

Package ‘RNifti’

July 5, 2018

Version 0.8.1

Date 2018-07-05

Title Fast R and C++ Access to NIFTI Images

Imports Rcpp (>= 0.11.0)

Suggests testthat (>= 0.11.0), reportr

Enhances oro.nifti, tractor.base

LinkingTo Rcpp

Description Provides very fast access to images stored in the NIFTI-1 file format <<http://www.nitrc.org/docman/view.php/26/64/nifti1.h>>, with seamless synchronisation between compiled C and interpreted R code. Not to be confused with 'RNiftyReg', which provides tools for image registration.

License GPL-2

URL <https://github.com/jonclayden/RNifti>

BugReports <https://github.com/jonclayden/RNifti/issues>

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation yes

Author Jon Clayden [cre, aut],
Bob Cox [aut],
Mark Jenkinson [aut],
Matt Hall [ctb],
Rick Reynolds [ctb],
Kate Fissell [ctb],
Jean-loup Gailly [cph],
Mark Adler [cph]

Maintainer Jon Clayden <code@clayden.org>

Repository CRAN

Date/Publication 2018-07-05 15:00:03 UTC

R topics documented:

dim.internalImage	2
dumpNifti	3
ndim	4
niftiVersion	4
pixdim	5
readNifti	6
retrieveNifti	7
updateNifti	8
voxelToWorld	9
writeNifti	10
xform	11

Index	13
--------------	-----------

dim.internalImage	<i>Internal images</i>
-------------------	------------------------

Description

An internal image is a simple R object with a few attributes including a pointer to an internal C structure, which contains the full image data. They are used in the package for efficiency, but can be converted to a normal R array using the `as.array` method. Attributes of these objects should not be changed.

Usage

```
## S3 method for class 'internalImage'
dim(x)

## S3 replacement method for class 'internalImage'
dim(x) <- value

## S3 method for class 'internalImage'
as.array(x, ...)
```

Arguments

<code>x</code>	An "internalImage" object.
<code>value</code>	Not used. Changing the dimensions of an internal image is invalid, and will produce an error.
<code>...</code>	Additional parameters to methods. Currently unused.

Author(s)

Jon Clayden <code@clayden.org>

`dumpNifti`*Dump the contents of an internal NIFTI-1 object*

Description

This function extracts the contents of an internal NIFTI-1 object into an R list. No processing is done to the elements.

Usage

```
dumpNifti(image)

## S3 method for class 'niftiHeader'
print(x, ...)
```

Arguments

<code>image</code>	An image, in any acceptable form (see retrieveNifti).
<code>x</code>	A "niftiHeader" object.
<code>...</code>	Ignored.

Value

For `dumpNifti`, a list of class "niftiHeader", with named components corresponding to the elements in a raw NIFTI-1 file.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIFTI-1 standard (<http://www.nitrc.org/docman/view.php/26/64/nifti1.h>).

Examples

```
dumpNifti(system.file("extdata", "example.nii.gz", package="RNifti"))

# Default header for a standard R array
dumpNifti(array(0L, dim=c(10,10)))
```

ndim *Number of dimensions*

Description

This function is shorthand for `length(dim(object))`.

Usage

```
ndim(object)
```

Arguments

object An R object.

Value

The dimensionality of the object. Objects without a `dim` attribute will produce zero.

Author(s)

Jon Clayden <code@clayden.org>

Examples

```
ndim(array(0L, dim=c(10,10)))
```

niftiVersion *Check the format version of a file*

Description

This function identifies the likely NIfTI format variant used by one or more files on disk.

Usage

```
niftiVersion(file)
```

Arguments

file A character vector of file names.

Value

A vector of integers, of the same length as `file`. Each element will be 0 for ANALYZE format (the precursor to NIfTI-1), 1 for NIfTI-1 (which is now most common), 2 for NIfTI-2, or -1 if the file doesn't exist or doesn't look plausible in any of these formats.

Note

NIfTI-2 format, mostly a variant of NIfTI-1 with wider datatypes used for many fields, is not currently supported for reading, but it is detected by this function.

Author(s)

Jon Clayden <code@clayden.org>

See Also

[readNifti](#)

Examples

```
path <- system.file("extdata", "example.nii.gz", package="RNifti")
niftiVersion(path)      # 1
```

pixdim

Pixel dimensions and units

Description

By default, these generic functions return or replace the "pixdim" and "pixunits" attributes of their arguments. These represent the physical step size between pixel or voxel centre points, and the spatial and temporal units that they are given in. The former defaults to 1 in each dimension, if there is no attribute.

Usage

```
pixdim(object)

## Default S3 method:
pixdim(object)

pixdim(object) <- value

## Default S3 replacement method:
pixdim(object) <- value

pixunits(object)

## Default S3 method:
pixunits(object)

pixunits(object) <- value

## Default S3 replacement method:
pixunits(object) <- value
```

Arguments

object	An R object, generally an image.
value	Numeric vector of pixel dimensions along each axis, or character vector of abbreviated units. For dimensions, a scalar value will be recycled if necessary.

Value

`pixdim` returns a numeric vector of pixel dimensions. `pixunits` returns a character vector of length up to two, giving the spatial and temporal unit names.

Author(s)

Jon Clayden <code@clayden.org>

Examples

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
pixdim(im)
pixunits(im)
```

readNifti	<i>Read a NIFTI-1 format file</i>
-----------	-----------------------------------

Description

This function reads one or more NIFTI-1 files into R, using the standard NIFTI-1 C library.

Usage

```
readNifti(file, internal = FALSE, volumes = NULL)
```

Arguments

file	A character vector of file names.
internal	Logical value. If FALSE (the default), an array of class "niftiImage", containing the image pixel or voxel values, will be returned. If TRUE, the return value will be an object of class "internalImage", which contains only minimal metadata about the image. Either way, the return value has an attribute which points to a C data structure containing the full image.
volumes	An integer vector giving the volumes to read (counting along all dimensions beyond the third jointly), or NULL, the default, in which case every volume is read. This cannot currently be set differently for each file read.

Value

An array or internal image, with class "niftiImage" (and possibly also "internalImage"), or a list of such objects if `file` has length greater than one.

Note

If the internal argument is FALSE (the default), the data type of the image pointer will be set to match one of R's native numeric data types, i.e., 32-bit signed integer or 64-bit double-precision floating-point. In these circumstances the data type reported by the `dumpNifti` function will therefore not, in general, match the storage type used in the file. See also the `datatype` argument to `writeNifti`.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIFTI-1 standard (<http://www.nitrc.org/docman/view.php/26/64/nifti1.h>).

See Also

`writeNifti`

Examples

```
path <- system.file("extdata", "example.nii.gz", package="RNifti")
readNifti(path)
readNifti(path, internal=TRUE)
```

retrieveNifti

Obtain an internal NIFTI representation of an object

Description

This function converts filenames, arrays and other image classes into an object of class "internalImage".

Usage

```
retrieveNifti(object)
```

Arguments

object Any suitable object (see Details).

Details

If the object has an internal NIFTI pointer, that will be retrieved directly. Otherwise, if it is a string, it will be taken to be a filename. If it looks like a "nifti" object (from package `oro.nifti`), or an "MriImage" object (from package `tractor.base`), a conversion will be attempted. A list will be assumed to be of the form produced by `dumpNifti`. Finally, a numeric array or matrix will be converted using default image parameters.

Value

An internal image.

Author(s)

Jon Clayden <code@clayden.org>

See Also

[readNifti](#), [updateNifti](#)

updateNifti

Update an internal NIfTI-1 object using a template

Description

This function adds or updates the internal NIfTI-1 object for an array, using metadata from the template. The dimensions and, if available, pixel dimensions, from the image will replace those from the template.

Usage

```
updateNifti(image, template = NULL, datatype = "auto")
```

Arguments

image	A numeric array.
template	An image, in any acceptable form (see retrieveNifti), or a named list of NIfTI-1 properties like that produced by dumpNifti . The default of NULL will have no effect.
datatype	The NIfTI datatype to use within the internal image. The default, "auto" uses the R type. Other possibilities are "float", "int16", etc., which may be preferred to reduce object size. However, no checks are done to ensure that the coercion maintains precision, and this option is for advanced usage only.

Details

If `template` is a complete list of NIfTI-1 header fields, like that produced by [dumpNifti](#), or an image, then it will be used to create the internal object, and then the data and metadata associated with the image will overwrite the appropriate parts. If `template` is an incomplete list, the image will be used to create the internal object, and then the specified fields will be overwritten from the list. This allows users to selectively update certain fields while leaving others alone (but see the note below).

Datatype information in a list `template` is ignored. The datatype can only be changed using the `datatype` argument, but in this case the internal object gets out of sync with the R array, so an internal image is returned to avoid the mismatch. Changing the internal datatype in this way is for advanced usage only.

Value

A copy of the original image, with its internal image attribute set or updated appropriately. If datatype is not "auto" then the result is an internal image.

Note

The `scl_slope` and `scl_inter` fields affect the numerical interpretation of the pixel data, so it is impossible in general to change them without also changing the array values on both the C and the R side. Therefore, to avoid unexpected side-effects, these fields are not affected by this function.

Author(s)

Jon Clayden <code@clayden.org>

voxelToWorld	<i>Transform points between voxel and "world" coordinates</i>
--------------	---

Description

These functions are used to transform points from dimensionless pixel or voxel coordinates to "real-world" coordinates, typically in millimetres, and back. Actual pixel units can be obtained using the [pixunits](#) function. The `origin` function gives the voxel coordinates of the real-world origin.

Usage

```
voxelToWorld(points, image, simple = FALSE, ...)
```

```
worldToVoxel(points, image, simple = FALSE, ...)
```

```
origin(image, ...)
```

Arguments

<code>points</code>	A vector giving the coordinates of a point, or a matrix with one point per row.
<code>image</code>	The image in whose space the points are given.
<code>simple</code>	A logical value: if TRUE then the transformation is performed simply by rescaling the points according to the voxel dimensions recorded in the <code>image</code> . Otherwise the full xform matrix is used.
<code>...</code>	Additional arguments to <code>xform</code> .

Value

A vector or matrix of transformed points.

Note

Voxel coordinates are assumed by these functions to use R's indexing convention, beginning from 1.

Author(s)

Jon Clayden <code@clayden.org>

See Also

[xform](#), [pixdim](#), [pixunits](#)

Examples

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))

# Find the origin
origin(im)
```

writeNifti

Write a NIfTI-1 format file

Description

This function writes an image to NIfTI-1 format, using the standard NIfTI-1 C library.

Usage

```
writeNifti(image, file, template = NULL, datatype = "auto")
```

Arguments

image	An image, in any acceptable form (see retrieveNifti).
file	A character string containing a file name.
template	An optional template object to derive NIfTI-1 properties from. Passed to updateNifti if image is an array.
datatype	The NIfTI datatype to use when writing the data out. The default, "auto" uses the R type or, for internal images, the original datatype. Other possibilities are "float", "int16", etc., which may be preferred to reduce file size. However, no checks are done to ensure that the coercion maintains precision.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIfTI-1 standard (<http://www.nitrc.org/docman/view.php/26/64/nifti1.h>).

See Also

[readNifti](#), [updateNifti](#)

Examples

```
## Not run: writeNifti(im, "image.nii.gz", datatype="float")
```

xform

Obtain or replace the “xform” transforms for an image

Description

These functions convert the “qform” or “sform” information in a NIfTI header to or from a corresponding affine matrix. These two “xform” mechanisms are defined by the NIfTI standard, and may both be in use in a particular image header. They define the relationship between the storage order of the image and real space.

Usage

```
xform(image, useQuaternionFirst = TRUE)

qform(x) <- value

sform(x) <- value

orientation(x, useQuaternionFirst = TRUE)

orientation(x) <- value
```

Arguments

image, x	An image, in any acceptable form (see retrieveNifti).
useQuaternionFirst	A single logical value. If TRUE, the “qform” matrix will be used first, if it is defined; otherwise the “sform” matrix will take priority.
value	A new 4x4 qform or sform matrix, or orientation string. If a matrix has a “code” attribute, the appropriate qform or sform code is also set.

Value

For `xform`, an affine matrix corresponding to the “qform” or “sform” information in the image header. For orientation, a string with three characters indicating the (approximate) orientation of the image. Each character may be ‘R’ for left-to-right, ‘L’ for right-to-left, ‘A’ for posterior-to-anterior, ‘P’ for anterior-to-posterior, ‘S’ for inferior-to-superior, or ‘I’ for superior-to-inferior. The replacement forms return the modified object.

Note

The `qform` and `sform` replacement functions are for advanced users only. Modifying the transforms without knowing what you’re doing is usually unwise, as you can make the image object inconsistent.

Author(s)

Jon Clayden <code@clayden.org>

References

The NIfTI-1 standard (<http://www.nitrc.org/docman/view.php/26/64/nifti1.h>) is the definitive reference on “xform” conventions.

Examples

```
im <- readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
xform(im)

# Remove the qform information
qform(im) <- structure(diag(4), code=0L)

# The same as above, since the sform is unmodified
xform(im)
```

Index

`as.array.internalImage`
 (dim.internalImage), 2

`dim.internalImage`, 2
`dim<-.internalImage`
 (dim.internalImage), 2
`dumpNifti`, 3, 7, 8

`internalImage` (dim.internalImage), 2

`ndim`, 4
`niftiVersion`, 4

`orientation` (xform), 11
`orientation<-` (xform), 11
`origin` (voxelToWorld), 9

`pixdim`, 5, 10
`pixdim<-` (pixdim), 5
`pixunits`, 9, 10
`pixunits` (pixdim), 5
`pixunits<-` (pixdim), 5
`print.niftiHeader` (dumpNifti), 3

`qform<-` (xform), 11

`readNifti`, 5, 6, 8, 11
`retrieveNifti`, 3, 7, 8, 10, 11

`sform<-` (xform), 11

`updateNifti`, 8, 8, 10, 11

`voxelToWorld`, 9

`worldToVoxel` (voxelToWorld), 9
`writeNifti`, 7, 10

`xform`, 9, 10, 11