Package 'neurobase'

October 23, 2022

Type Package

Title 'Neuroconductor' Base Package with Helper Functions for 'nifti' Objects

Version 1.32.3

Maintainer John Muschelli <muschellij2@gmail.com>

Description Base package for 'Neuroconductor', which includes many helper functions that interact with objects of class 'nifti', implemented by package 'oro.nifti', for reading/writing and also other manipulation functions.

Imports methods, abind, matrixStats, R.utils, graphics, grDevices, stats, RNifti

Depends oro.nifti (>= 0.11.3), R (>= 3.2.0)

License GPL-2

Suggests testthat, ggplot2, knitr, rmarkdown, dplyr, reshape2, httr, covr, brainR

VignetteBuilder knitr

BugReports https://github.com/muschellij2/neurobase/issues

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author John Muschelli [aut, cre]

Repository CRAN

Date/Publication 2022-10-23 16:32:36 UTC

R topics documented:

`applyEmptyImageDimensions-methods`
Apply Subsetting from Empty Image Dimensions

Description

Simple wrapper for subsetting an image with indices, dropping empty dimensions.

Usage

```
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

## S4 method for signature 'nifti'
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

## S4 method for signature 'character'
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

## S4 method for signature 'factor'
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

## S4 method for signature 'list'
```

```
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

## S4 method for signature 'array'
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

## S4 method for signature 'anlz'
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

## S4 method for signature 'ANY'
applyEmptyImageDimensions(img, inds, reorient = FALSE, ...)

apply_empty_dim(img, ...)
```

Arguments

img	image, nifti object, or array
inds	indices of subset from getEmptyImageDimensions or dropEmptyImageDimensions.
reorient	Should image be reoriented if a filename?
...	not used

Value

Object of class nifti or array if nifti is not supplied

Note

apply_empty_dim is a shorthand for applyEmptyImageDimensions with all the same arguments.

See Also

getEmptyImageDimensions, dropEmptyImageDimensions

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
inds = getEmptyImageDimensions(nim)
inds_arr = getEmptyImageDimensions(arr)
testthat::expect_equal(inds, inds_arr)

out = applyEmptyImageDimensions(nim, inds = inds)
out_arr = applyEmptyImageDimensions(arr, inds = inds)
testthat::expect_equal(out_arr, array(out, dim = dim(out)))
out = apply_empty_dim(nim, inds = inds)

set.seed(5)
dims = rep(10, 3)
```

```
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
inds = getEmptyImageDimensions(nim)
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
apply_empty_dim(nim, inds = inds)
func = function(...) applyEmptyImageDimensions(..., inds = inds)
func(arr)
func(nim)
func(rnifti)
func(timg)
func(limg)
```

`boxplot.nifti` *Boxplot of Values in an Image*

Description

Computes the boxplot of values of an image with the option for a mask.

Usage

```
## S3 method for class 'nifti'
boxplot(x, ..., mask)

## S3 method for class 'anlz'
boxplot(x, ..., mask)
```

Arguments

`x`	Object of class `nifti`
`...`	Arguments passed to `boxplot.default`
`mask`	object to subset the image. If missing, then all values of the image are plotted.

Value

Output of `boxplot`

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
mask = img > 0
boxplot(img)
boxplot(img, mask = mask)
boxplot(as.anlz(img))
```

breaker *Find Breaks for nifti Image Plotting*

Description

Helper function for plotting - returns breaks for `image` plot function for object of class `nifti`

Usage

```
breaker(x, zlim, col, breaks = NULL)
```

Arguments

`x`	Object of class `nifti`
`zlim`	A user-specified `zlim`. If `NULL`, will calculate how `ortho2` would calculate `zlim`
`col`	colors to be plotted. Only used for `length(col)`, so can be a vector of length cols to be plotted
`breaks`	if `!is.null(breaks)`, then will calculate breaks. Otherwise will return this breaks vector

Value

Vector of length 2

If `breaks = NULL`, then vector of `length(col) + 1`, otherwise returns `breaks`

checkimg-methods *Force object to filename*

Description

Ensures the output to be a character filename (or vector) from an input image or `nifti`.

Usage

```
checkimg(file, allow_array = FALSE, ...)

## S4 method for signature 'nifti'
checkimg(file, allow_array = FALSE, ...)

## S4 method for signature 'ANY'
checkimg(file, allow_array = FALSE, ...)

## S4 method for signature 'character'
checkimg(file, allow_array = FALSE, ...)

## S4 method for signature 'list'
checkimg(file, allow_array = FALSE, ...)
```

Arguments

file	character or nifti object
allow_array	allow arrays to be passed in
...	options passed to tempimg

Value

character filename of image or temporary nii, with .nii extension

Author(s)

John Muschelli <muschellij2@gmail.com>

---

checknii-methods *Force object to filename with .nii extension*

Description

Ensures the output to be a character filename (or vector) from an input image or nifti, but not gzipped and has .nii extension

Usage

```
checknii(file, ...)

## S4 method for signature 'nifti'
checknii(file, ...)

## S4 method for signature 'factor'
checknii(file, ...)

## S4 method for signature 'character'
checknii(file, ...)

## S4 method for signature 'list'
checknii(file, ...)

## S4 method for signature 'ANY'
checknii(file, ...)

ensure_nii(file, ...)
```

Arguments

file	character or nifti object
...	options passed to checkimg

Value

character filename of image or temporary nii, with .nii extension

Author(s)

John Muschelli <muschellij2@gmail.com>

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
func = checknii
func(nim)
func(rnifti)
func(timg)
func(limg)
```

---

checkniigz-methods *Force object to filename with .nii.gz extension*

Description

Ensures the output to be a character filename (or vector) from an input image or `nifti`, to be gzipped and has `.nii.gz` extension

Usage

```
checkniigz(file, ...)

## S4 method for signature 'nifti'
checkniigz(file, ...)

## S4 method for signature 'ANY'
checkniigz(file, ...)

## S4 method for signature 'factor'
checkniigz(file, ...)

## S4 method for signature 'character'
checkniigz(file, ...)

## S4 method for signature 'list'
```

```
checkniigz(file, ...)

ensure_nii_gz(file, ...)
```

Arguments

file	character or nifti object
...	options passed to checkimg

Value

Character filename of image or temporary nii, with .nii.gz extension

Author(s)

John Muschelli <muschellij2@gmail.com>

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
func = checkniigz
func(nim)
func(rnifti)
func(timg)
func(limg)
```

---

check_mask *Check Mask is Binary*

Description

Determine if only values in a mask are 0/1

Usage

```
check_mask(mask, allow.NA = FALSE, allow.array = TRUE)
```

Arguments

mask	Object of class nifti
allow.NA	allow NAs in the mask
allow.array	if class(mask) is "array", is this OK?

Value

Logical indicating if object is binary mask with only 0, 1, and NA if applicable

Examples

```
arr = array(rbinom(1000, size = 1, prob = 0.2), dim = c(10,10,10))
nim = oro.nifti::nifti(arr)
check_mask(nim)
```

---

check_mask_fail *Check Mask is Binary, Fail otherwise*

Description

Determine if only values in a mask are 0/1. Will error otherwise.

Usage

```
check_mask_fail(...)
```

Arguments

...	arguments to pass to check_mask

Value

Either will error if conditions not met or an invisible NULL

Examples

```
arr = array(rbinom(1000, size = 1, prob = 0.2), dim = c(10,10,10))
nim = oro.nifti::nifti(arr)
check_mask_fail(nim)
```

---

check_nifti-methods *Check if nifti image or read in a nifti image*

Description

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

Usage

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

## S4 method for signature 'nifti'
check_nifti(
  x,
  reorient = FALSE,
  allow.array = FALSE,
  fast = FALSE,
  need_header = TRUE,
  ...
)

## S4 method for signature 'character'
check_nifti(
  x,
  reorient = FALSE,
  allow.array = FALSE,
  fast = FALSE,
  need_header = TRUE,
  ...
)

## S4 method for signature 'factor'
check_nifti(
  x,
  reorient = FALSE,
  allow.array = FALSE,
  fast = FALSE,
  need_header = TRUE,
  ...
)

## S4 method for signature 'list'
check_nifti(
  x,
  reorient = FALSE,
  allow.array = FALSE,
  fast = FALSE,
  need_header = TRUE,
  ...
```

```
)

## S4 method for signature 'array'
check_nifti(
  x,
  reorient = FALSE,
  allow.array = FALSE,
  fast = FALSE,
  need_header = FALSE,
  ...
)

## S4 method for signature 'anlz'
check_nifti(
  x,
  reorient = FALSE,
  allow.array = FALSE,
  fast = FALSE,
  need_header = TRUE,
  ...
)

## S4 method for signature 'ANY'
check_nifti(
  x,
  reorient = FALSE,
  allow.array = FALSE,
  fast = FALSE,
  need_header = TRUE,
  ...
)
```

Arguments

x	character path of image or an object of class nifti, or array
reorient	(logical) passed to readnii if the image is to be re-oriented
allow.array	(logical) Are array types allowed (TRUE) or should there be an error if the object is not character or class nifti.
fast	if TRUE, then fast_readnii will be used versus readnii if the files need to be read in.
need_header	if TRUE, then an image type with header information will be returned. If not, then an array is fine. Used really only in conjunction with allow.array
...	additional arguments to pass to readnii if relevant

Value

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

Author(s)

John Muschelli <muschellij2@gmail.com>

See Also

readnii

Examples

```
x = nifti()
check_nifti(x)
set.seed(5)
dims = rep(10, 4)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
check_nifti(nim)
check_nifti(as.anlz(nim))
testthat::expect_error(check_nifti(arr, allow.array = FALSE))
tfile = tempimg(nim)
check_nifti(c(tfile, tfile))
check_nifti(list(tfile, tfile))
check_nifti(factor(c(tfile, tfile)))
check_nifti(RNifti::readNifti(tfile))
```

---

check_nifti_header-methods

Check if nifti image or read in a nifti header

Description

Simple check to see if input is character or of class nifti and read in the header

Usage

```
check_nifti_header(x)

## S4 method for signature 'nifti'
check_nifti_header(x)

## S4 method for signature 'character'
check_nifti_header(x)

## S4 method for signature 'factor'
check_nifti_header(x)

## S4 method for signature 'list'
check_nifti_header(x)
```

```
## S4 method for signature 'array'
check_nifti_header(x)

## S4 method for signature 'anlz'
check_nifti_header(x)

## S4 method for signature 'ANY'
check_nifti_header(x)
```

Arguments

x	character path of image or an object of class nifti, or array

Value

nifti object or character

Author(s)

John Muschelli <muschellij2@gmail.com>

Examples

```
set.seed(5)
dims = rep(10, 4)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
check_nifti_header(nim)
check_nifti_header(as.anlz(nim))
testthat::expect_error(check_nifti_header(arr))
tfile = tempimg(nim)
check_nifti_header(tfile)
check_nifti_header(RNifti::readNifti(tfile))
check_nifti_header(c(tfile, tfile))
check_nifti_header(list(tfile, tfile))
check_nifti_header(factor(tfile))
```

---

check_outfile *Check output filename*

Description

This function checks if an output filename is not NULL in conjunction whether the user would like to return an image

Usage

```
check_outfile(outfile, retimg, fileext = ".nii.gz")
```

Arguments

outfile	output filename or NULL
retimg	Should an image be returned
fileext	a non-empty character vector giving the file extension

Value

Filename of output file or a temporary filename

---

cog *Image Center of Gravity*

Description

Find Center of Gravity of Image, after thresholding

Usage

```
cog(img, thresh = 0, ceil = FALSE, warn = TRUE)
```

Arguments

img	Object of class nifti
thresh	threshold for image, will find img > 0
ceil	Run ceiling to force integers (usu for plotting)
warn	Produce a warning if the image is empty after thresholding

Value

Vector of length 3

Examples

```
dims = rep(20, 3)
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))
cog(img)
```

colorbar *Add a colorbar to an ortho2 object*

Description

Adds a series of colors mapped to a value

Usage

```
colorbar(breaks, col, text.col = "white", labels = TRUE, maxleft = 0.95)
```

Arguments

breaks	a set of finite numeric breakpoints for the colours (see image
col	a list of colors (see image
text.col	axis and text label color
labels	labels for tick marks - see axis
maxleft	Extent the left hand for colorbar

Value

A plot

Note

Much of this was taken from `vertical.image.legend` from the `aqfig` package

copyNIfTIHeader *Copy NIfTI Header to an array*

Description

Copies slots of a `nifti` object to an array. This is useful if you're subsetting 4D data and getting an array out

Usage

```
copyNIfTIHeader(
  img,
  arr,
  drop_slots = c(".Data", "dim_"),
  drop = TRUE,
  onlylast = TRUE,
  warn = TRUE,
  ...
)
```

Arguments

img	object of class nifti to copy header
arr	array to copy header information
drop_slots	Slots not to copy over from header
drop	Should dropImageDimension be called before returning?
onlylast	if drop = TRUE, passed to dropImageDimension, if only the last dimensions should be dropped
warn	if drop = TRUE, passed to dropImageDimension, for warning print out
...	arguments to pass to nifti

Value

Object of class nifti the size of arr

Examples

```
img = nifti(img = array(rnorm(10^4), dim=rep(10, 4)), dim=rep(10, 4), datatype = 16)
sub = img[,,,1:3]
copyNIfTIHeader(img, sub)
sub = img[,,,1, drop=FALSE]
copyNIfTIHeader(img, sub)
copyNIfTIHeader(img, sub, drop = FALSE)
```

---

cut.nifti *Perform Cut on an image*

Description

Cuts a numeric image into an integer factor, with the option of a mask.

Usage

```
## S3 method for class 'nifti'
cut(x, breaks, ..., mask)

## S3 method for class 'anlz'
cut(x, ..., mask)
```

Arguments

x	Object of class nifti
breaks	either a numeric vector of two or more unique cut points or a single number (greater than or equal to 2) giving the number of intervals into which x is to be cut. Passed to cut)
...	Arguments passed to cut
mask	object to subset the image. If missing, then all values of the image are used

Value

Object of class `nifti` with an `attribute` of levels

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
mask = img > 0
cut(img, mask = mask, breaks = 4)
```

`datatype` *Change Data type for img*

Description

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

Usage

```
datatyper(
  img,
  type_string = NULL,
  datatype = NULL,
  bitpix = NULL,
  trybyte = TRUE,
  warn = TRUE
)
```

Arguments

`img`	nifti object (or character of filename)
`type_string`	(NULL) character of datatype and bitpix. Supercedes both datatype and bitpix. If specified `convert.datatype[[type_string]]` and `convert.bitpix[[type_string]]` will be used.
`datatype`	(NULL) character of datatype see `convert.datatype`
`bitpix`	(NULL) character of bitpix see `convert.bitpix`
`trybyte`	(logical) Should you try to make a byte (UINT8) if image in c(0, 1)?
`warn`	Should a warning be issued if defaulting to FLOAT32?

Value

object of type nifti

Examples

```
img = nifti(array(rnorm(10^3, sd = 1000), dim = rep(10, 3)))
rimg = round(img)
newnii(datatyper(rimg))
rimg = datatyper(rimg, type_string= "FLOAT32")
```

density.nifti *Density of Values in an Image*

Description

Computes the density of values of an image with the option for a mask.

Usage

```
## S3 method for class 'nifti'
density(x, ..., mask)

## S3 method for class 'anlz'
density(x, ..., mask)
```

Arguments

x	Object of class nifti
...	Arguments passed to density.default
mask	object to subset the image. If missing, then all values of the image are plotted.

Value

Output of density

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
mask = img > 0
density(img, mask = mask)
density(img)

density(as.anlz(img), mask = mask)
density(as.anlz(img))
```

dicer *Calculate Dice from a Table*

Description

Simple wrapper to calculate the Dice Coefficient/Similarity Index from a table

Usage

```
dicer(tab, verbose = TRUE)
```

Arguments

tab	table or matrix that is 2 by 2
verbose	should the Dice be printed before returned?

Value

Numeric scalar (one number)

Examples

```
tab = matrix(c(1000, 20, 20, 400), ncol = 2)
dicer(tab)
```

---

double_ortho *Double Orthographic Display*

Description

Copy of oro.nifti's orthographic function with some tweaks such as adding L/R designations for left and right

Usage

```
double_ortho(
  x,
  y = NULL,
  col.y = gray(0:64/64),
  NA.x = TRUE,
  mfrow = c(2, 4),
  add = FALSE,
  ...
)
```

Arguments

x	is an object of class nifti or similar.
y	is an object of class nifti or similar to be set aside x.
col.y	is grayscale (by default).
NA.x	Set any values of 0 in x to NA
mfrow	(numeric) layout of the 3 slices
add	Should the y-plot be added or its own plot? Used in double_ortho
...	other arguments to ortho2

See Also

orthographic

dropEmptyImageDimensions
Drop Empty Image Dimensions

Description

Drops dimensions of an image that has all irrelevant values

Usage

```
dropEmptyImageDimensions(
  img,
  value = 0,
  threshold = 0,
  other.imgs = NULL,
  keep_ind = FALSE,
  reorient = FALSE
)

drop_empty_dim(
  img,
  value = 0,
  threshold = 0,
  other.imgs = NULL,
  keep_ind = FALSE,
  reorient = FALSE
)
```

Arguments

img	nifti object
value	Value to check against. If zero, then dropEmptyImageDimensions will drop any dimension that has fewer than threshold zeroes. May be a vector of values, matched with match
threshold	Drop dimension if fewer than threshold voxels are in the slice
other.imgs	List of other nifti objects or filenames to apply the same dropping as img.
keep_ind	keep indices in output. Will return list, even if other.imgs not specified
reorient	Should image be reoriented if a filename?

Value

List of output image indices, and other images if other.imgs not specified or keep_ind = TRUE. Otherwise object of class nifti

Note

drop_empty_dim is a shorthand for dropEmptyImageDimensions with all the same arguments. Also, NA are set to zero.

See Also

getEmptyImageDimensions

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)

dnim = dropEmptyImageDimensions(nim, keep_ind = TRUE)
new_nim = dnim$outimg
names(dnim)
dnim = dropEmptyImageDimensions(nim, keep_ind = TRUE, other.imgs = nim)
dims = rep(10, 4)
arr = array(rnorm(prod(dims)), dim = dims)
nim = oro.nifti::nifti(arr)

testthat::expect_error(
{dnim = dropEmptyImageDimensions(nim, keep_ind = TRUE)}
)
```

---

emptyImageDimensionsMask

Make Mask from Empty Image Dimensions

Description

Make a mask of an image that has all irrelevant values

Usage

```
emptyImageDimensionsMask(img, ..., reorient = FALSE)

empty_dim_mask(img, ..., reorient = FALSE)
```

Arguments

img	nifti object
...	Arguments to be passed to getEmptyImageDimensions.
reorient	Should image be reoriented if a filename?

Value

Object of class `nifti`, with binary values

Note

`empty_dim_mask` is a shorthand for `emptyImageDimensionsMask` with all the same arguments.

See Also

`getEmptyImageDimensions`

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
out = emptyImageDimensionsMask(nim)
out_arr = emptyImageDimensionsMask(arr)
testthat::expect_equal(out_arr, array(out, dim = dim(out)))
out_arr = empty_dim_mask(arr)
```

---

ensure_array *Ensure an array output*

Description

Forces an array output for manipulation and overall conversion

Usage

```
ensure_array(img)
```

Arguments

`img`	Image object (`nifti` or `niftiImage`)

Value

Array of same dimensions as image object

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
func = ensure_array
func(arr)
func(nim)
func(rnifti)
func(timg)
func(limg[[1]])
```

---

fast_dice_tab *Fast Dice Tabulation*

Description

Fast Dice Tabulation

Usage

```
fast_dice_tab(x, y)

fast_dice(x, y, verbose = FALSE)
```

Arguments

x	A nifti image, filename, or niftiImage
y	A nifti image, filename, or niftiImage
verbose	A logical indicating output

Value

A table object

Examples

```
library(oro.nifti)
set.seed(20161007)
dims = rep(10, 3)
arr = array(rnorm(10*10*10), dim = dims)
nim = oro.nifti::nifti(arr) > -1
fast_dice_tab(nim, nim)
fast_dice(nim, nim) == 1
```

fast_readnii *Reading NIfTI images through RNifti*

Description

This function calls the readNifti function from the RNifti package, and then converts the image to a nifti object

Usage

```
fast_readnii(fname, dtype = TRUE, drop_dim = TRUE)
```

Arguments

fname	file name of the NIfTI file.
dtype	Should datatyper be run after reading?
drop_dim	Should drop_img_dim be run after reading?

Value

A nifti object

Examples

```
set.seed(5)
dims = rep(10, 4)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
tfile = tempfile(fileext = ".nii.gz")
write_nifti(nim, tfile)
rimg = fast_readnii(tfile)
```

file_imgext *Get Image file extension*

Description

Get the image file extension, either .nii, .hdr, .nii.gz, or .hdr.gz

Usage

```
file_imgext(file, withdot = TRUE)
```

Arguments

file	Vector of character filenames
withdot	Should the extension begin with "."?

Value

Vector of extensions. If `withdot = FALSE`, then will return `nii` instead of `.nii`

`finite_img-methods` *Finite Image*

Description

Simple wrapper for setting non-finite values to zero

Usage

```
finite_img(img, replace = 0)

## S4 method for signature 'nifti'
finite_img(img, replace = 0)

## S4 method for signature 'array'
finite_img(img, replace = 0)

## S4 method for signature 'ANY'
finite_img(img, replace = 0)

## S4 method for signature 'character'
finite_img(img, replace = 0)

## S4 method for signature 'list'
finite_img(img, replace = 0)
```

Arguments

`img`	character path of image or an object of class `nifti`, or list of images
`replace`	Value to replace non-finite values to

Value

nifti object

Author(s)

John Muschelli <muschellij2@gmail.com>

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
arr[c(5, 6, 7, 8)] = c(NA, NaN, Inf, -Inf)
nim = nifti(arr)
finite_img(nim)
finite_img(arr)
tfile = tempimg(nim)
checkimg(c(tfile, tfile))
checkimg(list(tfile, tfile))
finite_img(list(tfile, tfile))
finite_img(c(tfile, tfile))
img = RNifti::readNifti(tfile)
checkimg(img)
img[c(5, 6, 7, 8)] = c(NA, NaN, Inf, -Inf)
stopifnot(!any(c(is.na(c(finite_img(img))))))
```

---

flip_img *Flip NifTI Image*

Description

This image will flip x, y, or z direction

Usage

```
flip_img(img, x = FALSE, y = FALSE, z = FALSE, ...)
```

Arguments

img	nifti object or character filename
x	(logical) Flip x direction
y	(logical) Flip y direction
z	(logical) Flip z direction
...	Arguments passed to check_nifti

Value

Object of class nifti

Examples

```
img = random_nifti(rep(15, 3))
flipped = flip_img(img, x = TRUE, y = TRUE, z = TRUE)
img = random_nifti(rep(15, 2))
flipped = flip_img(img, x = TRUE)
testthat::expect_error(flip_img(img, z= TRUE))
```

getEmptyImageDimensions *Get Empty Image Dimensions*

Description

Creates a list of indices of an image that has all irrelevant values

Usage

```
getEmptyImageDimensions(img, value = 0, threshold = 0, reorient = FALSE, ...)

get_empty_dim(img, value = 0, threshold = 0, reorient = FALSE, ...)
```

Arguments

img	nifti object or array
value	Value to check against. If zero, then getEmptyImageDimensions will include any dimension that has fewer than threshold zeroes. May be a vector of values, matched with match
threshold	Include dimension if fewer than threshold voxels are in the slice
reorient	Should image be reoriented if a filename?
...	additional arguments to pass to check_nifti

Value

List of length 3 of indices.

Note

get_empty_dim is a shorthand for getEmptyImageDimensions with all the same arguments. Also, NA are set to zero.

Examples

```
arr = array(rbinom(1000, size = 1, prob = 0.2), dim = c(10,10,10))
arr[,,1] = 0
arr[2:3,,] = 0
getEmptyImageDimensions(arr)
```

hist.nifti *Histogram of Values in an Image*

Description

Computes and displays a histogram of the values of an image with the option for a mask.

Usage

```
## S3 method for class 'nifti'
hist(x, ..., mask)

## S3 method for class 'anlz'
hist(x, ..., mask)
```

Arguments

x	Object of class nifti
...	Arguments passed to hist.default
mask	object to subset the image. If missing, then all values of the image are plotted.

Value

Output of hist

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
mask = img > 0
hist(img, mask = mask)
```

images2matrix *Transform set of images to matrix*

Description

Creates a matrix, where the voxels are on the rows and images are on the columns

Usage

```
images2matrix(imgs, mask = NULL)
```

Arguments

imgs	Vector of files or list of images (niftiImage, array, or nifti)
mask	Binary image to subset the voxels

Value

Matrix of V by p, where V is the product of the dimensions of one image or the number of voxels in the mask, and p is the number of images

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
imgs = list(nim, arr)
mask = nim > 2
mat1 = images2matrix(imgs)
mat2 = images2matrix(list(nim, nim))
if (packageVersion("oro.nifti") >= package_version("0.10.2")) {
testthat::expect_equal(mat1, mat2)
} else {
testthat::expect_error(testthat::expect_equal(mat1, mat2))
}
mat1 = images2matrix(imgs, mask = mask)
mat2 = images2matrix(list(nim, nim), mask)
```

---

img_colour_df *Convert Image to Data.frame with Colors*

Description

Takes in an image and a color scheme, converts that image into a `data.frame` with the data and a color mapping.

Usage

```
img_colour_df(img, zlim = NULL, breaks = NULL, col = gray(0:64/64))

img_color_df(...)
```

Arguments

`img`	an object to be coerced to `nifti` using `check_nifti`
`zlim`	Limits for the domain of the intensities
`breaks`	Breaks for the intensities to map to colors
`col`	Colors to map intensities
`...`	not used

Value

A `data.frame` with the first columns being the x,y,z (maybe t) coordinates (named `dim` and the dimension number), a `value` column that contains the intensity information, and a `colour` column representing the color that voxel maps to

Note

`img_color_df` is a duplicate of `img_colour_df`

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
df = img_colour_df(img)
df = img_color_df(img)
```

---

img_indices *Retrieve Image Indices*

Description

Extract image xyz indices (in voxels or millimeters), with the option to append the values

Usage

```
img_indices(img, mask = NULL, add_values = FALSE, units = c("index", "mm"))
```

Arguments

`img`	Object of class `nifti`
`mask`	Mask to be applied for indices the index
`add_values`	Should the value be column-bound to the matrix
`units`	Should the indices be in xyz-coordinates or millimeters.

Value

Matrix of 3 columns if `add_values = FALSE` or 4 columns, otherwise.

Examples

```
set.seed(5)
dims = rep(10, 4)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
ind = img_indices(nim)
ind2 = img_indices(nim, mask = nim > 2)
# 3d example
set.seed(5)
dims = rep(10, 3)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
ind = img_indices(nim)
ind2 = img_indices(nim, mask = nim > 2)
testthat::expect_equal(colnames(ind2), c("x", "y", "z"))
ind2 = img_indices(nim, mask = nim > 2, add_values = TRUE)
testthat::expect_equal(colnames(ind2), c("x", "y", "z", "value"))
```

img_list_to_ts *Image List to Time Series*

Description

Turns a a list of 3D images into a 4D time series image

Usage

```
img_list_to_ts(imgs, copy_nifti = TRUE, warn = TRUE)
```

Arguments

imgs	object of class list, each with 3 dimensions,
copy_nifti	Should a nifti object be returned (TRUE) or a simply array (FALSE). Should only be used for slight speed up when array is adequate
warn	Should a warning be printed if object is not class nifti

Value

Object of class nifti

Note

If the object is not of class list, then the object is returned

img_ts_to_df *Image Time Series to Data.frame*

Description

Turns a 4D time series image to a Data.frame

Usage

```
img_ts_to_df(imgs, warn = FALSE)
```

Arguments

imgs	object of class nifti with 4 dimensions, aka a 4D time series
warn	Should a warning be printed if object is not class nifti (e.g. a list instead)

Value

Matrix of values

`img_ts_to_list` *Image Time Series to list*

Description

Turns a 4D time series image to a list of 3D images

Usage

```
img_ts_to_list(imgs, copy_nifti = TRUE, warn = TRUE)
```

Arguments

`imgs`	object of class `nifti` with 4 dimensions, aka a 4D time series
`copy_nifti`	Should `nifti` objects be returned (`TRUE`) or simply arrays (`FALSE`). Should only be used for slight speed up when array is adequate
`warn`	Should a warning be printed if object is not class `nifti`

Value

List of images

Note

If the object is not of class `nifti` or have 4 dimensions, then the object is returned

Examples

```
set.seed(5)
dims = rep(10, 4)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
simg = img_ts_to_list(nim)
simg_arr = img_ts_to_list(arr)
back = img_list_to_ts(simg_arr)
slist = lapply(simg, function(x) array(x, dim(x)))
testthat::expect_equal(slist, simg_arr)
simg_arr = img_ts_to_matrix(arr)
simg_arr = img_ts_to_df(arr)
```

img_ts_to_matrix *Image Time Series to Matrix*

Description

Turns a 4D time series image to a Matrix

Usage

```
img_ts_to_matrix(imgs, warn = FALSE)
```

Arguments

imgs	object of class nifti with 4 dimensions, aka a 4D time series
warn	Should a warning be printed if object is not class nifti (e.g. a list instead)

Value

Matrix of values

maskEmptyImageDimensions-methods *Apply Masking from Empty Image Dimensions*

Description

Simple wrapper for replacing indices with a value

Usage

```
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

## S4 method for signature 'nifti'
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

## S4 method for signature 'character'
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

## S4 method for signature 'factor'
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

## S4 method for signature 'list'
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

## S4 method for signature 'array'
```

```
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

## S4 method for signature 'anlz'
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

## S4 method for signature 'ANY'
maskEmptyImageDimensions(img, inds, reorient = FALSE, mask.value = 0, ...)

mask_empty_dim(img, ...)
```

Arguments

img	image, nifti object, or array
inds	indices of subset from getEmptyImageDimensions or dropEmptyImageDimensions.
reorient	Should image be reoriented if a filename?
mask.value	Value to replace voxels outside the mask.
...	not used

Value

Object of class nifti or array if nifti is not supplied

Note

mask_empty_dim is a shorthand for maskEmptyImageDimensions with all the same arguments.

See Also

getEmptyImageDimensions, dropEmptyImageDimensions

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
inds = getEmptyImageDimensions(nim)
inds_arr = getEmptyImageDimensions(arr)
res = maskEmptyImageDimensions(nim, inds = inds, mask.value = NA)
res2 = maskEmptyImageDimensions(arr, inds = inds_arr, mask.value = NA)
testthat::expect_equal(array(res, dim = dim(res)),
array(res2, dim = dim(res2)))

set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
inds = getEmptyImageDimensions(nim)
```

```
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
mask_empty_dim(nim, inds = inds)
func = function(...) maskEmptyImageDimensions(..., inds = inds)
func(arr)
func(nim)
func(rnifti)
func(timg)
func(limg)
```

`mask_img` *Mask Image*

Description

Takes an image, masks it by `mask`, and returns an object of class `nifti`

Usage

```
mask_img(img, mask, allow.NA = TRUE)
```

Arguments

`img`	object of class `nifti`
`mask`	array or object of class `nifti`, same dimensions as img
`allow.NA`	allow NAs in the mask

Value

Object of class nifti with values outside mask set to 0 if mask is 0 and NA if mask is NA and img if mask == 1

Examples

```
  set.seed(5)
  dims = rep(10, 3)
  arr = array(rnorm(prod(dims)), dim = dims)
  nim = oro.nifti::nifti(arr)
  mask = abs(nim) > 1
  masked = mask_img(nim, mask)
  mask[mask == 0] = NA
  na_masked = mask_img(nim, mask, allow.NA = TRUE)

set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
```

```
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
mask = nim > 0
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
func = function(...) mask_img(..., mask = mask)
func(arr)
func(nim)
func(rnifti)
func(timg)
lapply(limg, func)
```

`mask_vals` *Extract or Replace Values Inside a Mask*

Description

Methods that act on the `.Data` field in a NIfTI/ANALYZE image but only on values inside a mask.

Usage

```
mask_vals(object, mask)

mask_vals(object, mask) <- value

## S4 replacement method for signature 'nifti'
mask_vals(object, mask) <- value

## S4 replacement method for signature 'anlz'
mask_vals(object, mask) <- value

## S4 replacement method for signature 'array'
mask_vals(object, mask) <- value
```

Arguments

`object`	is an object of class `nifti` or `anlz`.
`mask`	is an object of class `nifti` or `anlz`.
`value`	is the value to assign to the `.Data` field.

Examples

```
set.seed(2022)
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
mask = img > 1.5
mask_vals(img, mask)
testthat::expect_equal(sum(mask_vals(img, mask)), 117.628200302518)
mask_vals(img, mask) = rep(4, sum(mask))
```

```
mask_vals(img, as(mask, "array")) = rep(4, sum(mask))
mask_vals(as(img, "array"),
    as(mask, "array")) = rep(4, sum(mask))
```

`mean.nifti` *Mean of Values in an Image*

Description

Computes the mean of values of an image with the option for a mask.

Usage

```
## S3 method for class 'nifti'
mean(x, ..., mask)

## S3 method for class 'anlz'
mean(x, ..., mask)
```

Arguments

`x`	Object of class `nifti`
`...`	Arguments passed to `mean.default`
`mask`	object to subset the image. If missing, then all values of the image are plotted.

Value

Output of `mean`

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
mask = img > 0
mean(img, mask = mask)
```

`minmax_img-methods` *Normalize Image using Range*

Description

Calculates the range of values in an image, then scales the image minimum to 0 and maximum to 1

Usage

```
minmax_img(img)

## S4 method for signature 'nifti'
minmax_img(img)

## S4 method for signature 'array'
minmax_img(img)

## S4 method for signature 'ANY'
minmax_img(img)

## S4 method for signature 'character'
minmax_img(img)

## S4 method for signature 'factor'
minmax_img(img)

## S4 method for signature 'list'
minmax_img(img)
```

Arguments

img	character path of image or an object of class nifti, or list of images

Value

A nifti object (or list of them) or class of object passed in if not specified

Examples

```
set.seed(5)
dims = rep(10, 4)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
mimg = minmax_img(nim)
marr = minmax_img(arr)
testthat::expect_equal(array(mimg, dim = dim(mimg)), marr)

set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
func = minmax_img
func(arr)
func(nim)
```

```
func(rnifti)
func(timg)
func(limg)
```

`multi_overlay` *Create Multi-Image Plot with Overlays*

Description

Creates a multi-row or multi-column plot with image slices and the potential for overlays as well.

Usage

```
multi_overlay(
  x,
  y = NULL,
  z = NULL,
  w = 1,
  mask = NULL,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
  zlim.y = NULL,
  ybreaks = NULL,
  plane = c("axial", "coronal", "sagittal"),
  xlab = "",
  ylab = "",
  axes = FALSE,
  direction = c("horizontal", "vertical"),
  par.opts = list(oma = c(0, 0, 0, 0), mar = rep(0, 4), bg = "black"),
  text = NULL,
  text.x = 0.5,
  text.y = 1.4,
  text.cex = 2.5,
  text.col = "white",
  main = NULL,
  main.col = text.col,
  main.cex = text.cex,
  NA.x = TRUE,
  NA.y = TRUE,
  pdim = NULL,
  useRaster = TRUE,
  ...
)

multi_overlay_center(x, y = NULL, ...)
```

Arguments

x	List of images of class nifti or character vector of filenames
y	List of images of class nifti or character vector of filenames. Same length as x.
z	Slice to display.
w	3D volume to display if x has 4-D elements
mask	nifti image to drop empty image dimensions if wanted. Passed to dropEmptyImageDimensions
col.x	Color to display x images
col.y	Color to display y images
zlim.x	Limits for x to plot
zlim.y	Limits for y to plot
ybreaks	(numeric) breaks for y to passed to image
plane	the plane of acquisition to be displayed
xlab	Label for x-axis
ylab	Label for y-axis
axes	Should axes be displayed
direction	Should images be a row or column? Ignored if mfrow is in par.opts
par.opts	Options to pass to par
text	Text to be displayed
text.x	Location of text in x-domain
text.y	Location of text in y-domain
text.cex	Multiplier for text font
text.col	Color for text and main.
main	Title for each plot
main.col	Color for main. Will default to text.col
main.cex	Multiplier for text font. Will default to text.cex
NA.x	Should 0's in x be set to NA?
NA.y	Should 0's in y be set to NA?
pdim	Pixel dimensions if passing in arrays. Will be overridden if x is a nifti object
useRaster	if TRUE, a bitmap raster is used to plot the image instead of polygons. Passed to image
...	Additional arguments to pass to image

Examples

```
set.seed(5)
dims = rep(10, 4)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,,c(3, 5)] = rpois(1000*2, lambda = 2)
nim = oro.nifti::nifti(arr)
mask = nim > 2
add_mask = nim[,,,1] > 0
imgs = img_ts_to_list(nim)
masks = img_ts_to_list(mask)
multi_overlay(imgs, masks)
multi_overlay(imgs, masks,
main = "hey", direction = "vertical", plane = "coronal")
multi_overlay(imgs, masks, mask = add_mask,
main = "hey")



## Not run:

 if (requireNamespace("brainR", quietly = TRUE)) {
   visits = 1:3
   y = paste0("Visit_", visits, ".nii.gz")
   y = system.file(y, package = "brainR")
   y = lapply(y, readnii)

   y = lapply(y, function(r){
     pixdim(r) = c(0, rep(1, 3), rep(0, 4))
     dropImageDimension(r)
   })

   x = system.file("MNI152_T1_1mm_brain.nii.gz",
                package = "brainR")
   x = readnii(x)
   mask = x >0
   x = lapply(visits, function(tmp){
       x
   })
   alpha = function(col, alpha = 1) {
       cols = t(col2rgb(col, alpha = FALSE)/255)
       rgb(cols, alpha = alpha)
   }
   multi_overlay(x, y,
         col.y = alpha(hotmetal(), 0.5),
         mask = mask,
         main = paste0("\n", "Visit ", visits),
         text = LETTERS[visits],
         text.x = 0.9,
         text.y = 0.1,
         text.cex = 3)
 }
```

```
## End(Not run)
```

`newnii` *Resets image parameters for a copied nifti object*

Description

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

Usage

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

Arguments

`img`	nifti object (or character of filename)
`...`	arguments to be passed to `datatype`

Value

object of type nifti

`niftiarr` *Make new nifti from array*

Description

Make new nifti object by passing in old nifti and array

Usage

```
niftiarr(img, arr)
```

Arguments

`img`	object of class nifti
`arr`	array to be passed in to .Data slot

Value

object of class nifti

nii.stub *Grab nii file stubname*

Description

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

Usage

```
nii.stub(x, bn = FALSE)
```

Arguments

x	character vector of filenames ending in .nii or .nii.gz
bn	Take basename of file?

Value

A character vector with the same length as x

orient_rpi *Reorient an Image to RPI orientation*

Description

Reorient an Image to RPI orientation

Usage

```
orient_rpi(file, verbose = TRUE)

orient_rpi_file(file, verbose = TRUE)

is_rpi_oriented(file, verbose = FALSE)
```

Arguments

file	Object of class nifti 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))

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))
```

---

ortho2 *Orthographic Display, added options*

Description

Copy of `oro.nifti`'s `orthographic` function with some tweaks such as adding L/R designations for left and right

Usage

```
ortho2(
  x,
  y = NULL,
  xyz = NULL,
  w = 1,
  col = gray(0:64/64),
  col.y = oro.nifti::hotmetal(),
  zlim = NULL,
  zlim.y = NULL,
  NA.x = FALSE,
  NA.y = TRUE,
  crosshairs = TRUE,
  col.crosshairs = "red",
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = c(0, 0, 0, ifelse(ycolorbar, 5, 0)),
  mar = rep(0, 4),
  bg = "black",
```

```
  text = NULL,
  text.color = "white",
  text.cex = 2,
  text.x = 32,
  text.y = 32,
  add.orient = FALSE,
  mfrow = c(2, 2),
  ybreaks = NULL,
  breaks = NULL,
  addlegend = FALSE,
  leg.x = 32,
  leg.y = 32,
  legend,
  leg.col,
  leg.title = NULL,
  leg.cex,
  window = NULL,
  ycolorbar = FALSE,
  clabels = TRUE,
  add = TRUE,
  pdim = NULL,
  useRaster = is.null(y),
  mask = NULL,
  ...
)
```

Arguments

x	is an object of class nifti or similar.
y	is an object of class nifti or similar for the overlay.
xyz	is the coordinate for the center of the crosshairs.
w	is the time point to be displayed (4D arrays only).
col	is grayscale (by default).
col.y	is hotmetal (by default).
zlim	is the minimum and maximum 'z' values passed into image.
zlim.y	is the minimum and maximum 'z' values passed into image for the overlay.
NA.x	Set any values of 0 in x to NA
NA.y	Set any values of 0 in y to NA
crosshairs	is a logical value for the presence of crosshairs in all three orthogonal planes (default = TRUE).
col.crosshairs	is the color of the crosshairs (default = red).
xlab	is set to "" since all margins are set to zero.
ylab	is set to "" since all margins are set to zero.
axes	is set to FALSE since all margins are set to zero.

oma	is the size of the outer margins in the par function.
mar	is the number of lines of margin in the par function.
bg	is the background color in the par function.
text	allows the user to specify text to appear in the fourth (unused) pane.
text.color	is the color of the user-specified text (default = "white").
text.cex	is the size of the user-specified text (default = 2).
text.x	x coordinate for text
text.y	y coordinate for text
add.orient	(logical) Add left/right, A/P, etc. orientation
mfrow	(numeric) layout of the 3 slices
ybreaks	(numeric) breaks for y to passed to image
breaks	(numeric) breaks for x to passed to image
addlegend	(logical) add legend?
leg.x	(numeric) x coordinate for legend
leg.y	(numeric) y coordinate for legend
legend	(character) legend text
leg.col	(character) Colors for legend
leg.title	(character) title for legend
leg.cex	(numeric) cex for legend
window	(vector) Length-2 vector to limit image to certain range
ycolorbar	(logical) Should a colorbar for y be plotted
clabels	Label for colorbar (see colorbar)
add	Should the y-plot be added or its own plot? Used in double_ortho
pdim	Pixel dimensions if passing in arrays. Will be overridden if x is a nifti object
useRaster	logical; if TRUE a bitmap raster is used to plot the image instead of polygons. Passed to image.
mask	If a mask is passed, drop_empty_dim is applied to both x and y
...	other arguments to the image function may be provided here.

See Also

orthographic

Examples

```
set.seed(10)
x = oro.nifti::nifti(array(rnorm(1000), dim = rep(10, 3)))
ortho2(x)
y = x > 2
mask = x > 2.5
ortho2(x, y)
```

```
ortho2(x, y, mask = mask, add.orient = TRUE)
ortho2(x, y, mask = mask, add.orient = TRUE, add = FALSE)
nim = RNifti::asNifti(x, internal = FALSE)
ortho2(nim, y, mask = mask)
neurobase::ortho2(nim, x, mask = mask,
ybreaks = seq(min(x), max(x), length.out = 65), ycolorbar = TRUE)

ortho2(nim, y, mask = mask, add = FALSE)
arr_x = as.array(x)
arr_y = as.array(y)
ortho2( arr_x)
ortho2( arr_x, arr_y, useRaster = FALSE)

set.seed(10)
x = oro.nifti::nifti(array(rnorm(10000), dim = rep(10, 4)))
y = x > 2
mask = x > 2.5
ortho2(x, y)

set.seed(10)
x = oro.nifti::nifti(array(rnorm(100), dim = rep(10, 2)))
y = x > 2
mask = x > 2.5
ortho2(x, y)
```

---

ortho_diff *Plot differences for Prediction and Gold Standard*

Description

Uses ortho2 to plot differences between a predicted binary image and the assumed ground truth (roi).

Usage

```
ortho_diff(
  img,
  pred,
  roi,
  xyz = NULL,
  cols = c("#56B4E9", "#D55E00", "#009E73"),
  levels = c("False Negative", "False Positive", "True Positive"),
  addlegend = TRUE,
  center = TRUE,
  leg.cex = 1.5,
  ...
)

multi_overlay_diff(
```

```
  x,
  pred,
  roi,
  z = NULL,
  cols = c("#56B4E9", "#D55E00", "#009E73"),
  ...
)
```

Arguments

img	image to be underlaid
pred	binary segmentation (prediction)
roi	binary manual segmentation (ground truth)
xyz	coordinate for the center of the crosshairs.
cols	colors for false negatives/positives
levels	labels for false negatives/positives
addlegend	add legend, passed to ortho2
center	run xyz on roi. Disregarded if xyz is not NULL
leg.cex	multiplier for legend size
...	arguments to be passed to ortho2 or multi_overlay
x	List of images of class nifti or character vector of filenames
z	slice to display

See Also

ortho2

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
roi = nim > 2
pred = nim > 1.5
ortho_diff(nim, pred, roi)
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
nim = oro.nifti::nifti(arr)
mask = nim > 2
pred = nim > 1.5
multi_overlay_diff(nim, roi = mask, pred = pred)
```

```
if (requireNamespace("brainR", quietly = TRUE)) {
  visits = 1:3
  y = paste0("Visit_", visits, ".nii.gz")
  y = system.file(y, package = "brainR")
  y = lapply(y, readnii)

  y = lapply(y, function(r){
    pixdim(r) = c(0, rep(1, 3), rep(0, 4))
    dropImageDimension(r)
  })

  x = system.file("MNI152_T1_1mm_brain.nii.gz",
                  package = "brainR")
  x = readnii(x)
  mask = x > 0
  alpha = function(col, alpha = 1) {
      cols = t(col2rgb(col, alpha = FALSE)/255)
      rgb(cols, alpha = alpha)
  }
  roi = y[[2]]
  pred = y
  multi_overlay_diff(x, roi = roi, pred = pred)
  multi_overlay_diff(x, roi = roi, pred = pred,
        mask = mask,
        main = paste0("\n", "Visit ", visits),
        text = LETTERS[visits],
        text.x = 0.9,
        text.y = 0.1,
        text.cex = 3)
}
```

`parse_img_ext` *Parse Image Extensions*

Description

Get image extensions from a filename

Usage

```
parse_img_ext(file)
```

Arguments

`file`	Character filename to parse

Value

Extension of file

Examples

```
parse_img_ext("blah.nii.gz")
parse_img_ext("blah.mnc")
parse_img_ext("blah.nii")
parse_img_ext("blah")
parse_img_ext("blah.img")
parse_img_ext("blah.hdr")
parse_img_ext("blah.hdr.gz")
```

---

quantile.nifti *Sample Quantiles*

Description

Computes sample quantiles for an image, with the option of a mask.

Usage

```
## S3 method for class 'nifti'
quantile(x, ..., mask)

## S3 method for class 'anlz'
quantile(x, ..., mask)
```

Arguments

x	Object of class nifti
...	Arguments passed to quantile
mask	object to subset the image. If missing, then all values of the image are used

Value

Output of quantile

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
mask = img > 0
quantile(img, mask = mask)
```

quantile_img *Create Quantile Image*

Description

Creates output image of the quantiles that each voxel is in, after applying the mask

Usage

```
quantile_img(img, mask = NULL, ...)
```

Arguments

img	Character vector, or object of class nifti
mask	Mask to determine cumulative distribution function (cdf) from
...	Additional arguments to pass to ecdf

Value

Object of class nifti

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
nim = oro.nifti::nifti(arr)
qimg = quantile_img(nim)
qarr = quantile_img(arr)
testthat::expect_equal(qarr, array(qimg, dim = dim(qarr)))
qimg = quantile_img(nim, mask = nim > 0)
```

randomize_mask *Randomize Image based on Mask*

Description

Randomize the values within a mask

Usage

```
randomize_mask(img, mask)
```

Arguments

img	Object of class nifti with values to be randomized
mask	Binary mask indicating which values should be randomized.

Value

Object of class nifti

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
nim = oro.nifti::nifti(arr)
mask = abs(nim) > 1
randomize_mask(nim, mask)
```

---

random_nifti *Create Random 'nifti' object*

Description

Create Random 'nifti' object

Usage

```
random_nifti(dim, ...)
```

Arguments

dim	dimensions for the 'nifti' object
...	arguments to send to nifti

Value

A 'nifti' object

Examples

```
random_nifti(c(10, 10, 2))
random_nifti(c(10, 10))
```

readNIfTI2 *readNIfTI with default non-reorientation*

Description

This function calls the readNIfTI function from the oro.nifti package, but sets the reorientation to FALSE by default

Usage

```
readNIfTI2(..., reorient = FALSE)

readnii(
  ...,
  reorient = FALSE,
  dtype = TRUE,
  drop_dim = TRUE,
  reset_slope = FALSE,
  warn = FALSE,
  rm_extensions = TRUE
)
```

Arguments

...	Arguments to pass to readNIfTI
reorient	Reorientation argument to pass to readNIfTI
dtype	Should datatyper be run after reading?
drop_dim	Should drop_img_dim be run after reading?
reset_slope	Reset slope/intercept of image
warn	Should warnings from readNIfTI be printed? If not, suppressWarnings is called. Also passed to datatyper
rm_extensions	should niftiExtensions be converted to simple nifti objects?

Value

nifti object

read_rpi *Read NIfTI file reoriented to RPI*

Description

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

Usage

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

Arguments

file	file name of the NIfTI file.
...	Arguments to pass to readnii
verbose	print diagnostics, passed to orient_rpi_file

Note

'read_rpi' uses 'RNifti' to ensure the reading orientation

remake_img *Remake Image from Vector*

Description

Wrapper function to take a vector of values and result in a nifti object

Usage

```
remake_img(vec, img, mask = NULL, warn = FALSE, ...)
```

Arguments

vec	vector of values to be in resulting image
img	object of class nifti to put vector into
mask	binary array/ nifti object to denote where vector values are to be.
warn	Should a warning be issued if defaulting to FLOAT32?
...	additional arguments passed to datatyper

Value

Object of class nifti

See Also

niftiarr

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
remake_img(c(nim), nim)
mask = nim > 5
vals = nim[mask]
vals = sqrt(vals)
remake_img(vals, nim, mask = mask)
```

---

remap_filename *Build Filename (usually for images)*

Description

This is a simple function that helps with the case where you want to construct a filename (usually for an image) with the same base of the filename, the same directory (default), but things added to the front or end of that base filename, with the same extension.

Usage

```
remap_filename(x, sub_dir = NULL, prefix = "", suffix = "")
```

Arguments

x	input filename/character vector
sub_dir	sub-directory for the new filename. If NULL, then the directory is the the same directory as x
prefix	string to put in front of base of filename
suffix	string to put at the end of base of filename

Value

Character vector

Examples

```
fname = file.path("/path/to/file", "original.nii.gz")
remap_filename(fname, prefix = "preproc_", "_with_gz")
fname = "original.nii"
remap_filename(fname, prefix = "note_", "_has_directory")
remap_filename(c(fname, "other.nii.gz"), prefix = "note_", "_has_directory")
```

`replaceEmptyImageDimensions-methods`
Replace Subsetting from Empty Image Dimensions

Description

Simple wrapper for subsetting an image with indices, dropping empty dimensions.

Usage

```
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
)

## S4 method for signature 'nifti'
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
)

## S4 method for signature 'character'
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
)

## S4 method for signature 'factor'
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
```

```
)

## S4 method for signature 'list'
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
)

## S4 method for signature 'array'
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
)

## S4 method for signature 'anlz'
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
)

## S4 method for signature 'ANY'
replaceEmptyImageDimensions(
  img,
  inds,
  target_dim,
  value = 0,
  reorient = FALSE,
  ...
)

replace_empty_dim(img, ...)
```

Arguments

img	image, nifti object, or array
inds	indices of subset from getEmptyImageDimensions or dropEmptyImageDimensions.

target_dim	Original dimension from which the data was subset, the final dimension of the output
value	value to replace in the image where outside the indices
reorient	Should image be reoriented if a filename?
...	not used

Value

Object of class nifti or array if nifti is not supplied

Note

replace_empty_dim is a shorthand for replaceEmptyImageDimensions with all the same arguments.

See Also

getEmptyImageDimensions, dropEmptyImageDimensions

Examples

```
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
arr[,,10] = 0
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg, timg))
inds = getEmptyImageDimensions(nim)
inds_arr = getEmptyImageDimensions(arr)
testthat::expect_equal(inds, inds_arr)

out = applyEmptyImageDimensions(nim, inds = inds)
result = replaceEmptyImageDimensions(out, inds = inds,
target_dim = dim(nim))
testthat::expect_equal(array(result, dim = dim(result)),
array(nim, dim = dim(nim)))
replace_empty_dim(out, inds = inds,
target_dim = dim(nim))

target_dim = dim(nim)

arr = array(out, dim = dim(out))
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
func = function(...) replaceEmptyImageDimensions(...,
target_dim = target_dim, inds = inds)
func(arr)
func(nim)
```

```
func(rnifti)
func(timg)
func(limg)
```

`replace_dropped_dimensions` *Remake Dropped Dimensions*

Description

This function is the reverse of `dropEmptyImageDimensions`. If `dropEmptyImageDimensions` was run, and the output is a list, usually if `keep_ind = TRUE`, this function reverses that.

Usage

```
replace_dropped_dimensions(img, inds, orig.dim)
```

Arguments

`img`	Object of class `nifti` where image dimensions were dropped.
`inds`	List of length 3 of indices from `dropEmptyImageDimensions` or `getEmptyImageDimensions`
`orig.dim`	Original dimension of pre-dropped image. Output image will have dimensions same as this value

Value

Object of class `nifti`

Examples

```
## Not run:
# nim is an object of class nifti
dd = dropEmptyImageDimensions(nim, keep_ind = TRUE)
remake = replace_dropped_dimensions(img = dd$outimg,
inds = dd$inds,
orig.dim = dd$orig.dim)
all.equal(nim, remake)

## End(Not run)
```

replace_outside_surface

Replace Values Outside Surface of image

Description

Determines values outside the surface of an image and gives a mask back with those values set to a replacement.

Usage

```
replace_outside_surface(
  img,
  value = 0,
  threshold = 0,
  replace_value = NA,
  reorient = FALSE
)
```

Arguments

img	nifti object or array
value	Value to check against. If zero, then replace_outside_surface will include any dimension that has fewer than threshold zeroes. May be a vector of values, matched with match
threshold	Include dimension if fewer than threshold voxels are in the slice
replace_value	Value to replace those outside the surface.
reorient	Should image be reoriented if a filename? Passed to check_nifti

Value

Creates an array of binary values. If img is a nifti object, then a nifti is returned

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(0, dim = dims)

arr[ 3:5, 4:6, c(2, 6:8, 5)] = 1
nim = oro.nifti::nifti(arr)
out = replace_outside_surface(nim, replace_value = 0)
out_arr = replace_outside_surface(arr, replace_value = 0)
testthat::expect_equal(out_arr, array(out, dim = dim(out)))
```

rescale_img *Image Rescaler*

Description

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

Usage

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

Arguments

filename	filename of image to be read into R or nifti object
pngname	filename of png of histogram of values of image to be made. For no png - set to NULL (default)
write.nii	logical - should the image be written.
outfile	if write.nii = TRUE, filename of output file
min.val	minimum value of image (default -1024 (for CT)). If no thresholding set to -Inf
max.val	maximum value of image (default 3071 (for CT)). If no thresholding set to Inf
ROIformat	if TRUE, any values < 0 will be set to 0
drop_dim	Should drop_img_dim be applied?
writer	character value to add to description slot of NIfTI header
...	extra methods to be passed to writenii

Value

Object of class nifti

Examples

```
img = nifti(array(rnorm(10^3, sd = 1000), dim = rep(10, 3)))
outfile = tempfile(fileext = ".nii.gz")
pngname = tempfile(fileext = ".png")
rescale_img(img, write.nii = TRUE, outfile = outfile,
pngname = pngname)
```

`reverse_orient_rpi` *Reverse Reorientation an Image to RPI orientation*

Description

Reverse Reorientation an Image to RPI orientation

Usage

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

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

Arguments

`file`	Object of class `nifti` or character path
`convention`	Convention of original image (usually from `orient_rpi`)
`orientation`	Vector of length 3 from original image (usually from `orient_rpi`)
`verbose`	print diagnostic messages

Value

Object of class `nifti`

Note

‘reverse_orient_rpi‘ and ‘reverse_orient_rpi_file‘ uses ‘RNifti‘ to ensure the reading orientation

robust_window *Window image based on quantiles of Image*

Description

Takes an image, finds the quantiles given by `probs`, then sets values outside these values to other values, as determined by `replace` argument passed to to `window_img`

Usage

```
robust_window(img, non_zero = FALSE, probs = c(0, 0.999), ..., mask = NULL)
```

Arguments

`img`	object of class nifti
`non_zero`	Should zeroes be excluded from the calculation of quantiles?
`probs`	quantiles to constrain the image these define the window sent to `window_img`
`...`	additional arguments sent to `window_img`
`mask`	binary image to use to to calculate quantiles

Value

Object of class nifti with values outside quantiles replaced by values depending on replace argument passed to `window_img`

same_dims *Check if Objects have Same Dimensions*

Description

Wrapper to check if multiple objects all have the same dimensions

Usage

```
same_dims(...)
```

Arguments

`...`	Arguments (matrices or arrays) where the dimension will be checked against the first object's dimension

Value

Logical indicating if all have the same dimensions or not

Examples

```
mat1 = matrix(1:9, ncol = 3)
mat2 = matrix(rnorm(9), ncol = 3)
mat3 = matrix(rnorm(16), ncol = 4)
mat4 = matrix(rnorm(9), ncol = 3)
same_dims(mat1, mat2)
same_dims(mat1, mat3)
same_dims(mat1, mat2, mat4)
```

---

separate_img-methods *Separate Labeled Image into Multiple Binary Images*

Description

Takes in an image, gets the unique values, then creates a list of binary images for each one of those values.

Usage

```
separate_img(img, levels = NULL, drop_zero = TRUE)

## S4 method for signature 'nifti'
separate_img(img, levels = NULL, drop_zero = TRUE)

## S4 method for signature 'array'
separate_img(img, levels = NULL, drop_zero = TRUE)

## S4 method for signature 'ANY'
separate_img(img, levels = NULL, drop_zero = TRUE)

## S4 method for signature 'factor'
separate_img(img, levels = NULL, drop_zero = TRUE)

## S4 method for signature 'character'
separate_img(img, levels = NULL, drop_zero = TRUE)

## S4 method for signature 'list'
separate_img(img, levels = NULL, drop_zero = TRUE)
```

Arguments

img	character path of image or an object of class nifti, or list of images
levels	if levels is given, then the separation is only done for those levels and not unique values of the image.
drop_zero	Should zeroes be dropped from the labels? Zero usually denotes background or non-interesting voxels

Value

A `nifti` object (or list of them) or class of object passed in if not specified

Note

Exact equalling is using ==

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
simg = separate_img(nim)
simg_arr = separate_img(arr)
slist = lapply(simg, function(x) array(x, dim(x)))
testthat::expect_equal(slist, simg_arr)

rnifti = RNifti::asNifti(nim)
timg = tempimg(nim)
limg = list(factor(timg), factor(timg))
func = separate_img
func(arr)
func(nim)
func(rnifti)
func(timg)
func(limg)
```

`slice_colour_df` *Slice a Image Color Data.frame*

Description

Slices a image color `data.frame` along the 3 planes (axial, coronal, sagittal) and returns it in a ggplot-ready format for faceting.

Usage

```
slice_colour_df(img_df, xyz = NULL)
```

Arguments

`img_df`	an image `data.frame`, usually from `img_colour_df`. Must have the columns: `dim1`, `dim2`, `dim3`, `colour`, and `value`.
`xyz`	coordinates to slice the `data.frame` in x, y, and z - domains

Value

A data.frame with x and y coordinates, colour, and intensity values, along with the associated planes that were sliced.

Examples

```
img = nifti(array(rnorm(10^3), dim = rep(10, 3)))
df = img_colour_df(img)
sliced = slice_colour_df(df, c(5, 5, 4))
```

---

subset_dti-methods *Subset DTI data based on b-values #'*

Description

Subset DTI data based on b-values #'

Usage

```
subset_dti(
  img,
  bvals,
  bvecs,
  b_step = 1,
  maximum = Inf,
  shells = NULL,
  verbose = TRUE,
  ...
)

## S4 method for signature 'nifti'
subset_dti(
  img,
  bvals,
  bvecs,
  b_step = 1,
  maximum = Inf,
  shells = NULL,
  verbose = TRUE,
  ...
)

## S4 method for signature 'ANY'
subset_dti(
  img,
  bvals,
  bvecs,
```

```
  b_step = 1,
  maximum = Inf,
  shells = NULL,
  verbose = TRUE,
  ...
)

## S4 method for signature 'character'
subset_dti(
  img,
  bvals,
  bvecs,
  b_step = 1,
  maximum = Inf,
  shells = NULL,
  verbose = TRUE,
  ...
)

## S4 method for signature 'list'
subset_dti(
  img,
  bvals,
  bvecs,
  b_step = 1,
  maximum = Inf,
  shells = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

`img`	character or `nifti` object
`bvals`	filename of b-values (assuming 1 row)
`bvecs`	filename of b-vectors (assuming 3 rows)
`b_step`	step of b-values to round to
`maximum`	maximum b-value threshold
`shells`	Shells to keep (after rounding)
`verbose`	print diagnostic messages
`...`	options passed to `checkimg`

Value

List of filenames of image, b-values, and b-vectors that were subsetted.

Author(s)

John Muschelli <muschellij2@gmail.com>

Examples

```
## Not run:
img = "~/Downloads/data.nii.gz"
bvals = "~/Downloads/bvals"
bvecs = "~/Downloads/bvals"
verbose = TRUE
b_step = 50
maximum = 1500
shells = NULL
sub = subset_dti(img = img, bvals = bvals, bvecs = bvecs,
maximum = 1500,
b_step = 50)


## End(Not run)
```

---

tempimg *Create temporary nii.gz file*

Description

Takes in a object of class nifti, writes it to a temp file, appends .nii.gz as `writenii` adds it.

Usage

```
tempimg(
  nim,
  gzipped = TRUE,
  checknan = TRUE,
  check_type = FALSE,
  warn = FALSE,
  drop_dim = TRUE,
  dtype = TRUE,
  ...
)
```

Arguments

nim	object of class nifti
gzipped	Should file be gzipped? Passed to `writenii`
checknan	Check for NAs or NaNs
check_type	Check the datatype for an image. Will run `datatyper`.
warn	Should warnings be displayed if `writenii` has any? Passed to `writenii`.

drop_dim	Should drop_img_dim be applied?
dtype	Should datatyper be run before writing? Should override 'check_type'
...	Passed to writenii.

Value

filename of output nii.gz

---

window_img *nifti image windower*

Description

Windows an image to min and max values and also changes cal_max and cal_min parameters

Usage

```
window_img(
  x,
  window = c(0, 100),
  replace = c("window", "missing", "zero"),
  ...
)
```

Arguments

x	is either a character name for the image or an object of class nifti
window	numeric of length 2 that gives min and max for window
replace	either "window" if the any values outside of c(min, max) are set to the min or max or "missing" for these to be set to NA
...	not used

Value

Object of class nifti

See Also

readnii

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
nim = oro.nifti::nifti(arr)
window_img(nim, window = c(1, 5))
window_img(nim, window = c(1, 5), replace = "missing")
tfile = tempimg(nim)
window_img(tfile)
window_img(as.factor(tfile))
arr = window_img(img_data(nim))
rnim = RNifti::readNifti(tfile)
window_img(rnim, window = c(1, 5))
range(window_img(rnim, window = c(1, 5)))
window_img(rnim, window = c(1, 5), replace = "missing")
range(window_img(rnim, window = c(1, 5), replace = "missing"))
```

---

writeNIfTI2 *writeNIfTI with default non-reorientation*

Description

This function calls the writeNIfTI function from the oro.nifti package, but makes sure to remove .nii extension and warnings can be suppressed.

Usage

```
writeNIfTI2(nim, filename, dtype = FALSE, compression = 9, ...)

writenii(
  nim,
  filename,
  dtype = TRUE,
  drop_dim = TRUE,
  warn = FALSE,
  compression = 9,
  rm_extensions = TRUE,
  ...
)
```

Arguments

nim	object of class nifti, passed to writeNIfTI
filename	path to save the NIfTI file. Suffix will be removed
dtype	Should datatyper be run before writing?

compression	compression level for gzipped files.
...	Additional arguments passed to writeNIfTI
drop_dim	Should drop_img_dim be run before writing?
warn	Should warnings from writeNIfTI be printed? If not, suppressWarnings is called
rm_extensions	should niftiExtensions be converted to simple nifti objects before writing?

Value

Nothing

Note

While writeNIfTI2 does not run datatyper as default, writenii does. Additional functionality will be added to writenii likely but will not to writeNIfTI2

Examples

```
set.seed(5)
dims = rep(10, 3)
arr = array(rnorm(prod(dims)), dim = dims)
nim = oro.nifti::nifti(arr)
rnifti = RNifti::asNifti(nim)
tfile = tempfile(fileext = ".nii.gz")
timg = writenii(nim, tfile, rm_extensions = TRUE, warn = TRUE)
timg = writeNIfTI2(nim, tfile, dtype = TRUE)
```

---

write_nifti *General NIfTI Writer*

Description

Writes out NIfTI files for multiple formats. Currently, for nifti objects and niftiImage objects from RNifti

Usage

```
write_nifti(nim, filename, ...)
```

Arguments

nim	Container for NIfTI Image
filename	Filename of image to be written out
...	additional arguments, to be passed to writeNifti or writenii

Value

Output from NIfTI writer

Examples

```
set.seed(5)
dims = rep(10, 4)
arr = array(rpois(prod(dims), lambda = 2), dim = dims)
nim = oro.nifti::nifti(arr)
tfile = tempfile(fileext = ".nii.gz")
write_nifti(nim, tfile)
rimg = RNifti::readNifti(tfile)
write_nifti(rimg, tfile)
```

---

xyz *Image Center of Gravity Wrapper*

Description

Find Center of Gravity of Image, after thresholding and take ceiling (wrapper for cog)

Usage

```
xyz(...)
```

Arguments

...	Arguments passed to cog

Value

Vector of length 3

Note

Just a convenience wrapper for `cog(ceil=TRUE)`

---

zero_pad *Zero pads an image*

Description

This function zero pads an image by a certain number of dimensions, usually for convolution

Usage

```
zero_pad(img, kdim, invert = FALSE, pad_value = 0L, ...)
```

Arguments

img	Array or class nifti
kdim	Dimensions of kernel
invert	(logical) If FALSE, does zero padding. If TRUE, reverses the process.
pad_value	Value to pad the image with. May use other values, such as -1024 for CT data
...	Options to copyNIfTIHeader

Value

Object of class nifti

Examples

```
kdim = c(3,3,5)
img = array(rnorm(30*30*36), dim = c(30, 30, 36))
pad = zero_pad(img, kdim)
back = zero_pad(pad, kdim, invert=TRUE)
all.equal(back, img)
```

---

zlimmer *Find Image z-limits*

Description

Helper function for plotting - returns zlim for image plot function

Usage

```
zlimmer(x, zlim = NULL, computed_range = NULL)
```

Arguments

x	Object of class nifti
zlim	A user-specified zlim. If NULL, will calculate how ortho2 would calculate zlim
computed_range	If the range of the data was already computed, this can be passed in and will be used if relevant.

Value

If zlim = NULL, then vector of length 2, otherwise returns zlim

`zscore_img` *Get Z-score over a margin of an img*

Description

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

Usage

```
zscore_img(
  img,
  mask = NULL,
  margin = NULL,
  centrality = c("mean", "median", "trimmed_mean"),
  variability = c("sd", "iqrdiff", "mad", "maddiff", "iqr", "trimmed_sd"),
  trim = 0.2,
  remove.na = TRUE,
  remove.nan = TRUE,
  remove.inf = TRUE,
  remove.val = 0,
  remask = TRUE
)
```

Arguments

`img`	character path of image or an object of class nifti
`mask`	character path of mask or an object of class nifti
`margin`	Margin of image to z-score over (`NULL` - whole brain, 3-Axial, 2-Sagittal, 1-Coronal)
`centrality`	(character) Measure to center the data, either mean or median
`variability`	(character) Measure to scale the data
`trim`	if centrality is `trimmed_mean` or variability is `trimmed_sd`, then the amount of trimming
`remove.na`	(logical) change NAs to remove.val
`remove.nan`	(logical) change NaN to remove.val
`remove.inf`	(logical) change Inf to remove.val
`remove.val`	(logical) value to put the NA/NaN/Inf
`remask`	(logical) Should the image be remasked after normalizing?

Value

Array of object of class nifti

See Also

aperm

Examples

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

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

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

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

z = zscore_img(img, margin=NULL)
ztrim = zscore_img(img, margin=NULL,
centrality = "trimmed_mean", variability = "trimmed_sd")

z = zscore_img(img, centrality = "median", variability = "iqr")
z = zscore_img(img, centrality = "median", variability = "iqrdiff")
z = zscore_img(img, centrality = "median", variability = "maddiff")
```

Index