,
    verbose = TRUE,
    opts = "",
    ...
)

fsl_epi_reg(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

epi_reg(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

```

Arguments

epi	EPI image, character or nifti object
t1	whole head T1 image , character or nifti object
t1_brain	brain extracted T1 image
outfile	output registered image filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
fmap	fieldmap image (in rad/s)
fmap_mag	fieldmap magnitude image - whole head extracted
fmap_mag_brain	fieldmap magnitude image - brain extracted
echo_spacing	Effective EPI echo spacing (sometimes called dwell time) - in seconds
phase_enc_dir	phase encoding direction, dir = x/y/z/-x/-y/-z
weight	weighting image (in T1 space)
verbose	(logical) print out command before running
opts	(character) operations to be passed to fslmaths
...	additional arguments passed to readnii .

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

`fslode.help`*fslode Help*

Description

This function calls `fslmaths`'s help, as `fslode` is a wrapper for `fslmaths`

Usage

```
fslode.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

`fslexp.help`*fslexp Help*

Description

This function calls `fslmaths`'s help, as `fslexp` is a wrapper for `fslmaths`

Usage

```
fslexp.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslfill.help

fslfill Help

Description

This function calls fslmaths's help, as fslfill is a wrapper for fslmaths

Usage

```
fslfill.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslfill2

Fill image holes with dilation then erosion

Description

This function calls fslmaths to dilate an image, then calls it again to erode it.

Usage

```
fslfill2(
  file,
  outfile = NULL,
  kopts = "",
  remove.ends = TRUE,
  refill = TRUE,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  verbose = TRUE,
  ...
)
```


Arguments

file	(character) filename of image to be filled
outfile	(character) name of resultant filled file
kopts	(character) Options passed for kernel before erosion/dilation
remove.ends	(logical) Remove top and bottom dilation.
refill	(logical) Run fslfill after dilation/erosion.
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

character or logical depending on intern

Note

This function binarizes the image before running.

fslgetorient

FSL Orientation Wrappers

Description

This function calls [fslorient -get*](#) and is a simple wrapper of [fslorient](#)

Usage

```
fslgetorient(file, verbose = TRUE)

fslgetsform(file, verbose = TRUE)

fslgetqform(file, verbose = TRUE)

fslgetsformcode(file, verbose = TRUE)

fslgetqformcode(file, verbose = TRUE)
```

Arguments

file	(character) image to be manipulated
verbose	(logical) print out command before running

Value

Result from system command, output from FSL

fslhd	<i>Get NIfTI header using FSL</i>
-------	-----------------------------------

Description

This function calls fslhd to obtain a nifti header

Usage

```
fslhd(file, opts = "", verbose = TRUE, ...)
```

Arguments

file	(character) image filename or character of class nifti
opts	(character) additional options to be passed to fslhd
verbose	(logical) print out command before running
...	options passed to checking

Value

Character of information from fslhd

Examples

```
if (have.fsl()){
  mnifile = file.path(fslhdr(), "data", "standard",
    "MNI152_T1_2mm.nii.gz")
  fslhd(mnifile)
}
```

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

Examples

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

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

Examples

```
if (have.fsl()){
  mnifile = mni_fname("2")
  hd = fslhd(mnifile)
  fslhd.parse(hd)
}
```

fslhelp	<i>Wrapper for getting fsl help</i>
---------	-------------------------------------

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

fslindex.help	<i>fslindex Help</i>
---------------	----------------------

Description

This function calls fslmaths's help, as fslindex is a wrapper for fslmaths

Usage

```
fslindex.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

`fsllog.help`*fsllog Help*

Description

This function calls `fslmaths`'s help, as `fsllog` is a wrapper for `fslmaths`

Usage

```
fsllog.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

`fslmask.help`*fslmask Help*

Description

This function calls `fslmaths`'s help, as `fslmask` is a wrapper for `fslmaths`

Usage

```
fslmask.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

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()
}
```

fslmax

Get min/max of an image

Description

This function calls the range or robust range functions from FSL and then extracts the min/max

Usage

```
fslmax(file, ...)
```

```
fslmin(file, ...)
```

Arguments

file	(character) filename of image to be checked
...	options passed to fslrange

Value

Numeric vector of mins/maxs or just one depending if ts = TRUE

Examples

```
if (have.fsl()){
  mnifile = file.path(fsldir(), "data", "standard",
    "MNI152_T1_2mm.nii.gz")
  fslmax(mnifile)
}
```

fslmean

*Image Mean***Description**

Estimates Mean of Image from FSL

Usage

```
fslmean(img, nonzero = FALSE, verbose = TRUE, ts = FALSE)
```

Arguments

img	Object of class nifti, or path of file
nonzero	(logical) Should the statistic be taken over non-zero voxels
verbose	(logical) print out command before running
ts	(logical) is the series a timeseries (4D), invoking -t option

Value

Vector of unless ts option invoked, then matrix

Note

This uses option -m or -M in [fslstats](#)

fslmerge.help

*FSLMerge help***Description**

This function calls fslmerge's help

Usage

```
fslmerge.help()
```

Value

Prints help output and returns output as character vector

Examples

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

fslmul.help

fslmul Help

Description

This function calls fslmaths's help, as fslmul is a wrapper for fslmaths

Usage

```
fslmul.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslnan.help

fslnan Help

Description

This function calls fslmaths's help, as fslnan is a wrapper for fslmaths

Usage

```
fslnan.help(...)
```


Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslnanm.help

fslnanm Help

Description

This function calls fslmaths's help, as fslnanm is a wrapper for fslmaths

Usage

```
fslnanm.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslor

*Perform OR/Union operation on Images using FSL***Description**

This function calls `fslmaths file -add file2 -bin` after binarizing `file` and `file2` using [fslbin](#).

Usage

```
fslor(
  file,
  file2,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  ...
)
```

```
fsl_or(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

Arguments

<code>file</code>	(character) input image
<code>file2</code>	(character) image to be unioned
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class nifti
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>...</code>	additional arguments passed to readnii .

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fslorient	<i>FSL Orient</i>
-----------	-------------------

Description

This function calls `fslorient`

Usage

```
fslorient(
  file,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

<code>file</code>	(character) image to be manipulated
<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 readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslorient</code>
<code>verbose</code>	(logical) print out command before running
<code>...</code>	additional arguments passed to readnii .

Value

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

<code>fslorient.help</code>	<i>fslorient help</i>
-----------------------------	-----------------------

Description

This function calls `fslorient`'s help

Usage

```
fslorient.help()
```

Value

Prints help output and returns output as character vector

Examples

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

fslorienter

Wrapper for FSL Get Orientation

Description

This function calls `fslorient -getorient` and is a simple wrapper of [fslorient](#)

Usage

```
fslorienter(file, opts = "", verbose = TRUE)
```

Arguments

file	(character) image to be manipulated
opts	option to send to fslorient
verbose	(logical) print out command before running

Value

Result from system command, output from FSL

fslrand.help

fslrand Help

Description

This function calls `fslmaths`'s help, as `fslrand` is a wrapper for `fslmaths`

Usage

```
fslrand.help(...)
```

Arguments

...	passed to fslmaths.help
-----	---

Value

Prints help output and returns output as character vector

Examples

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

fslrandn.help

fslrandn Help

Description

This function calls fslmaths's help, as fslrandn is a wrapper for fslmaths

Usage

```
fslrandn.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslrange

Get range of an image

Description

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

Usage

```
fslrange(file, robust = FALSE, verbose = TRUE, ts = FALSE, ...)
```

Arguments

file	(character) filename of image to be checked
robust	(logical) Should the range be robust (-r)
verbose	(logical) print out command before running
ts	(logical) is the series a timeseries (4D), invoking -t option
...	options passed to checking

Value

numeric vector of length 2

Examples

```
if (have.fsl()){
  mnifile = file.path(fsldir(), "data", "standard",
    "MNI152_T1_2mm.nii.gz")
  fslrange(mnifile)
}
```

fslrecip.help

fslrecip Help

Description

This function calls fslmaths's help, as fslrecip is a wrapper for fslmaths

Usage

```
fslrecip.help(...)
```

Arguments

...	passed to fslmaths.help
-----	---

Value

Prints help output and returns output as character vector

Examples

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

fslrem.help

*fslrem Help***Description**

This function calls fslmaths's help, as fslrem is a wrapper for fslmaths

Usage

```
fslrem.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslreorient2std

*FSL Orient to MNI***Description**

This function calls fslreorient2std

Usage

```
fslreorient2std(
  file,
  reting = TRUE,
  reorient = FALSE,
  intern = FALSE,
  verbose = TRUE,
  opts = "",
  ...
)

fslreorient2std_mat(
  file,
```

```

    matfile = tempfile(fileext = ".mat"),
    verbose = TRUE,
    ...
)

```

Arguments

file	(character) image to be manipulated
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
verbose	(logical) print out command before running
opts	additional options to pass to fslreorient2std
...	additional arguments passed to readnii .
matfile	Output file for the matrix for reorientation

Value

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

fslreorient2std.help *fslreorient2std help*

Description

This function calls fslreorient2std's help

Usage

```
fslreorient2std.help()
```

Value

Prints help output and returns output as character vector

Examples

```

if (have.fsl()){
  fslreorient2std.help()
}

```


Description

This function calls robustfov to automatically crop the image

Usage

```
fslrobustfov(  
  file,  
  brain_size = NULL,  
  mat_name = NULL,  
  roi_name = NULL,  
  retimg = TRUE,  
  reorient = FALSE,  
  intern = FALSE,  
  verbose = TRUE,  
  ...  
)  
  
fsl_robustfov(retimg = FALSE, ...)
```

Arguments

file	(character) image to be manipulated
brain_size	size of brain in z-dimension (default 150mm)
mat_name	matrix output name
roi_name	ROI volume output name
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

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

fslrobustfov.help	<i>FSL Robust Field of View Help</i>
-------------------	--------------------------------------

Description

This function calls robustfov help

Usage

```
fslrobustfov.help()
```

fslroi	<i>FSL ROI</i>
--------	----------------

Description

This function calls fslroi

Usage

```
fslroi(
    file,
    xmin = 0,
    xsize = -1,
    ymin = 0,
    ysize = -1,
    zmin = 0,
    zsize = -1,
    tmin = NULL,
    tsize = NULL,
    outfile = NULL,
    retimg = TRUE,
    reorient = FALSE,
    intern = FALSE,
    verbose = TRUE,
    ...
)

fsl_roi(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

fslroi_time(file, tmin = NULL, tsize = NULL, ...)
```

Arguments

file	(character) image to be manipulated
xmin	Minimum index for x-dimension
xsize	Size of ROI in x-dimension
ymin	Minimum index for y-dimension
ysize	Size of ROI in y-dimension
zmin	Minimum index for z-dimension
zsize	Size of ROI in z-dimension
tmin	Minimum index for t-dimension
tsize	Size of ROI in t-dimension
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 readnii .
intern	(logical) to be passed to system
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

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

Note

Indexing (in both time and space) starts with 0 not 1! Inputting -1 for a size will set it to the full image extent for that dimension.

fslsd	<i>Image Standard Deviation</i>
-------	---------------------------------

Description

Estimates Standard Deviation of Image from FSL

Usage

```
fslsd(img, nonzero = FALSE, verbose = TRUE, ts = FALSE)
```

Arguments

img	Object of class nifti, or path of file
nonzero	(logical) Should the statistic be taken over non-zero voxels
verbose	(logical) print out command before running
ts	(logical) is the series a timeseries (4D), invoking -t option

Value

Vector of unless ts option invoked, then matrix

Note

This uses option -s or -S in [fslstats](#)

<code>fslsin</code>	<i>Sine Transform Image using FSL</i>
---------------------	---------------------------------------

Description

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

Usage

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

Arguments

- `file` (character) input image to sine transform
- `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 [readnii](#).
- `intern` (logical) to be passed to [system](#)
- `opts` (character) operations to be passed to `fslmaths`
- `...` additional arguments passed to [readnii](#).

Value

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

fslsin.help

*fslsin Help***Description**

This function calls fslmaths's help, as fslsin is a wrapper for fslmaths

Usage

```
fslsin.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslslicetimer

*FSL Slice Timing Correction***Description**

This function calls slicetimer and performs slice timing correction for fMRI data

Usage

```
fslslicetimer(
  file,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  tr = 3,
  direction = "z",
  indexing = c("up", "down"),
  acq_order = c("contiguous", "interleaved"),
  verbose = TRUE,
  ...)
```

```
)

fsl_slicetimer(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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 readnii .
intern	(logical) to be passed to system
tr	(numeric) Repeat time in seconds
direction	(character) Direction of acquisition
indexing	(character) Whether indexing was bottom up (default) or down using --down option
acq_order	(character) Order of acquisition, either contiguous or interleaved
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fslsmooth.help

fslsmooth Help

Description

This function calls fslmaths's help, as fslsmooth is a wrapper for fslmaths

Usage

```
fslsmooth.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslsmooth_in_mask	<i>Smooth Image Within a Mask Only</i>
-------------------	--

Description

This function smooth an image within a mask and replaces the values of the original image with the smoothed values.

Usage

```
fslsmooth_in_mask(file, sigma = 10, mask = NULL, ...)

fsl_smooth_in_mask(...)
```

Arguments

file	(character) image to be smoothed
sigma	(numeric) sigma (in mm) of Gaussian kernel for smoothing
mask	(character) optional mask given for image
...	additional arguments passed to fslsmooth .

Value

Object of class nifti

Examples

```
if (have.fsl()){
  system.time({
    dims = c(50, 50, 20)
    x = array(rnorm(prod(dims)), dim = dims)
    img = nifti(x, dim= dims,
      datatype = convert.datatype()$FLOAT32, cal.min = min(x),
      cal.max = max(x), pixdim = rep(1, 4))
    mask = abs(img ) > 1
    s.img = fslsmooth_in_mask(img, mask = mask)
  })
}
```

fslsplit

*Split images using FSL***Description**

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

Usage

```
fslsplit(
  infile,
  direction = c("t", "x", "y", "z"),
  output_basename = NULL,
  retimg = TRUE,
  reorient = FALSE,
  verbose = TRUE
)

fsl_split(..., retimg = FALSE)
```

Arguments

<code>infile</code>	(character) input filename
<code>direction</code>	(character) direction to split over: t (time), x, y, z
<code>output_basename</code>	(character) prefix to have for output
<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 readnii .
<code>verbose</code>	(logical) print out command before running
<code>...</code>	not used

Value

List of output files

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

`fslsplit.help`*FSL Split help*

Description

This function calls `fslsplit`'s help

Usage

```
fslsplit.help()
```

Value

Prints help output and returns output as character vector

Examples

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

`fslsqr.help`*fslsqr Help*

Description

This function calls `fslmaths`'s help, as `fslsqr` is a wrapper for `fslmaths`

Usage

```
fslsqr.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

<code>fslsqrt.help</code>	<i>fslsqrt Help</i>
---------------------------	---------------------

Description

This function calls `fslmaths`'s help, as `fslsqrt` is a wrapper for `fslmaths`

Usage

```
fslsqrt.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

<code>fslstats</code>	<i>FSL Stats</i>
-----------------------	------------------

Description

This function calls `fslstats`

Usage

```
fslstats(file, opts = "", verbose = TRUE, ts = FALSE, ...)
```

Arguments

- `file` (character) filename of image to be checked
- `opts` (character) operation passed to `fslstats`
- `verbose` (logical) print out command before running
- `ts` (logical) is the series a timeseries (4D), invoking `-t` option
- ... options passed to [checking](#)

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()
}
```

fslsub.help

fslsub Help

Description

This function calls fslmaths's help, as fslsub is a wrapper for fslmaths

Usage

```
fslsub.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslsub2.help

fslsub2 Help

Description

This function calls fslmaths's help, as fslsub2 is a wrapper for fslmaths

Usage

```
fslsub2.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

```
if (have.fsl()){  
  fslsub2.help()  
}
```

fslsum	<i>FSL Sum</i>
--------	----------------

Description

This function calls `fslstats -M -V` to get product, aka the approximate sum.

Usage

```
fslsum(file, opts = "", ts = FALSE, ...)
```

Arguments

file	(character) filename of image to be checked
opts	Additional options to pass to fslstats
ts	(logical) is the series a timeseries (4D), invoking <code>-t</code> option
...	options passed to fslstats

Value

Numeric value

Note

This may be approximate due to rounding

fslswapdim.help	<i>fslswapdim help</i>
-----------------	------------------------

Description

This function calls `fslswapdim`'s help

Usage

```
fslswapdim.help()
```

Value

Prints help output and returns output as character vector

Examples

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

fsltan.help

fsltan Help

Description

This function calls fslmaths's help, as fsltan is a wrapper for fslmaths

Usage

```
fsltan.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

fslthresh.help

fslthresh Help

Description

This function calls fslmaths's help, as fslthresh is a wrapper for fslmaths

Usage

```
fslthresh.help(...)
```

Arguments

... passed to [fslmaths.help](#)

Value

Prints help output and returns output as character vector

Examples

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

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

Description

This function calls `fslval` to obtain a nifti header

Usage

```
fslval(file, keyword = "", verbose = TRUE, ...)
```

Arguments

<code>file</code>	(character) image filename or character of class nifti
<code>keyword</code>	(character) keyword to be taken from <code>fslhd</code>
<code>verbose</code>	(logical) print out command before running
<code>...</code>	options passed to checking

Value

Character of information from `fslhd` field specified in `keyword`

Examples

```
if (have.fsl()){
  mnifile = file.path(fsldir(), "data", "standard",
    "MNI152_T1_2mm.nii.gz")
  fslval(mnifile, keyword = "dim1")
}
```

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

Description

This function calls `fslval`'s help

Usage

```
fslval.help()
```

Value

Prints help output and returns output as character vector

Examples

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

fslview

Open image in FSLView

Description

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

Usage

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

```
fsleyes(file, intern = TRUE, opts = "", verbose = TRUE, ...)
```

Arguments

file	(character) filename of image to be thresholded
intern	(logical) pass to system
opts	(character) options for FSLView
verbose	(logical) print out command before running
...	options passed to checking

Value

character or logical depending on intern

Note

As of FSL version 5.0.10, this is deprecated: <https://fsl.fmrib.ox.ac.uk/fsl/docs/#/development/history/index>

fslview.help

*FSLView help***Description**

This function calls fslview's help

Usage

```
fslview.help()
```

Value

Prints help output and returns output as character vector

Examples

```
library(fslr)
if (have.fsl()){
  print(fsl_version())
  in_ci <- function() {
    nzchar(Sys.getenv("CI"))
  }
  if (!in_ci() && package_version(fslversion()) < package_version("6.0")) {
    fslview.help()
  }
}
```

fslvol

*FSL Volume in mL (or cubic centimeters)***Description**

This function wraps [fslsum](#) and [voxdim](#)

Usage

```
fslvol(file, ...)
```

Arguments

file	(character) filename of image to be checked
...	options passed to fslsum

Value

Numeric value of volume in mL

Note

This may be approximate due to rounding

fslvolume	<i>Image Volume</i>
-----------	---------------------

Description

Estimates Volume of Image from FSL

Usage

```
fslvolume(img, nonzero = FALSE, verbose = TRUE, ts = FALSE)
```

Arguments

- | | |
|---------|---|
| img | Object of class nifti, or path of file |
| nonzero | (logical) Should the statistic be taken over non-zero voxels |
| verbose | (logical) print out command before running |
| ts | (logical) is the series a timeseries (4D), invoking -t option |

Value

Vector of unless ts option invoked, then matrix

Note

This uses option -v or -V in [fslstats](#)

fslxor	<i>Perform XOR/Exclusive Or operation on Images using FSL</i>
--------	---

Description

This function calls `fslmaths file -add file2 -bin` after binarizing file and file2 using [fslbin](#) and then uses [fsl_thresh](#) to threshold any values greater than 1 back to zero.

Usage

```
fslxor(
  file,
  file2,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  ...
)

fsl_xor(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

Arguments

file	(character) input image
file2	(character) image to be XOR'd
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 readnii .
intern	(logical) to be passed to system
...	additional arguments passed to readnii .

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_abs	<i>Absolute Value Image using FSL</i>
---------	---------------------------------------

Description

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

Usage

```
fsl_abs(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to absolute value
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_acos	<i>Arc Cosine Transform Image using FSL</i>
----------	---

Description

This function calls fslmaths -acos. The R functions wraps fslmaths

Usage

```
fsl_acos(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to arc cosine transform
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_add

Add Images using FSL

Description

This function calls fslmaths -add. The R functions wraps fslmaths

Usage

```
fsl_add(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

fsladd(
  file,
  file2,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  ...
)
```

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image
file2	(character) image to be added
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_anat

FSL Anatomical Processing Script

Description

This function calls fsl_anat from FSL

Usage

```
fsl_anat(
  file,
  modality = c("T1", "T2", "PD"),
  outdir = NULL,
  intern = FALSE,
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

file	(character) image to be manipulated, should be full path
modality	(character) Modality of Image to be run
outdir	(character) output directory, if none specified, will default to dirname(file)
intern	(logical) to be passed to system
opts	(character) operations to be passed to fsl_anat
verbose	(logical) print out command before running
...	options passed to checking

Value

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

fsl_anat.help	<i>fsl_anat help</i>
---------------	----------------------

Description

This function calls fsl_anat's help

Usage

```
fsl_anat.help()
```

Value

Prints help output and returns output as character vector

Examples

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

fsl_applywarp

*Apply Warp from FNIRT***Description**

This function applies a coefficient map from [fnirt](#) to other images

Usage

```
fsl_applywarp(
  infile,
  reffile,
  warpfile,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

infile	(character) input filename
reffile	(character) reference image to be registered to
warpfile	(character) reference image to be registered to
outfile	(character) output filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
opts	(character) additional options to FLIRT
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

character or logical depending on intern

fsl_applywarp.help	<i>FSL applywarp help</i>
--------------------	---------------------------

Description

This function calls applywarp's help

Usage

```
fsl_applywarp.help()
```

Value

Prints help output and returns output as character vector

fsl_asin	<i>Arc Sine Transform Image using FSL</i>
----------	---

Description

This function calls fslmaths -asin. The R functions wraps fslmaths

Usage

```
fsl_asin(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to arc sine transform
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_atan

*Arc Tangent Transform Image using FSL***Description**

This function calls fslmaths -atan. The R functions wraps fslmaths

Usage

```
fsl_atan(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to arc tangent transform
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_atlas_dir

Get FSL's Standard Data Directory

Description

Finds the FSLDIR from system environment or `getOption("fsl.path")` and pastes on "data/standard"

Usage

```
fsl_atlas_dir()
```

Value

Character path

fsl_avscale

Scale Affine Matrix using avscale

Description

This function calls avscale to get individual matrices for FSL

Usage

```
fsl_avscale(file, volume = NULL, parsed = TRUE, verbose = TRUE)
```

```
avscale(...)
```

Arguments

file	(character) matrix filename
volume	(character) non-reference volume filename or nifti image
parsed	(logical) should parse_avscale be run after?
verbose	(logical) print out command before running
...	not used, but used for duplicating avscale as alias

Value

Character of information from avscale

fsl_bet

*Use FSL's Brain Extraction Tool (BET)***Description**

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

Usage

```
fsl_bet(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) output filename
retimg	(logical) return image of class nifti
infile	(character) input filename
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
opts	(character) additional options to bet
betcmd	(character) Use bet or bet2 function
verbose	(logical) print out command before running

Value

character or logical depending on intern

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_biascorrect	<i>FSL Bias Correct</i>
-----------------	-------------------------

Description

This function wraps a call to fast that performs bias correction

Usage

```
fsl_biascorrect(  
  file,  
  outfile = NULL,  
  retimg = TRUE,  
  reorient = FALSE,  
  intern = FALSE,  
  opts = "",  
  verbose = TRUE,  
  remove.seg = TRUE,  
  ...  
)
```

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 readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fast
verbose	(logical) print out command before running
remove.seg	(logical) Should segmentation from FAST be removed?
...	additional arguments passed to readnii .

Value

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

fsl_bin

*Binarize Image using FSL***Description**

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

Usage

```
fsl_bin(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

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

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

Examples

```

set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
nim = oro.nifti::nifti(arr)
if (have.fsl()){
  fslbin(nim)
  fsl_bin(nim)
}

```

fsl_binv

*Binarized Inverse Image using FSL***Description**

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

Usage

```
fsl_binv(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

```

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

```

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class nifti
<code>file</code>	(character) input image to take the binarized inverse
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_bin_tab	<i>Quick Tabulation for logical images</i>
-------------	--

Description

Creates a 2 by 2 table for

Usage

```
fsl_bin_tab(x, y, dnames = c("x", "y"), verbose = FALSE)
```

Arguments

- x filename of logical or 0/1 image
- y filename of logical or 0/1 image
- dnames names for table
- verbose Should fsl commands be printed?

Value

table of x vs y

Note

fsl_bin will be run to make these images binary before running

fsl_cluster	<i>Form clusters, report information about clusters and/or perform cluster-based inference. Wrapper for cluster</i>
-------------	---

Description

Form clusters, report information about clusters and/or perform cluster-based inference. Wrapper for cluster

Usage

```
fsl_cluster(
  file,
  threshold,
  retimg = FALSE,
  reorient = FALSE,
  opts = "",
  cope_image = NULL,
  pthresh = NULL,
  peakdist = 0,
  volume = FALSE,
  smooth_est = NULL,
  voxel_resel = NULL,
  fractional = FALSE,
  connectivity = 26,
  mm = FALSE,
  find_minima = FALSE,
  standard_image = NULL,
  verbose = TRUE,
  ...
)

fslcluster(..., retimg = TRUE)

read_cluster_table(file)
```

Arguments

file	filename of input volume
threshold	threshold for input volume
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
opts	(character) operations to be passed to cluster
cope_image	filename of input cope volume
pthresh	p-threshold
peakdist	minimum distance between local maxima/minima, in mm (default 0)
volume	number of voxels in the mask
smooth_est	smoothness estimate = $\sqrt{\det(\text{Lambda})}$
voxel_resel	Size of one resel in voxel units
fractional	interprets the threshold as a fraction of the robust range
connectivity	the connectivity of voxels (default 26)
mm	use mm, not voxel, coordinates
find_minima	find minima instead of maxima
standard_image	filename for standard-space volume

verbose (logical) print out command before running
 ... additional arguments to pass to `fslcmd`

Value

A list of filenames of outputs and tables:

- opvals filename for image output of log pvals
- oindex filename for output of cluster index (in size order)
- othresh filename for output of thresholded image
- olmax filename for output of local maxima text file
- olmaxim filename for output of local maxima volume
- osize filename for output of size image
- omax filename for output of max image
- omean filename for output of mean image

Examples

```
if (have_fsl()) {
  file = mni_fname(brain = TRUE, mask = FALSE)
  threshold = 6000
  clus = fsl_cluster(file, threshold)
}
```

fsl_cos

Cosine Transform Image using FSL

Description

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

Usage

```
fsl_cos(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to cosine transform
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_data_dir	<i>Get FSL's Data Directory</i>
--------------	---------------------------------

Description

Finds the FSLDIR from system environment or `getOption("fsl.path")` and pastes on "data"

Usage

```
fsl_data_dir()
```

Value

Character path

fsl_deface

*Tool to deface a structural T1w image.***Description**

Tool to deface a structural T1w image.

Usage

```
fsl_deface(
    file,
    outfile = NULL,
    retimg = TRUE,
    opts = "",
    deface_cropped = FALSE,
    bet_fractional_intensity = NULL,
    bias_correct = FALSE,
    shift_xyz = NULL,
    cog_xyz = NULL,
    reorient = FALSE,
    intern = FALSE,
    verbose = TRUE,
    ...
)
```

Arguments

file	(character) input image to estimate edge strength
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
opts	(character) operations to be passed to fsl_deface
deface_cropped	apply the defacing to the cropped image instead of the original image
bet_fractional_intensity	fractional intensity for bet (0->1); default=0.5;
bias_correct	Bias-correct the input image (with fast);
shift_xyz	Shift, in mm, x-, y- and z-directions, to shift face mask by;
cog_xyz	centre-of-gravity for bet (voxels, not mm);
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
verbose	print diagnostic messages
...	additional arguments passed to fslcmd .

Examples

```
if (have_fsl()) {  
  file = mni_fname(mm = 1, brain = FALSE)  
  out = fsl_deface(file, retimg = FALSE)  
}
```

fsl_dice	Calculate Dice Coefficient of 2 Binary images
----------	---

Description

Creates a 2 by 2 table for

Usage

```
fsl_dice(x, y, ...)
```

Arguments

- x filename of logical or 0/1 image
- y filename of logical or 0/1 image
- ... arguments passed to [fsl_bin_tab](#)

Value

Single number of the dice coefficient

fsl_dilate	Dilate image using FSL
------------	------------------------

Description

This function calls `fslmaths -ero` after inverting the image to dilate 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

```
fsl_dilate(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant dilated image name
retimg	(logical) return image of class nifti
file	(character) image to be dilated
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
kopts	(character) options for kernel
opts	(character) additional options to be passed to fslmaths
verbose	(logical) print out command before running

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

Examples

```
if (have.fsl()){
  system.time({
    dims = c(50, 50, 20)
    x = array(rnorm(prod(dims)), dim = dims)
    img = nifti(x, dim= dims,
    datatype = convert.datatype())$FLOAT32, cal.min = min(x),
    cal.max = max(x), pixdim = rep(1, 4))
    mask = img > .5
```

```
dilated = fsldilate(mask, kopts = "-kernel boxv 5", retimg=TRUE)
})
}
```

fsl_div

*Divide Images using FSL***Description**

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

Usage

```
fsl_div(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

fsldiv(
  file,
  file2,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  ...
)
```

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class nifti
<code>file</code>	(character) input image
<code>file2</code>	(character) image to be divided
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_edge

*Edge Strength Image using FSL***Description**

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

Usage

```
fsl_edge(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>file</code>	(character) input image to estimate edge strength
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_erode	<i>Erode image using FSL</i>
-----------	------------------------------

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

```
fsl_erode(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant eroded image name
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>file</code>	(character) image to be eroded
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>kopts</code>	(character) options for kernel
<code>opts</code>	(character) additional options to be passed to <code>fslmaths</code>
<code>verbose</code>	(logical) print out command before running

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

Examples

```
if (have.fsl()){
  system.time({
    dims = c(50, 50, 20)
    x = array(rnorm(prod(dims)), dim = dims)
    img = nifti(x, dim= dims,
    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)
  })
}
```

fsl_exp

Exponentiate Image using FSL

Description

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

Usage

```
fsl_exp(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

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

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_fill	<i>Fill image holes</i>
----------	-------------------------

Description

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

Usage

```
fsl_fill(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) name of resultant filled file
retimg	(logical) return image of class nifti
file	(character) filename of image to be filled
bin	(logical) binarize the image before filling
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
verbose	(logical) print out command before running

Value

character or logical depending on intern

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

Examples

```
if (have.fsl()){
  system.time({
    dims = c(50, 50, 20)
    x = array(rnorm(prod(dims)), dim = dims)
    img = nifti(x, dim= dims,
    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)
  })
}
```

fsl_index

*Index Image using FSL***Description**

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

Usage

```
fsl_index(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class nifti
<code>file</code>	(character) input image to have non-zero entries replaced with index
<code>reorient</code>	(logical) If retimg, should file be reoriented when read in? Passed to readnii .

intern (logical) to be passed to [system](#)
 opts (character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_log	<i>Log Transform Image using FSL</i>
---------	--------------------------------------

Description

This function calls fslmaths -log. The R functions wraps fslmaths

Usage

```
fsl_log(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

... additional arguments passed to [readnii](#).
 outfile (character) resultant image name (optional)
 retimg (logical) return image of class nifti
 file (character) input image to log transform
 reorient (logical) If retimg, should file be reoriented when read in? Passed to [readnii](#).
 intern (logical) to be passed to [system](#)
 opts (character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_mask

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

```
fsl_mask(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant masked image name
retimg	(logical) return image of class nifti
file	(character) image to be masked
mask	(character) mask given for image
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) additional options to be passed to fslmask
verbose	(logical) print out command before running

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

Examples

```
if (have.fsl()){
  system.time({
    x = array(rnorm(1e5), dim = c(100, 100, 10))
    img = nifti(x, dim= c(100, 100, 10),
    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)
  })
}
```

fsl_maths

*FSL Maths***Description**

This function calls fslmaths

Usage

```
fsl_maths(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

... additional arguments passed to [readnii](#).
 outfile (character) resultant image name (optional)

retimg	(logical) return image of class nifti
file	(character) image to be manipulated
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths
verbose	(logical) print out command before running

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_merge	<i>Merge images using FSL</i>
-----------	-------------------------------

Description

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

Usage

```
fsl_merge(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) output filename
retimg	(logical) return image of class nifti

infiles	(character) input filenames
direction	(character) direction to merge over, x, y, z, t (time), a (auto)
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
verbose	(logical) print out command before running

Value

character or logical depending on intern

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_mul	<i>Multiply Images using FSL</i>
---------	----------------------------------

Description

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

Usage

```
fsl_mul(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

fslmul(
  file,
  file2,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  ...
)
```

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image
file2	(character) image to be multiplied
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_nan	<i>Replace NaNs in Image using FSL</i>
---------	--

Description

This function calls fslmaths -nan. The R functions wraps fslmaths

Usage

```
fsl_nan(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to replace NaNs (improper numbers) with 0
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_nanm

*Mask NaNs in Image using FSL***Description**

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

Usage

```
fsl_nanm(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class nifti
<code>file</code>	(character) input image to set to 1 for NaN voxels, 0 otherwise
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_rand

*Add Random Uniform Noise Image using FSL***Description**

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

Usage

```
fsl_rand(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>file</code>	(character) input image to add random uniform noise to
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_randn

*Add Random Standard Gaussian Noise Image using FSL***Description**

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

Usage

```
fsl_randn(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>file</code>	(character) input image to add random standard to Gaussian noise
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_recip

*Reciprocal Image using FSL***Description**

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

Usage

```
fsl_recip(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>file</code>	(character) input image to take the reciprocal (1/image)
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_rem

*Modulus Remainder of 2 Images using FSL***Description**

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

Usage

```
fsl_rem(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

fslrem(
  file,
  file2,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  ...
)
```

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant image name (optional)
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>file</code>	(character) input image
<code>file2</code>	(character) image to divide the current image by and take remainder
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>opts</code>	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_resample

*Resample an Image to Specific Voxel Size***Description**

Resample an Image to Specific Voxel Size

Usage

```
fsl_resample(
  file,
  voxel_size,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  opts = NULL,
  verbose = TRUE
)
```

Arguments

file	Input file to resample
voxel_size	Voxel size (in mm). This should be a scalar number.
outfile	(character) output filename
retimg	(logical) return image of class nifti
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
opts	options to pass to flirt
verbose	(logical) print out command before running

Value

If retimg then object of class nifti. Otherwise, the output file.

Examples

```
if (have_fsl()) {
  file = mni_fname(mm = 1, brain = TRUE)
  est2 = fsl_resample(file = file, voxel_size = 1, retimg = FALSE)
  pixdim(est2)
  est = fsl_resample(file = file, voxel_size = 1)
  pixdim(est)
}
```


fsl_smooth

*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

```
fsl_smooth(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

```
fslsmooth(
  file,
  sigma = 10,
  mask = NULL,
  smooth_mask = TRUE,
  smoothed_mask = NULL,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

<code>...</code>	additional arguments passed to readnii .
<code>outfile</code>	(character) resultant smoothed image name (optional) if not give, will be the stub of the filename then <code>_sigma</code>
<code>retimg</code>	(logical) return image of class <code>nifti</code>
<code>file</code>	(character or <code>nifti</code>) 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>smooth_mask</code>	(logical) Smooth mask? If TRUE, the masked image will be divided by the smoothed mask.
<code>smoothed_mask</code>	(character or <code>nifti</code>) If specified and <code>smooth_mask = TRUE</code> , then will use this as the smoothed mask for division.
<code>reorient</code>	(logical) If <code>retimg</code> , should file be reoriented when read in? Passed to readnii .
<code>intern</code>	(logical) to be passed to system
<code>verbose</code>	(logical) print out command before running

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

Examples

```
if (have.fsl()){
  system.time({
    dims = c(50, 50, 20)
    x = array(rnorm(prod(dims)), dim = dims)
    img = nifti(x, dim= dims,
    datatype = convert.datatype()$FLOAT32, cal.min = min(x),
    cal.max = max(x), pixdim = rep(1, 4))
    s.img = fslsmooth(img, retimg=TRUE)
  })
}
```

fsl_smoothest	<i>Smoothness Estimation using smoothest</i>
---------------	--

Description

Smoothness Estimation using smoothest

Usage

```
fsl_smoothest(
  file,
  residual_image,
  z_image,
  dof = NULL,
  opts = "",
  verbose = TRUE,
  ...
)
```

Arguments

- file filename of input brain mask
- residual_image 4d residual image. If specified, then dof must be specified.
- z_image z-statistic image. Cannot be specified if residual_image is specified
- dof number of degrees of freedom
- opts (character) operations to be passed to smoothest

verbose (logical) print out command before running
 ... additional arguments to pass to [fslcmd](#)

Value

An output of smoothness estimate

Examples

```
if (have_fsl()) {
  file = mni_fname(mm = 2, brain = TRUE, mask = TRUE)
  img = mni_img(mm = 2, brain = TRUE, mask = FALSE)
  mask = mni_img(mm = 2, brain = TRUE, mask = TRUE)
  img = zscore_img(img = img, mask = mask)
  est = fsl_smoothest(file = file, z_image = img)
}
```

fsl_sqr	<i>Square Image using FSL</i>
---------	-------------------------------

Description

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

Usage

```
fsl_sqr(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

... additional arguments passed to [readnii](#).
 outfile (character) resultant image name (optional)
 retimg (logical) return image of class nifti
 file (character) input image to square
 reorient (logical) If retimg, should file be reoriented when read in? Passed to [readnii](#).
 intern (logical) to be passed to [system](#)
 opts (character) operations to be passed to `fslmaths`

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_sqrt	<i>Square Root Image using FSL</i>
----------	------------------------------------

Description

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

Usage

```
fsl_sqrt(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to square root
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to <code>fslmaths</code>

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_std_dir

Get FSL's Standard Data Directory

Description

Finds the FSLDIR from system environment or `getOption("fsl.path")` and pastes on "data/standard"

Usage

```
fsl_std_dir()

fsl_std_file(file = NULL)
```

Arguments

file A file from the standard data file

Value

Character path

fsl_sub

Subtract Images using FSL

Description

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

Usage

```
fsl_sub(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

fslsub(
  file,
  file2,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  opts = "",
  ...
)
```

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image
file2	(character) image to be subtracted
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_sub2	<i>Subsample image by factor of 2</i>
----------	---------------------------------------

Description

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

Usage

```
fsl_sub2(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) name of resultant subsampled file
retimg	(logical) return image of class nifti
file	(character) filename of image to be subsampled
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) pass to system
verbose	(logical) print out command before running

Value

character or logical depending on intern

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

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))
  })
}
```

fsl_swapdim

FSL Swap Dimensions

Description

This function calls fslswapdim

Usage

```
fsl_swapdim(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

fslswapdim(
  file,
  outfile = NULL,
  retimg = TRUE,
```

```

    reorient = FALSE,
    intern = FALSE,
    a = "x",
    b = "y",
    c = "z",
    verbose = TRUE,
    ...
  )

```

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) image to be manipulated
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
a	(character) Option for x domain in fslswapdim
b	(character) Option for y domain in fslswapdim
c	(character) Option for z domain in fslswapdim
verbose	(logical) print out command before running

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_tan

Tangent Transform Image using FSL

Description

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

Usage

```
fsl_tan(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)

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

Arguments

...	additional arguments passed to readnii .
outfile	(character) resultant image name (optional)
retimg	(logical) return image of class nifti
file	(character) input image to tangent transform
reorient	(logical) If retimg, should file be reoriented when read in? Passed to readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to fslmaths

Value

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

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

fsl_thresh	<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

```
fsl_thresh(..., outfile = tempfile(fileext = ".nii.gz"), retimg = FALSE)
```

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

Arguments

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

Value

character or logical depending on intern

Note

Functions with underscores have different defaults and will return an output filename, so to be used for piping

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)
```

```

}))
}

```

fsl_tsplot

FSL Timeseries Plot using 'fsl_tsplot' (not 'tsplot')

Description

FSL Timeseries Plot using 'fsl_tsplot' (not 'tsplot')

Usage

```

fsl_tsplot(
  infile,
  outfile = tempfile(fileext = ".png"),
  plot_title = NULL,
  legend = NULL,
  labels = NULL,
  ymin = NULL,
  ymax = NULL,
  xlabel = NULL,
  ylabel = NULL,
  height = NULL,
  width = NULL,
  precision = NULL,
  unit = NULL,
  scientific_notation = FALSE,
  start_position = NULL,
  end_position = NULL,
  ...
)

fsl_tsplot.help()

```

Arguments

infile	comma-separated list of input file names (ASCII text matrix, one column per timecourse)
outfile	output filename for the PNG file
plot_title	plot title
legend	file name of ASCII text file, one row per legend entry
labels	comma-separated list of labels
ymin	minimum y-value
ymax	maximum y-value
xlabel	X-axis label

ylabel	Y-axis label
height	plot height in pixels (default 150)
width	plot width in pixels (default 600)
precision	precision of x-axis labels
unit	scaling units for x-axis (default 1...length of infile)
scientific_notation	switch on scientific notation
start_position	Position of first column to plot
end_position	Position of final column to plot
...	additional options to pass to fslcmd

Value

Name of PNG file

<code>fsl_version</code>	<i>Find FSL Version</i>
--------------------------	-------------------------

Description

Finds the FSL version from FSLDIR/etc/fslversion

Usage

```
fsl_version(full = FALSE)

fslversion()

fsl_version_gt5()
```

Arguments

full provide the full version, versus the numeric version

Value

If the version file does not exist, it will throw a warning, but it will return an empty string. Otherwise it will be a string of the version.

Note

This will use `fsldir()` to get the directory

Examples

```
if (have_fsl()) {  
    fslversion()  
    fsl_version()  
}
```

get.fsl

Create command declaring FSLDIR

Description

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

Usage

```
get.fsl(add_bin = TRUE)
```

```
get_fsl(add_bin = TRUE)
```

Arguments

add_bin	Should bin be added to the fsl path? All executables are assumed to be in FSLDIR/bin/. If not, and add_bin = FALSE, they will be assumed to be in FSLDIR/.
---------	--

Value

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

Note

This will use `Sys.getenv("FSLDIR")` before `getOption("fsl.path")`. If the directory is not found for FSL in `Sys.getenv("FSLDIR")` and `getOption("fsl.path")`, it will try the default directory `/usr/local/fsl`.

<code>get.fsloutput</code>	<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

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

Description

Runs `get.fsloutput()` 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, verbose = FALSE, ...)
```

Arguments

file	(character) filename of image to pass to header
verbose	(logical) passed to fslhd
...	options passed to checking

Value

list with elements of sform and qform and their respective codes

Examples

```
if (have.fsl()){
  mnifile = mni_fname("2")
  getForms(mnifile)
}
```

get_quickshear_mask	<i>Face Removal Mask using "Quickshear Defacing for Neuroimages" (Schimke et al. 2011)</i>
---------------------	--

Description

Face Removal Mask using "Quickshear Defacing for Neuroimages" (Schimke et al. 2011)

Usage

```
get_quickshear_mask(brain_mask, buffer = 10, verbose = TRUE)

quickshear_deface_image(
  file,
  brain_mask = NULL,
  buffer = 10,
  verbose = TRUE,
  ...
)
```

Arguments

brain_mask	Brain mask image. If NULL, then <code>fslbet</code> will be run
buffer	buffer to add to intercept for face mask equation
verbose	print diagnostic messages
file	input image - same orientation as brain mask
...	additional arguments passed to <code>fslmask</code>

Value

A binary image of the non-face areas

Note

adapted from <https://github.com/nipy/quickshear/blob/master/quickshear.py>

Examples

```
if (have_fsl()) {
  file = "~/Downloads/sample_T1_input.nii.gz"
  if (file.exists(file)) {
    res = quickshear_deface_image(file)
    brain_mask = fslbet(file) > 0
    mask = get_quickshear_mask(brain_mask)
    image = fslmask(file, mask)
  }
}
```

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(...)
```

```
have_fsl(...)
```

Arguments

... options to pass to `get.fsl`

Value

Logical TRUE is FSL is accessible, FALSE if not

Examples

```
have.fsl()
```

intent_code-methods	<i>Extract Image intent_code attribute</i>
---------------------	--

Description

intent_code method for character types

Usage

```
## S4 method for signature 'character'  
intent_code(object)
```

Arguments

object is a filename to pass to [fslval](#)

intent_name-methods	<i>Extract Image intent_name attribute</i>
---------------------	--

Description

intent_name method for character types

Usage

```
## S4 method for signature 'character'  
intent_name(object)
```

Arguments

object is a filename to pass to [fslval](#)

intent_p1-methods	<i>Extract Image intent_p1 attribute</i>
-------------------	--

Description

intent_p1 method for character types

Usage

```
## S4 method for signature 'character'
intent_p1(object)
```

Arguments

object is a filename to pass to [fslval](#)

intent_p2-methods	<i>Extract Image intent_p2 attribute</i>
-------------------	--

Description

intent_p2 method for character types

Usage

```
## S4 method for signature 'character'
intent_p2(object)
```

Arguments

object is a filename to pass to [fslval](#)

intent_p3-methods	<i>Extract Image intent_p3 attribute</i>
-------------------	--

Description

intent_p3 method for character types

Usage

```
## S4 method for signature 'character'
intent_p3(object)
```

Arguments

object is a filename to pass to [fslval](#)

invert_xfm	<i>Convert a Transformation</i>
------------	---------------------------------

Description

Convert a Transformation

Usage

```
invert_xfm(inmat, omat = tempfile(fileext = ".mat"), verbose = TRUE)

concat_xfm(inmat, inmat2, omat = tempfile(fileext = ".mat"), verbose = TRUE)

fixscaleskew_xfm(
  inmat,
  inmat2,
  omat = tempfile(fileext = ".mat"),
  verbose = TRUE
)
```

Arguments

inmat	input matrix transformation
omat	output matrix transformation
verbose	print diagnostic messages
inmat2	second matrix filename to be concatenated or fixscaleskew to first

Value

A filename of the output matrix file

Examples

```
if (have_fsl()) {
  img = mni_fname()
  mat = fslreorient2std_mat(img)
  inverted = invert_xfm(mat)
  readLines(inverted)
  catted = concat_xfm(mat, mat)
  readLines(catted)
  fixed = fixscaleskew_xfm(mat, mat)
  readLines(fixed)
}
```

magic-methods	<i>Extract Image magic attribute</i>
---------------	--------------------------------------

Description

magic method for character types

Usage

```
## S4 method for signature 'character'
magic(object)
```

Arguments

object	is a filename to pass to fslval
--------	---

mcflirt	<i>FSL Motion Correction</i>
---------	------------------------------

Description

This function calls mcflirt

Usage

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

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 readnii .
intern	(logical) to be passed to system
opts	(character) operations to be passed to mcflirt. Cannot use -o or -verbose, as output file should be specified in outfile.
verbose	(logical) print out command before running
...	additional arguments passed to readnii .

Value

If retimg then object of class nifti. Otherwise, it will have additional attributes in the additional_files field.

mcflirt.help

MCFLIRT help

Description

This function calls mcflirt's help

Usage

```
mcflirt.help()
```

Value

Prints help output and returns output as character vector

Examples

```
library(fslr)
if (have.fsl()){
  mcflirt.help()
}
```

melodic

Run MELODIC ICA

Description

This function calls melodic

Usage

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

Arguments

file	(character) image to be run
outdir	(character) output directory. (Default dirname(file))
intern	(logical) pass to system
opts	(character) options for melodic
verbose	(logical) print out command before running
...	arguments passed to checking

Value

character or logical depending on intern

melodic.help

MELODIC help

Description

This function calls melodic's help

Usage

```
melodic.help()
```

Value

Prints help output and returns output as character vector

Examples

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

mid_sagittal_align	<i>Mid-Sagittal Plane Alignment</i>
--------------------	-------------------------------------

Description

This function takes in an image, flips the image over the left/right plane, registers that flipped image to the original image, then applies the half transformation

Usage

```
mid_sagittal_align(
    file,
    opts = "",
    translation = TRUE,
    force_rpi = TRUE,
    verbose = TRUE
)

apply_mid_sagittal_align(
    file,
    file_mat,
    apply_opts = "",
    force_rpi = TRUE,
    verbose = TRUE
)
```

Arguments

file	(character) input filename or class nifti
opts	(character) options passed to flirt
translation	(logical) should the translation parameters be preserved (TRUE) or set to zero (FALSE)
force_rpi	Should rpi_orient_file be run?
verbose	(logical) print diagnostic messages
file_mat	file name of mat file for half transform from [mid_sagittal_align]
apply_opts	options to pass to [fslr::flirt_apply]

Value

Filename of output or nifti depending on ret ing

Examples

```
if (have.fsl()){
  mnifile = file.path(fsldir(), "data", "standard",
    "MNI152_T1_2mm.nii.gz")
  aligned = mid_sagittal_align(mnifile)
  thresh = readnii(mnifile) > 0
  file_mat = attr(aligned, "half_transform")
  force_rpi = attr(aligned, "force_rpi")
  flipped_thresh = apply_mid_sagittal_align(
    file = thresh,
    file_mat = file_mat,
    apply_opts = "-interp nearestneighbour",
    force_rpi = force_rpi
  )
}
```

mni_fname	<i>Construct MNI Filename</i>
-----------	-------------------------------

Description

Finds the standard data directory for FSL and pastes together the string for an MNI template image

Usage

```
mni_fname(mm = c("1", "0.5", "2"), brain = FALSE, linear = FALSE, mask = FALSE)

mni_face_fname(mm = c("1", "0.5", "2"))
```

Arguments

mm	Resolution (in mm) of the brain image (isotropic)
brain	Should the brain be returned (default) or the T1 with the skull
linear	Should the linearized MNI template be used
mask	should the mask be given? Generally, only MNI152_T1_1mm_brain_mask exists.

Value

Character path of filename, warning if that file does not exist

`mni_img`*Read MNI Filename*

Description

Simple wrapper for reading in the MNI image constructed from `mni_fname`

Usage

```
mni_img(...)
```

Arguments

... Arguments passed to `mni_fname`

Value

Object of class `nifti`

`mrdefacer`*MRI Defacer*

Description

MRI Defacer

Usage

```
mrdefacer(file, ..., verbose = TRUE)
```

```
get_mrdefacer_mask(  
  file,  
  brain_mask = NULL,  
  bet_opts = "-f 0.5",  
  search_radius = 90,  
  opts = NULL,  
  template_brain = NULL,  
  template_brain_weight = NULL,  
  template_biometric_mask = NULL,  
  verbose = TRUE  
)
```

Arguments

file	input file image to remove face/ears
...	not used
verbose	print diagnostic messages. If > 1, more verbose
brain_mask	brain mask of file. If NULL, <code>fslbet</code> will be applied
bet_opts	options to pass to <code>fslbet</code> if applied
search_radius	search radius option to pass to <code>flirt</code>
opts	additional options to pass to <code>flirt</code>
template_brain	template brain image, may be NULL
template_brain_weight	template brain weight image, used for registration may be NULL
template_biometric_mask	template biometric mask. Everything that is wanted should be 1, may be NULL

Value

A character filename of the output image

Note

Adapted from <https://github.com/mih/mridefacer>

Examples

```
if (have_fsl()) {
  file = "~/Downloads/sample_T1_input.nii.gz"
  if (file.exists(file)) {
    res = mridefacer(file)
  }
}
```

parse_avscale	<i>Parse output from avscale</i>
---------------	----------------------------------

Description

This function parses the output from `fsl_avscale` into something more manageable

Usage

```
parse_avscale(av_out)
```

Arguments

av_out	output from <code>fsl_avscale</code> , character vector
--------	---

Value

List of output values

pixdim-methods	<i>Extract Image pixdim attribute</i>
----------------	---------------------------------------

Description

Gets pixdim from a character

Usage

```
## S4 method for signature 'character'
pixdim(object)
```

Arguments

object is a filename to pass to [fslval](#)

probtrackx	<i>Probabilistic diffusion tractography with multiple fibre orientations</i>
------------	--

Description

This function wraps probtrackx from FSL

Usage

```
probtrackx(
  samples = "merged",
  mask,
  seed,
  outdir = "fdt_paths",
  verbose = TRUE,
  mode = NULL,
  targetmasks = NULL,
  mask2 = NULL,
  waypoints = NULL,
  network = FALSE,
  mesh = NULL,
  seedref = NULL,
  dir = FALSE,
  forcedir = FALSE,
  opd = FALSE,
  pd = FALSE,
```

```

os2t = FALSE,
avoid = NULL,
stop = NULL,
xfm = NULL,
invxfm = NULL,
nsamples = 5000,
nsteps = 2000,
distthresh = 0,
cthr = 0.2,
fibthresh = 0.01,
sampvox = FALSE,
steplength = 0.5,
loopcheck = FALSE,
usef = FALSE,
randfib = c(0, 1, 2, 3),
fibst = 1,
modeuler = FALSE,
rseed = NULL,
s2tastext = FALSE,
opts = ""
)

```

Arguments

samples	(nifti/character) Basename for samples files
mask	(nifti/character) Bet binary mask file in diffusion space
seed	(nifti/character) Seed volume, or voxel, or ascii file with multiple volumes, or freesurfer label file
outdir	(character) Output file (default='fdt_paths')
verbose	(logical/numeric) Verbose level, [0-2]
mode	(character) Use -mode=simple for single seed voxel
targetmasks	(character) File containing a list of target masks - required for seeds_to_targets classification
mask2	(nifti/character) Second mask in twomask_symm mode.
waypoints	(nifti/character) Waypoint mask or ascii list of waypoint masks - only keep paths going through ALL the masks
network	(logical) Activate network mode - only keep paths going through at least one seed mask (required if multiple seed masks)
mesh	(character) Freesurfer-type surface descriptor (in ascii format)
seedref	(nifti/character) Reference vol to define seed space in simple mode - diffusion space assumed if absent
dir	(logical) Directory to put the final volumes in - code makes this directory - default='logdir'
forcedir	(logical) Use the actual directory name given - i.e. don't add + to make a new directory

opd	(logical) Output path distribution
pd	(logical) Correct path distribution for the length of the pathways
os2t	(logical) Output seeds to targets
avoid	(nifti/character) Reject pathways passing through locations given by this mask
stop	(nifti/character) Stop tracking at locations given by this mask file
xfm	(character) Transform taking seed space to DTI space (either FLIRT matrix or FNIRT warpfield) - default is identity
invxfm	(character) Transform taking DTI space to seed space (compulsory when using a warpfield for seeds_to_dti)
nsamples	(numeric) Number of samples - default=5000
nsteps	(numeric) Number of steps per sample - default=2000
distthresh	(numeric) Discards samples shorter than this threshold (in mm - default=0)
cthr	(numeric) Curvature threshold - default=0.2
fibthresh	(numeric) Volume fraction before subsidiary fibre orientations are considered - default=0.01
sampvox	(logical) Sample random points within seed voxels
steplength	(numeric) Steplength in mm - default=0.5
loopcheck	(logical) Perform loopchecks on paths - slower, but allows lower curvature threshold
usef	(logical) Use anisotropy to constrain tracking
randfib	(numeric) Default 0. Set to 1 to randomly sample initial fibres (with $f > \text{fibthresh}$). Set to 2 to sample in proportion fibres (with $f > \text{fibthresh}$) to f . Set to 3 to sample ALL populations at random (even if $f < \text{fibthresh}$)
fibst	(numeric) Force a starting fibre for tracking - default=1, i.e. first fibre orientation. Only works if $\text{randfib}==0$
modeuler	(logical) Use modified euler streamlining
rseed	(numeric) Random seed
s2tastext	(logical) Output seed-to-target counts as a text file (useful when seeding from a mesh)
opts	Additional options or way to specify things instead of command line arguments

Value

A filename of the output file

qform, character-method

Extract NIfTI 3D Image Orientation

Description

Gets q/s-forms from a character

Usage

```
## S4 method for signature 'character'
qform(object)
```

```
## S4 method for signature 'character'
sform(object)
```

Arguments

object is a nifti object

qform_code-methods

Extract Image qform_code attribute

Description

qform_code method for character types

Usage

```
## S4 method for signature 'character'
qform_code(object)
```

Arguments

object is a filename to pass to [fslval](#)

readrpi	<i>Read NIfTI file reoriented to RPI</i>
---------	--

Description

This function calls the `readnii` function after calling `rpi_orient_file` to force RPI orientation.

Usage

```
readrpi(file, ..., verbose = TRUE)
```

Arguments

<code>file</code>	file name of the NIfTI file.
<code>...</code>	Arguments to pass to <code>readnii</code>
<code>verbose</code>	print diagnostics, passed to <code>rpi_orient_file</code>

Examples

```
if (have.fsl()){
  print(fsl_version())
  in_ci <- function() {
    nzchar(Sys.getenv("CI"))
  }
  if (in_ci()) {
    destfile = tempfile(fileext = ".nii.gz")
    url = paste0("https://ndownloader.figshare.com/",
      "files/18068546")
    old_url = paste0("https://github.com/muschellij2/",
      "Neurohacking/files/3454385/113-01-MPRAGE2.nii.gz")
    dl = tryCatch(download.file(url,
      destfile = destfile))
    if (inherits(dl, "try-error") || dl != 0) {
      dl = download.file(old_url, destfile = destfile)
    }
    res = readrpi(destfile)
  }
}
```

read_xfm	<i>Read FSL Transformation</i>
----------	--------------------------------

Description

Read FSL Transformation

Usage

```
read_xfm(file)
```

Arguments

file transformation file from [flirt](#), usually ending in ‘.mat’

Value

A numeric matrix of numeric class

reverse_rpi_orient	<i>Reverse Reorientation an Image to RPI orientation</i>
--------------------	--

Description

This function uses `fslswapdim` to reorient an image

Usage

```
reverse_rpi_orient(
  file,
  convention = c("NEUROLOGICAL", "RADIOLOGICAL"),
  orientation,
  verbose = TRUE
)

reverse_rpi_orient_file(
  file,
  convention = c("NEUROLOGICAL", "RADIOLOGICAL"),
  orientation,
  verbose = TRUE
)
```

Arguments

file	Object of class <code>nifti</code> or character path
convention	Convention of original image (usually from rpi_orient)
orientation	Vector of length 3 from original image (usually from rpi_orient)
verbose	print diagnostic messages

Value

Object of class `nifti`

rpi_orient	<i>Reorient an Image to RPI orientation</i>
------------	---

Description

This function uses `fslswapdim` to reorient an image

Usage

```
rpi_orient(file, verbose = TRUE)
rpi_orient_file(file, verbose = TRUE)
is_rpi(file, verbose = FALSE)
is.rpi(file, verbose = FALSE)
```

Arguments

file	Object of class <code>nifti</code> or character path
verbose	print diagnostic messages

Value

List of 3 elements

- `img`: Reoriented image of class `nifti`
- `convention`: Convention (Neurological/Radiological) of original image
- `orientation`: Original image orientations

Note

‘orient_rpi’ and ‘orient_rpi_file’ uses ‘RNifti’ to ensure the reading orientation

Examples

```
lr_fname = system.file( "nifti", "mniLR.nii.gz", package = "oro.nifti")
img = readnii(lr_fname)

rl_fname = system.file( "nifti", "mniRL.nii.gz", package = "oro.nifti")
rl_img = readnii(rl_fname)
stopifnot(all(rl_img[nrow(rl_img):1,,] == img))

## Not run:
if (have_fsl()) {

  reor = rpi_orient(rl_fname)
```

```
rev = reverse_rpi_orient(reor$img, convention = reor$convention,
orientation = reor$orientation)
stopifnot(all(rev == rl_img))
}

## End(Not run)

reor = orient_rpi(rl_fname)
stopifnot(all(img == reor$img))

rev = reverse_orient_rpi(reor$img, convention = reor$convention,
orientation = reor$orientation)
stopifnot(all(rev == rl_img))
```

run_first_all	<i>Run FIRST All</i>
---------------	----------------------

Description

Wrapper for run_first_all from FSL for FIRST analysis segmentation of subcortical structures

Usage

```
run_first_all(
  img,
  oprefix = tempfile(),
  brain_extracted = FALSE,
  structures = NULL,
  affine = NULL,
  opts = "",
  verbose = TRUE
)
```

Arguments

- img specifies the input image (T1-weighted)
- oprefix specifies the output image basename (extensions will be added to this)
- brain_extracted specifies that the input image has been brain extracted
- structures a restricted set of structures to be segmented
- affine specifies the affine registration matrix to standard space (optional)
- opts (character) operations to be passed to run_first_all
- verbose (logical) print out command before running

Value

List of results, including result of [system](#) and some output files

run_first_all.help	<i>Run FIRST All Help</i>
--------------------	---------------------------

Description

This function calls run_first_all's help

Usage

```
run_first_all.help()
```

Value

Prints help output and returns output as character vector

Examples

```
library(fslr)

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

scl_inter-methods	<i>Extract Image scl_inter attribute</i>
-------------------	--

Description

scl_inter method for character types

Usage

```
## S4 method for signature 'character'
scl_inter(object)
```

Arguments

object is a filename to pass to [fslval](#)

scl_slope-methods	<i>Extract Image scl_slope attribute</i>
-------------------	--

Description

scl_slope method for character types

Usage

```
## S4 method for signature 'character'
scl_slope(object)
```

Arguments

object is a filename to pass to [fslval](#)

sform_code-methods	<i>Extract Image sform_code attribute</i>
--------------------	---

Description

sform_code method for character types

Usage

```
## S4 method for signature 'character'
sform_code(object)
```

Arguments

object is a filename to pass to [fslval](#)

sizeof_hdr-methods	<i>Extract Image sizeof_hdr attribute</i>
--------------------	---

Description

'sizeof_hdr' method for character types

Usage

```
## S4 method for signature 'character'
sizeof_hdr(object)
```

Arguments

object is a filename to pass to [fslval](#)

slice_code-methods	<i>Extract Image slice_code attribute</i>
--------------------	---

Description

slice_code method for character types

Usage

```
## S4 method for signature 'character'
slice_code(object)
```

Arguments

object is a filename to pass to [fslval](#)

slice_duration-methods	<i>Extract Image slice_duration attribute</i>
------------------------	---

Description

slice_duration method for character types

Usage

```
## S4 method for signature 'character'
slice_duration(object)
```

Arguments

object is a filename to pass to [fslval](#)

slice_end-methods	<i>Extract Image slice_end attribute</i>
-------------------	--

Description

slice_end method for character types

Usage

```
## S4 method for signature 'character'  
slice_end(object)
```

Arguments

object is a filename to pass to [fslval](#)

slice_start-methods	<i>Extract Image slice_start attribute</i>
---------------------	--

Description

slice_start method for character types

Usage

```
## S4 method for signature 'character'  
slice_start(object)
```

Arguments

object is a filename to pass to [fslval](#)

susan

*FSL SUSAN noise reduction***Description**

Implements Smallest Univalve Segment Assimilating Nucleus (SUSAN) noise reduction technique from FSL

Usage

```
susan(
  file,
  outfile = NULL,
  retimg = TRUE,
  reorient = FALSE,
  intern = FALSE,
  bthresh = 0.1,
  sigma = 3,
  dimg = c(3, 2),
  use_median = FALSE,
  n_usans = c(0, 1, 2),
  extra.scans = list(),
  opts = "",
  verbose = TRUE,
  ...
)
```

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 readnii .
intern	(logical) to be passed to system
bthresh	brightness threshold and should be greater than noise level and less than contrast of edges to be preserved.
sigma	spatial size (sigma i.e. half-width) of smoothing in mm.
dimg	dimensionality (2 or 3) depending on whether smoothing is to be within-plane (2) or fully 3D (3).
use_median	determines whether to use a local median filter in the cases where single-point noise is detected (0 or 1).
n_usans	determines whether the smoothing area (USAN) is to be found from secondary images (0 1 or 2).

extra.scans	List of extra scans for USAN. List of n_usans elements, where each element has 2 named objects bthresh and filename
opts	(character) operations to be passed to susan, not currently used.
verbose	(logical) print out command before running
...	additional arguments passed to fslcmd .

Value

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

References

S.M. Smith and J.M. Brady. SUSAN -a new approach to low level image processing. International Journal of Computer Vision, 23(1):45-78, May 1997.

susan.help

FSL SUSAN Help

Description

This function calls susan's help

Usage

```
susan.help()
```

Value

Prints help output and returns output as character vector

Examples

```
library(fslr)
if (have.fsl()){
  susan.help()
}
```

toffset-methods	<i>Extract Image toffset attribute</i>
-----------------	--

Description

Gets toffset from a character

Usage

```
## S4 method for signature 'character'
toffset(object)
```

Arguments

object is a filename to pass to [fslval](#)

topup	<i>topup - calling FSL topup</i>
-------	----------------------------------

Description

A tool for estimating and correcting susceptibility induced distortions

Usage

```
topup(
  infile,
  datain,
  out = NULL,
  fout = NULL,
  iout = NULL,
  logout = NULL,
  warpres = 10,
  subsamp = 1,
  fwhm = 8,
  config = NULL,
  miter = 5,
  lambda = NULL,
  sslambda = 1,
  regmod = c("bending_energy", "membrane_energy"),
  estmov = 1,
  minmet = c(0, 1),
  splineorder = c(3, 2),
  numprec = c("double", "float"),
  interp = c("spline", "linear"),
```

```

    scale = c(0, 1),
    regrid = c(0, 1),
    verbose = TRUE
)

fsl_topup(...)
```

Arguments

<code>infile</code>	name of 4D file with images
<code>datain</code>	name of text file with PE directions/times
<code>out</code>	base-name of output files (spline coefficients (Hz) and movement parameters)
<code>fout</code>	name of image file with field (Hz)
<code>iout</code>	name of 4D image file with unwarped images
<code>logout</code>	Name of log-file
<code>warpres</code>	(approximate) resolution (in mm) of warp basis for the different sub-sampling levels, default 10
<code>subsamp</code>	sub-sampling scheme, default 1
<code>fwhm</code>	FWHM (in mm) of gaussian smoothing kernel, default 8
<code>config</code>	Name of config file specifying command line arguments
<code>miter</code>	Max # of non-linear iterations, default 5
<code>lambda</code>	Weight of regularisation, default depending on <code>ssqlambda</code> and <code>regmod</code> switches. See user documentation.
<code>ssqlambda</code>	If set (=1), <code>lambda</code> is weighted by current <code>ssq</code> , default 1
<code>regmod</code>	Model for regularisation of warp-field [<code>membrane_energy</code> <code>bending_energy</code>], default <code>bending_energy</code>
<code>estmov</code>	Estimate movements if set, default 1 (true)
<code>minmet</code>	Minimisation method 0=Levenberg-Marquardt, 1=Scaled Conjugate Gradient, default 0 (LM)
<code>splineorder</code>	Order of spline, 2->Quadratic spline, 3->Cubic spline. Default=3
<code>numprec</code>	Precision for representing Hessian, double or float. Default double
<code>interp</code>	Image interpolation model, linear or spline. Default spline
<code>scale</code>	If set (=1), the images are individually scaled to a common mean, default 0 (false)
<code>regrid</code>	If set (=1), the calculations are done in a different grid, default 1 (true)
<code>verbose</code>	Print diagnostic information while running
<code>...</code>	arguments passed to <code>topup</code> if using <code>fsl_topup</code>

vox_offset-methods	<i>Extract Image vox_offset attribute</i>
--------------------	---

Description

vox_offset method for character types

Usage

```
## S4 method for signature 'character'
vox_offset(object)
```

Arguments

object is a filename to pass to [fslval](#)

xfibres	<i>Bayesian Estimation of Diffusion Parameters Obtained using Sampling Techniques with Crossing Fibers</i>
---------	--

Description

Calls xfibres from FSL to fit, also known as bedpostx

Usage

```
xfibres(
  infile,
  bvecs,
  bvals,
  mask = NULL,
  nfibres = 1,
  bet.opts = "",
  verbose = TRUE,
  njumps = NULL,
  burnin = NULL,
  burnin_noard = NULL,
  sampleevery = NULL,
  updateproposalevery = NULL,
  seed = NULL,
  noard = FALSE,
  allard = FALSE,
  nospat = FALSE,
  nonlinear = FALSE,
  cnonlinear = FALSE,
```

```

    rician = FALSE,
    f0 = FALSE,
    ardf0 = FALSE,
    opts = ""
)

```

Arguments

infile	Input filename
bvecs	b-vectors: matrix of 3 columns or filename of ASCII text file
bvals	b-values: vector of same length as number of rows of b-vectors or filename of ASCII text file
mask	Mask filename
nfibres	Maximum number of fibres to fit in each voxel (default 1)
bet.opts	Options for fslbet if mask is not supplied
verbose	print diagnostic messages
njumps	num of jumps to be made by MCMC (default is 5000)
burnin	Total num of jumps at start of MCMC to be discarded (default is 0)
burnin_noard	num of burnin jumps before the ard is imposed (default is 0)
sampleevery	num of jumps for each sample (MCMC) (default is 1)
updateproposalevery	num of jumps for each update to the proposal density std (MCMC) (default is 40)
seed	for pseudo random number generator
noard	Turn ARD off on all fibres
allard	Turn ARD on on all fibres
nospat	Initialise with tensor, not spatially
nonlinear	Initialise with nonlinear fitting
cnonlinear	Initialise with constrained nonlinear fitting
rician	Use Rician noise modelling
f0	Add to the model an unattenuated signal compartment
ardf0	Use ard on f0
opts	Additional options for xfibres. There should not be any left out in the current arguments, but opts may be a way some prefer to input options.

Value

Output from [system](#)

Index

apply_mid_sagittal_align
 (mid_sagittal_align), 135
apply_topup (applytopup), 6
applytopup, 6
aux.file, character-method
 (aux.file-methods), 7
aux.file-methods, 7
avscale (fsl_avscale), 83

bedpostx (xfibres), 155
bitpix, character-method
 (bitpix-methods), 7
bitpix-methods, 7

cal.max, character-method
 (cal.max-methods), 8
cal.max-methods, 8
cal.min, character-method
 (cal.min-methods), 8
cal.min-methods, 8
check_file, 9
checking, 9, 42, 54, 66, 71, 72, 79, 127, 134
checkout, 9, 9
concat_xfm (invert_xfm), 131

data_type, character-method
 (data_type-methods), 10
data_type-methods, 10
datatype, character-method
 (datatype-methods), 10
datatype-methods, 10
deface_image (face_removal_mask), 16
descrip, character-method
 (descrip-methods), 11
descrip-methods, 11
dim_, character-method (dim_-methods), 11
dim_-methods, 11
download.file, 12
download_fsl, 12
dtifit, 12

eddy, 13, 14
eddy_correct, 15
enforce_form, 15
epi_reg (fslepi_reg), 37

face_removal_mask, 16
fast, 17
fast.help, 18
fast_all (fast), 17
fast_nobias (fast), 17
fast_nobias_all (fast), 17
fixscaleskew_xfm (invert_xfm), 131
flirt, 19, 20, 135, 138, 144
flirt.help, 20
flirt_apply, 20
flirt_apply.help (flirt.help), 20
fnirt, 21, 24, 80
fnirt.help, 22
fnirt_with_affine, 22
fnirt_with_affine_apply, 23
fsl_abs, 75
fsl_acos, 76
fsl_add, 77
fsl_anat, 78
fsl_anat.help, 79
fsl_and (fsland), 26
fsl_applytopup (applytopup), 6
fsl_applywarp, 80
fsl_applywarp.help, 81
fsl_asin, 81
fsl_atan, 82
fsl_atlas_dir, 83
fsl_avscale, 83, 138
fsl_bet, 84
fsl_biascorrect, 85
fsl_bin, 86
fsl_bin_tab, 88, 93
fsl_binv, 87
fsl_cluster, 88
fsl_cos, 90

fsl_data_dir, 91
 fsl_deface, 92
 fsl_dice, 93
 fsl_dilate, 93
 fsl_dir (fsldir), 35
 fsl_div, 95
 fsl_edge, 96
 fsl_epi_reg (fslepi_reg), 37
 fsl_erode, 97
 fsl_exp, 98
 fsl_fast (fast), 17
 fsl_fast_nobias (fast), 17
 fsl_fill, 99
 fsl_index, 100
 fsl_log, 101
 fsl_mask, 102
 fsl_maths, 103
 fsl_merge, 104
 fsl_mul, 105
 fsl_nan, 106
 fsl_nanm, 107
 fsl_or (fslor), 50
 fsl_rand, 108
 fsl_randn, 109
 fsl_recip, 110
 fsl_rem, 111
 fsl_resample, 112
 fsl_robustfov (fslrobustfov), 57
 fsl_roi (fslroi), 58
 fsl_slicetimer (fslslicetimer), 61
 fsl_smooth, 113
 fsl_smooth_in_mask (fslsmooth_in_mask),
 63
 fsl_smoothest, 114
 fsl_split (fslsplit), 64
 fsl_sqr, 115
 fsl_sqrt, 116
 fsl_std_dir, 117
 fsl_std_file (fsl_std_dir), 117
 fsl_sub, 117
 fsl_sub2, 118
 fsl_swapdim, 119
 fsl_tan, 120
 fsl_thresh, 74, 121
 fsl_topup (topup), 153
 fsl_tsplot, 123
 fsl_version, 124
 fsl_version_gt5 (fsl_version), 124
 fsl_xor (fslxor), 74
 fslabs (fsl_abs), 75
 fslabs.help, 25
 fslacos (fsl_acos), 76
 fslacos.help, 25
 fsladd (fsl_add), 77
 fsladd.help, 26
 fsland, 26
 fslasin (fsl_asin), 81
 fslasin.help, 27
 fslatan (fsl_atan), 82
 fslatan.help, 27
 fslbet, 13, 128, 138, 156
 fslbet (fsl_bet), 84
 fslbet.help, 28
 fslbin, 26, 50, 74
 fslbin (fsl_bin), 86
 fslbin.help, 29
 fslbinv (fsl_binv), 87
 fslbinv.help, 29
 fslchfiletype, 30
 fslchfiletype.help, 31
 fslcluster (fsl_cluster), 88
 fslcmd, 15, 31, 90, 92, 115, 124, 152
 fslcog, 32
 fslcos (fsl_cos), 90
 fslcos.help, 33
 fslcpgeom, 34
 fslcpgeom.help, 35
 fsldilate (fsl_dilate), 93
 fsldir, 35
 fsldiv (fsl_div), 95
 fsldiv.help, 36
 fsledge (fsl_edge), 96
 fsledge.help, 36
 fslentropy, 37
 fslentropy.help (fslstats.help), 67
 fslepi_reg, 37
 fslerode (fsl_erode), 97
 fslerode.help, 39
 fslexp (fsl_exp), 98
 fslexp.help, 39
 fsleyes (fslview), 72
 fslfast (fast), 17
 fslfast_nobias (fast), 17
 fslfill, 41
 fslfill (fsl_fill), 99
 fslfill.help, 40

- fslfill2, [40](#)
- fslgetorient, [41](#)
- fslgetqform (fslgetorient), [41](#)
- fslgetqformcode (fslgetorient), [41](#)
- fslgetsform (fslgetorient), [41](#)
- fslgetsformcode (fslgetorient), [41](#)
- fslhd, [42](#), [43](#), [127](#)
- fslhd.help, [42](#)
- fslhd.parse, [43](#)
- fslhelp, [43](#)
- fslindex (fsl_index), [100](#)
- fslindex.help, [44](#)
- fsllog (fsl_log), [101](#)
- fsllog.help, [45](#)
- fslmask, [128](#)
- fslmask (fsl_mask), [102](#)
- fslmask.help, [45](#)
- fslmaths (fsl_maths), [103](#)
- fslmaths.help, [25–29](#), [33](#), [36](#), [39](#), [40](#), [44](#), [45](#),
[46](#), [48](#), [49](#), [52–55](#), [61](#), [62](#), [65](#), [66](#), [68](#),
[70](#)
- fslmax, [46](#)
- fslmean, [47](#)
- fslmean.help (fslstats.help), [67](#)
- fslmerge (fsl_merge), [104](#)
- fslmerge.help, [47](#)
- fslmin (fslmax), [46](#)
- fslmul, [26](#)
- fslmul (fsl_mul), [105](#)
- fslmul.help, [48](#)
- fslnan (fsl_nan), [106](#)
- fslnan.help, [48](#)
- fslnanm (fsl_nanm), [107](#)
- fslnanm.help, [49](#)
- fslor, [50](#)
- fslorient, [41](#), [51](#), [52](#)
- fslorient.help, [51](#)
- fslorienter, [52](#)
- fslrand (fsl_rand), [108](#)
- fslrand.help, [52](#)
- fslrandn (fsl_randn), [109](#)
- fslrandn.help, [53](#)
- fslrange, [46](#), [53](#)
- fslrange.help (fslstats.help), [67](#)
- fslrecip (fsl_recip), [110](#)
- fslrecip.help, [54](#)
- fslrem (fsl_rem), [111](#)
- fslrem.help, [55](#)
- fslreorient2std, [55](#), [56](#)
- fslreorient2std.help, [56](#)
- fslreorient2std_mat (fslreorient2std),
[55](#)
- fslrobustfov, [57](#)
- fslrobustfov.help, [58](#)
- fslroi, [58](#)
- fslroi_time (fslroi), [58](#)
- fslsd, [59](#)
- fslsd.help (fslstats.help), [67](#)
- fslsin, [60](#)
- fslsin.help, [61](#)
- fslslicetimer, [61](#)
- fslsmooth, [63](#)
- fslsmooth (fsl_smooth), [113](#)
- fslsmooth.help, [62](#)
- fslsmooth_in_mask, [63](#)
- fslsplit, [64](#)
- fslsplit.help, [65](#)
- fslsqr (fsl_sqr), [115](#)
- fslsqr.help, [65](#)
- fslsqrt (fsl_sqrt), [116](#)
- fslsqrt.help, [66](#)
- fslstats, [37](#), [47](#), [60](#), [66](#), [69](#), [74](#)
- fslstats.help, [67](#)
- fslsub (fsl_sub), [117](#)
- fslsub.help, [67](#)
- fslsub2 (fsl_sub2), [118](#)
- fslsub2.help, [68](#)
- fslsum, [69](#), [73](#)
- fslswapdim (fsl_swapdim), [119](#)
- fslswapdim.help, [69](#)
- fsltan (fsl_tan), [120](#)
- fsltan.help, [70](#)
- fslthresh (fsl_thresh), [121](#)
- fslthresh.help, [70](#)
- fslval, [7](#), [8](#), [10](#), [11](#), [71](#), [129](#), [130](#), [132](#), [139](#),
[142](#), [147–150](#), [153](#), [155](#)
- fslval.help, [71](#)
- fslversion (fsl_version), [124](#)
- fslview, [72](#)
- fslview.help, [73](#)
- fslvol, [73](#)
- fslvolume, [74](#)
- fslxor, [74](#)
- get.fsl, [125](#), [128](#)
- get.fsloutput, [126](#)
- get.imgext, [126](#)

- get_fsl (get.fsl), 125
- get_mrdefacer_mask (mrdefacer), 137
- get_quickshear_mask, 127
- getForms, 9, 15, 127
- have.fsl, 128
- have_fsl (have.fsl), 128
- intent_code, character-method
 - (intent_code-methods), 129
- intent_code-methods, 129
- intent_name, character-method
 - (intent_name-methods), 129
- intent_name-methods, 129
- intent_p1, character-method
 - (intent_p1-methods), 130
- intent_p1-methods, 130
- intent_p2, character-method
 - (intent_p2-methods), 130
- intent_p2-methods, 130
- intent_p3, character-method
 - (intent_p3-methods), 130
- intent_p3-methods, 130
- invert_xfm, 131
- is.rpi (rpi_orient), 145
- is_rpi (rpi_orient), 145
- magic, character-method (magic-methods), 132
- magic-methods, 132
- mcflirt, 132
- mcflirt.help, 133
- melodic, 133
- melodic.help, 134
- mid_sagittal_align, 135
- mni_face_fname, 16
- mni_face_fname (mni_fname), 136
- mni_fname, 16, 136, 137
- mni_img, 137
- mrdefacer, 137
- nifti, 137
- parse_avscale, 83, 138
- pixdim, character-method
 - (pixdim-methods), 139
- pixdim-methods, 139
- probtrackx, 139
- qform, character
 - (qform, character-method), 142
- qform, character-method, 142
- qform_code, character-method
 - (qform_code-methods), 142
- qform_code-methods, 142
- quickshear_deface_image
 - (get_quickshear_mask), 127
- read_cluster_table (fsl_cluster), 88
- read_xfm, 143
- readnii, 18, 19, 21–24, 30, 32, 34, 38, 41, 50, 51, 56, 57, 59, 60, 62, 64, 75–78, 80–82, 84–87, 89, 91, 92, 94–113, 115, 116, 118–122, 132, 143, 151
- readrpi, 143
- reverse_rpi_orient, 144
- reverse_rpi_orient_file
 - (reverse_rpi_orient), 144
- rpi_orient, 144, 145
- rpi_orient_file, 135, 143
- rpi_orient_file (rpi_orient), 145
- run_first_all, 146
- run_first_all.help, 147
- scl_inter, character-method
 - (scl_inter-methods), 147
- scl_inter-methods, 147
- scl_slope, character-method
 - (scl_slope-methods), 148
- scl_slope-methods, 148
- sform, character
 - (qform, character-method), 142
- sform, character-method
 - (qform, character-method), 142
- sform_code, character-method
 - (sform_code-methods), 148
- sform_code-methods, 148
- sizeof_hdr, character-method
 - (sizeof_hdr-methods), 148
- sizeof_hdr-methods, 148
- slice_code, character-method
 - (slice_code-methods), 149
- slice_code-methods, 149
- slice_duration, character-method
 - (slice_duration-methods), 149
- slice_duration-methods, 149
- slice_end, character-method
 - (slice_end-methods), 150

slice_end-methods, [150](#)
slice_start,character-method
 (slice_start-methods), [150](#)
slice_start-methods, [150](#)
susan, [151](#)
susan.help, [152](#)
system, [18](#), [19](#), [21–24](#), [30](#), [32](#), [34](#), [38](#), [41](#), [50](#),
 [51](#), [56](#), [57](#), [59](#), [60](#), [62](#), [72](#), [75–82](#),
 [84–87](#), [91](#), [92](#), [94–99](#), [101](#), [102](#),
 [104–111](#), [113](#), [115](#), [116](#), [118–122](#),
 [132](#), [134](#), [146](#), [151](#), [156](#)

toffset,character-method
 (toffset-methods), [153](#)
toffset-methods, [153](#)
topup, [153](#)

vox_offset,character-method
 (vox_offset-methods), [155](#)
vox_offset-methods, [155](#)
voxdim, [73](#)

xfibres, [155](#)