Package 'oro.dicom'

October 28, 2019

Version 0.5.3

Date 2019-10-25

Title Rigorous - DICOM Input / Output

Depends R (>= 2.14.0)

Suggests testthat, hwriter

Imports utils, oro.nifti (>= 0.4.0)

Description Data input/output functions for data that conform to the Digital Imaging and Communications in Medicine (DICOM) standard, part of the Rigorous Analytics bundle.

License BSD_3_clause + file LICENSE

URL http://rig.oro.us.com, http://rigorousanalytics.blogspot.com

LazyData true

LazyDataCompression gzip

Encoding UTF-8

RoxygenNote 6.1.1

NeedsCompilation no

Author Brandon Whitcher [aut, cre]

Maintainer Brandon Whitcher <bwhitcher@gmail.com>

Repository CRAN

Date/Publication 2019-10-28 19:40:05 UTC

R topics documented:

`create3D` *Create Arrays from DICOM Headers/Images*

Description

A DICOM list structure is used to produce a multi-dimensional array representing a single acquisition of medical imaging data.

Usage

```
create3D(dcm, mode = "integer", transpose = TRUE, pixelData = TRUE,
  mosaic = FALSE, mosaicXY = NULL, sequence = FALSE,
  boffset = NULL)

create4D(dcm, mode = "integer", transpose = TRUE, pixelData = TRUE,
  mosaic = FALSE, mosaicXY = NULL, nslices = NULL, ntimes = NULL,
  instance = TRUE, sequence = FALSE)
```

Arguments

`dcm`	is the DICOM list structure (if `pixelData` = `TRUE`) or the DICOM header information (if `pixelData` = `FALSE`).
`mode`	is a valid character string for `storage.mode`.
`transpose`	is available in order to switch the definition of rows and columns from DICOM (default = `TRUE`.
`pixelData`	is a logical variable (default = `TRUE`) that is associated with the DICOM image data being pre-loaded.
`mosaic`	is a logical variable (default = `FALSE`) to denote storage of the data in Siemens 'Mosaic' format.
`mosaicXY`	is a vector of length two that provides the (x,y) dimensions of the individual images. Default behavior is to use the AcquisitonMatrix to determine the (x,y) values.

sequence	is a logical variable (default = FALSE) on whether to look in SequenceItem entries for DICOM header information.
boffset	is the number of bytes to skip at the beginning of the DICOM file (default = NULL which lets the code determine the starting point).
nslices	is the third dimension of the array. Attempts are made to determine this number from the DICOM data.
ntimes	is the fourth dimension of the array. Attempts are made to determine this number from the DICOM data.
instance	is a logical variable (default = TRUE) that determines whether or not to access the InstanceNumber field in the DICOM header to help order the slices.

Value

Multi-dimensional array of medical imaging data.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

array, readDICOM, storage.mode

Examples

```
load(system.file("hk-40/hk40.RData", package="oro.dicom"))
dcmList <- hk40
dcmImage <- create3D(dcmList)
image(dcmImage[,,1], col=grey(0:64/64), axes=FALSE, xlab="", ylab="",
      main=paste("First Slice from HK-40"))
imagePositionPatient <- attributes(dcmImage)$ipp
dSL <- abs(diff(imagePositionPatient[,3]))
plot(dSL, ylim=range(range(dSL) * 1.5, 0, 10), xlab="Image", ylab="mm",
     main="Difference in Slice Location")

## Not run:
## pixelData = FALSE
## The DICOM image data are read from create3D()
## This may save on memory for large batches of DICOM data
dcmList <- readDICOM(system.file("hk-40", package="oro.dicom"),
                     pixelData=FALSE)
dcmImage <- create3D(dcmList, pixelData=FALSE)
image(dcmImage[,,1], col=grey(0:64/64), axes=FALSE, xlab="", ylab="",
      main=paste("First Slice from HK-40 (again)"))
```

```
## End(Not run)
## mosaic = TRUE
mosaicFile <- system.file("dcm/MR-sonata-3D-as-Tile.dcm", package="oro.dicom")
dcm <- readDICOMFile(mosaicFile)
image(t(dcm$img), col=grey(0:64/64), axes=FALSE, xlab="", ylab="",
      main="Siemens MOSAIC")
dcmImage <- create3D(dcm, mode="integer", mosaic=TRUE)
z <- trunc(dim(dcmImage)[3]/2)
image(dcmImage[,,z], col=grey(0:64/64), axes=FALSE, xlab="", ylab="",
      main=paste("Slice", z, "from Siemens MOSAIC"))
```

---

dec2base *Convert Decimal to Base N Number in String*

Description

This function converts the nonnegative integer to the specified base.

Usage

```
dec2base(n, base, len = 0)

dec2hex(n, len = 0)
```

Arguments

n	Non-negative integer.
base	Number between 2 and 36.
len	Length of the character string.

Details

This function converts the nonnegative integer n to the specified base, where n must be a nonnegative integer smaller than 2^52, base must be an integer between 2 and 36 and len suggests the length of the character string.

Value

The returned argument is a string.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

Examples

```
x <- dec2base(23, 2)
```

`dicom.dic` *Lookup Tables for DICOM Header Information*

Description

Lookup Tables for DICOM Header Information

Usage

```
data(dicom.dic)
```

Format

An object of class `data.frame` with 4188 rows and 5 columns.

Source

See references.

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org
http://en.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine

See Also

`readDICOM`, `readDICOMFile`.

`dicom2analyze` *Convert DICOM Header to Analyze*

Description

A subset of header information from DICOM is placed into Analyze 7.5 format.

Usage

```
dicom2analyze(dcm, datatype = 4, reslice = TRUE, DIM = 3,
  descrip = "SeriesDescription", ...)
```

Arguments

dcm	DICOM object containing both header and image information.
datatype	is an integer that denotes the type of data contained in each voxel. See convert.datatype.anlz or the ANALYZE documentation for more details.
reslice	Logical variable (default = TRUE) indicating if the data volume should be resliced.
DIM	The dimension of the array to be used (default = 3D).
descrip	DICOM header field(s) to be included in the descrip
...	Arguments to be passed to anlz

Details

See the references.

Value

An object of class anlz.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Analyze 7.5
http://rportal.mayo.edu/bir/ANALYZE75.pdf

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

convert.datatype.anlz, dicom2nifti, anlz

Examples

```
## Not run:
dcmList <- dicomSeparate(system.file("hk-40", package="oro.dicom"))
require("oro.nifti")
dcmAnlz <- dicom2analyze(dcmList, datatype=4, mode="integer")
image(dcmAnlz)
orthographic(dcmAnlz)

## End(Not run)
```

dicom2nifti *Convert DICOM Header to NIfTI*

Description

A subset of header information from DICOM is placed into NIfTI-1 format.

Usage

```
dicom2nifti(dcm, datatype = 4, units = c("mm", "sec"),
  rescale = FALSE, reslice = TRUE, qform = TRUE, sform = TRUE,
  DIM = 3, descrip = "SeriesDescription", aux.file = NULL, ...)
```

Arguments

dcm	DICOM object containing both header and image information.
datatype	is an integer that denotes the type of data contained in each voxel. See convert.datatype or the NIfTI documentation for more details.
units	Spatial and temporal units for xyzt
rescale	Should slope and intercept parameters be extracted from the DICOM headers and saved?
reslice	Logical variable (default = TRUE) indicating if the data volume should be resliced.
qform	Logical variable (default = TRUE) indicating if the 3D image orientation should be used.
sform	Logical variable (default = TRUE) indicating if the 3D image orientation should be used.
DIM	The dimension of the array to be used (default = 3D).
descrip	DICOM header field(s) to be included in the descrip slot for the nifti class object.
aux.file	Character string to be included in the aux_file slot for the nifti class object.
...	Arguments to be passed to nifti

Details

See the references.

Value

An object of class nifti.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

NIfTI-1
http://nifti.nimh.nih.gov/nifti-1

See Also

convert.datatype, dicom2analyze, nifti

Examples

```
## Not run:
dcmList <- dicomSeparate(system.file("hk-40", package="oro.dicom"))
require("oro.nifti")
dcmNifti <- dicom2nifti(dcmList, datatype=4, mode="integer")
qform(dcmNifti)
sform(dcmNifti)
image(dcmNifti)
orthographic(dcmNifti)

## End(Not run)
```

---

dicomTable *Construct Data Frame from DICOM Headers*

Description

A data frame is created given the valid DICOM fields provided by the user.

Usage

```
dicomTable(hdrs, stringsAsFactors = FALSE, collapse = "-",
  colSort = TRUE, verbose = FALSE, debug = FALSE)
```

Arguments

hdrs	List object of DICOM headers.
stringsAsFactors	Logical variable to be passed to data.frame.
collapse	Character string used to paste DICOM group, element and value fields.
colSort	Logical variable (default = TRUE) to sort column names in the table.
verbose	Flag to provide text-based progress bar (default = FALSE).
debug	Logical variable (default = FALSE) that regulates to display of intermediate processing steps.

Value

Data frame where the rows correspond to images and the columns correspond to the UNION of all DICOM fields across all files in the list.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Whitcher, B., V. J. Schmid and A. Thornton (2011). Working with the DICOM and NIfTI Data Standards in R, *Journal of Statistical Software*, **44** (6), 1–28. http://www.jstatsoft.org/v44/i06

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

---

extractHeader *Extract Single Field from DICOM Headers*

Description

A particular DICOM field is extracted for a collection of DICOM headers.

Usage

```
extractHeader(hdrs, string, numeric = TRUE, names = FALSE,
  inSequence = TRUE)
```

Arguments

hdrs	List object of DICOM headers.
string	DICOM field name.
numeric	Logical; values are converted to numbers when TRUE.
names	Logical; file names are kept with elements of the vector.
inSequence	Logical; whether or not to look into SequenceItem elements.

Details

The DICOM field is extracted from each DICOM header and placed into a vector.

Value

Vector of values from the requested DICOM field.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

readDICOM

Examples

```
x <- readDICOMFile(system.file("dcm/Abdo.dcm", package="oro.dicom"))
seriesDescription <- extractHeader(x$hdr, "SeriesDescription", numeric=FALSE)
IOP <- extractHeader(x$hdr, "ImageOrientationPatient", numeric=FALSE)
```

---

getOrientation *Convert Direction Cosines to Anatomical Direction*

Description

For cross-sectional DICOM images the orientation must be derived from the Image Orientation (Patient) direction cosines.

Usage

```
getOrientation(xyz, delta = 0.0001)
```

Arguments

xyz	is a vector of direction cosines from "ImageOrientationPatient" (0020,0037).
delta	is the tolerance around zero for comparisons.

Details

C.7.6.2.1.1 Image Position And Image Orientation. The Image Position (0020,0032) specifies the x, y, and z coordinates of the upper left hand corner of the image; it is the center of the first voxel transmitted. Image Orientation (0020,0037) specifies the direction cosines of the first row and the first column with respect to the patient. These Attributes shall be provide as a pair. Row value for the x, y, and z axes respectively followed by the Column value for the x, y, and z axes respectively. The direction of the axes is defined fully by the patient's orientation. The x-axis is increasing to the left hand side of the patient. The y-axis is increasing to the posterior side of the patient. The z-axis is increasing toward the head of the patient. The patient based coordinate system is a right handed system; i.e., the vector cross product of a unit vector along the positive x-axis and a unit vector along the positive y-axis is equal to a unit vector along the positive z-axis.

Value

Anatomical direction shall be designated by the capital letters:

A	anterior
P	posterior
R	right
L	left
H	head
F	foot

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

http://www.dclunie.com/medical-image-faq/html/part2.html

See Also

swapDimension

---

header2matrix *Converts DICOM Header Field to a Matrix*

Description

Converts a vector of DICOM header information, assuming there are multiple entries per element of the vector, into a matrix.

Usage

```
header2matrix(hdr, ncol, sep = " ", byrow = TRUE)
```

Arguments

hdr	is the result from extracting information from a DICOM header field; e.g., using extractHeader.
ncol	is the number of columns.
sep	is the character string required to split entries in the header field.
byrow	is a logical variable (default = TRUE) telling the routine to populate the matrix by rows then columns.

Value

Matrix with length(hdr) rows and ncol columns.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

extractHeader, matrix

Examples

```
x <- readDICOMFile(system.file("dcm/Abdo.dcm", package="oro.dicom"))
pixelSpacing <- extractHeader(x$hdr, "PixelSpacing", numeric=FALSE)
pSmat <- header2matrix(pixelSpacing, ncol=2)
IOP <- extractHeader(x$hdr, "ImageOrientationPatient", numeric=FALSE)
IOPmat <- header2matrix(IOP, ncol=6)
```

---

matchHeader *Match String to DICOM Header Field*

Description

A convenient wrapper function that utilizes internal functions to match character strings with the DICOM header information.

Usage

```
matchHeader(hdr, string)
```

Arguments

hdr	is the result from extracting information from a DICOM header field; e.g., using extractHeader.
string	is a character string to be matched with the DICOM header.

Value

A logical vector of length length(hdr).

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

extractHeader

Examples

```
x <- readDICOMFile(system.file("dcm/Abdo.dcm", package="oro.dicom"))
modality <- extractHeader(x$hdr, "Modality", numeric=FALSE)
matchHeader(modality, "mr") # case insensitive by default
```

---

nextHeader *Check String Against DICOM Header Field to Produce Error Message or NEXT*

Description

A function designed to break out of loops given information (or the lack thereof) contained in the DICOM header.

Usage

```
nextHeader(dcm, string, reference, str.warning, htmlfile = NULL,
  heading = 3, numeric = FALSE)
```

Arguments

dcm	is the DICOM list structure.
string	is a character string to be matched with the DICOM header.
reference	is the scalar/vector of character strings to check against the DICOM header output.
str.warning	is a text string for the warning.
htmlfile	is the **hwriter** object for the HTML file (default = NULL.
heading	is the HTML tag <H?> (default = 3).
numeric	is the argument to be passed to matchHeader.

Value

An expression to be evaluated and HTML content.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

extractHeader, matchHeader

---

orthogonal-planes *Orthogonal Planes*

Description

Functions to test the orientation for a single slice.

Usage

```
is.axial(imageOrientationPatient, axial = c("L", "R", "A", "P"))

is.coronal(imageOrientationPatient, coronal = c("L", "R", "H", "F"))

is.sagittal(imageOrientationPatient, sagittal = c("A", "P", "H", "F"))
```

Arguments

imageOrientationPatient	A vector of length six taken from the DICOM header field "ImageOrientationPatient".
axial	Characters that are valid in defining an 'axial' slice.
coronal	Characters that are valid in defining a 'coronal' slice.
sagittal	Characters that are valid in defining a 'sagittal' slice.

Value

Logical value.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

See Also

getOrientation

Examples

```
x <- readDICOMFile(system.file("dcm/Abdo.dcm", package="oro.dicom"))
iop <- header2matrix(extractHeader(x$hdr, "ImageOrientationPatient", FALSE), 6)
is.axial(iop)
is.coronal(iop)
is.sagittal(iop)

x <- readDICOMFile(system.file("dcm/Spine1.dcm", package="oro.dicom"))
iop <- header2matrix(extractHeader(x$hdr, "ImageOrientationPatient", FALSE), 6)
is.axial(iop)
is.coronal(iop)
is.sagittal(iop)
```

`parsePixelData` *Parse DICOM Pixel or Spectroscopy Data*

Description

These subroutines process the information contained after the DICOM header and process this information into an image (2D or 3D) or complex-valued vector.

Usage

```
parsePixelData(rawString, hdr, endian = "little", flipupdown = TRUE)

parseSpectroscopyData(rawString, hdr, endian = "little")
```

Arguments

`rawString`	is a vector of `raw` values taken directly from the DICOM file.
`hdr`	is the list object of DICOM header information.
`endian`	is the endian-ness of the file (default is `"little"`).
`flipupdown`	is a logical variable for vertical flipping of the image (default is `TRUE`).

Details

A `while` loop is used to traverse the unknown number of DICOM header fields contained in a single file. Information contained in "sequences" may be included/excluded according to the logical variable `skipSequence` (default = `TRUE`).

A recursive implementation of the code breaks the DICOM file into segments and calls itself to parse each segment.

Value

A list containing two elements:

hdr all DICOM header fields (with or without "sequence" information).

img the 'image' information.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

Source

See references.

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org
http://en.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine

See Also

parseDICOMHeader, readDICOMFile.

---

readDICOM *Read All DICOM Files in a Directory*

Description

All DICOM files are imported and a text file summarizing their content recorded.

Usage

```
readDICOM(path, recursive = TRUE, exclude = NULL, verbose = FALSE,
  counter, ...)
```

Arguments

path	Path name to the DICOM directory.
recursive	Search recursively down from the given path name.
exclude	Exclude file names containing this character string.
verbose	Flag to provide text-based progress bar.
counter	Ignored.
...	Arguments to be passed to readDICOMFile.

Details

A `for` loop is used to process each DICOM file contained in the directory(ies). If only a single file is specified in the path, `readDICOM` will read that file only.

Value

A list structure with two major components:

`img`	All images associated with the DICOM directory(ies).
`hdr`	All header files associated with the DICOM directory(ies).

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Whitcher, B., V. J. Schmid and A. Thornton (2011). Working with the DICOM and NIfTI Data Standards in R, *Journal of Statistical Software*, **44** (6), 1–28. http://www.jstatsoft.org/v44/i06

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

`readDICOMFile`

Examples

```
## pixelData = TRUE
## The DICOM image data are read from readDICOM()

## Not run:
dcmSphere <- readDICOM(system.file("sphere3", package="oro.dicom"), verbose=TRUE)

## End(Not run)
```

---

readDICOMFile *Read Single DICOM File*

Description

All information, both header and image, is read into a list structure from a DICOM file.

Usage

```
readDICOMFile(fname, boffset = NULL, endian = "little",
  flipud = TRUE, skipSequence = FALSE, pixelData = TRUE, warn = -1,
  debug = FALSE)

parseDICOMHeader(rawString, sq.txt = "", endian = "little",
  verbose = FALSE)
```

Arguments

fname	is the file name of the DICOM image (with suffix).
boffset	is the number of bytes to skip at the beginning of the DICOM file (default = NULL which lets the code determine the starting point).
endian	is the endian-ness of the file (default is "little").
flipud	is a logical variable for vertical flipping of the image (default is TRUE).
skipSequence	is a logical variable to skip all content contained in SequenceItem tags (default = TRUE).
pixelData	is a logical variable (default = TRUE) on whether or not the PixelData should be read from the DICOM files. This is useful when one wants to gather the DICOM header information without loading the images.
warn	is a number to regulate the display of warnings (default = -1). See options for more details.
debug	is a logical variable (default = FALSE) that regulates to display of intermediate processing steps.
rawString	is a vector of raw values taken directly from the DICOM file.
sq.txt	is an character string (default = "") that indicates if the DICOM header field is embedded within a sequence.
verbose	is a logical variable (default = FALSE) that regulates to display of intermediate processing steps.

Details

A while loop is used to traverse the unknown number of DICOM header fields contained in a single file. Information contained in "sequences" may be included/excluded according to the logical variable skipSequence (default = TRUE).

A resursive implementation of the code breaks the DICOM file into segments and calls itself to parse each segment.

Strict adherence to the DICOM standard is not required. Specifically, content is allowed to start at the first byte and the four characters 'DICM' are not required at bytes 129-132.

Value

A list containing two elements:

hdr all DICOM header fields (with or without "sequence" information).

img the 'image' information.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Whitcher, B., V. J. Schmid and A. Thornton (2011). Working with the DICOM and NIfTI Data Standards in R, *Journal of Statistical Software*, **44** (6), 1–28. http://www.jstatsoft.org/v44/i06

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org
http://en.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine

See Also

readDICOM

Examples

```
x <- readDICOMFile(system.file("dcm/Abdo.dcm", package="oro.dicom"))
graphics::image(t(x$img), col=grey(0:64/64), axes=FALSE, xlab="", ylab="",
                main="Abdo.dcm")

x <- readDICOMFile(system.file("dcm/Spine1.dcm", package="oro.dicom"))
graphics::image(t(x$img), col=grey(0:64/64), axes=FALSE, xlab="", ylab="",
                main="Spine1.dcm")
```

---

str2time *Convert DICOM Time/Date Entry*

Description

The DICOM time entry (TM) is converted into two alternative formats: a text version of the original format and a number in seconds. The DICOM date entry (DA) is converted into a simple alternative format.

Usage

```
str2time(tt, format.out = "%02i:%02i:%08.5f")

str2date(dd, format.in = "%Y%m%d", format.out = "%d %b %Y")
```

Arguments

tt	TM field from a DICOM header.
dd	DA field from a DICOM header.
format.in, format.out	Appropriate formatting of input or output.

Details

DICOM "TM" format consists of a string of characters of the format `hhmmss.frac`; where `hh` contains hours (range "00" - "23"), `mm` contains minutes (range "00" - "59"), `ss` contains seconds (range "00" - "59"), and `frac` contains a fractional part of a second as small as 1 millionth of a second (range 000000 - 999999). A 24 hour clock is assumed. Midnight can be represented by only 0000 since 2400 would violate the hour range. The string may be padded with trailing spaces. Leading and embedded spaces are not allowed. One or more of the components `mm`, `ss`, or `frac` may be unspecified as long as every component to the right of an unspecified component is also unspecified. If `frac` is unspecified the preceding "." may not be included. `Frac` shall be held to six decimal places or less to ensure its format conforms to the ANSI HISPP MSDS Time common data type. Examples:

1. 070907.0705 represents a time of 7 hours, 9 minutes and 7.0705 seconds.

2. 1010 represents a time of 10 hours, and 10 minutes.

3. 021 is an invalid value.

Notes: For reasons of backward compatibility with versions of this standard prior to V3.0, it is recommended that implementations also support a string of characters of the format `hh:mm:ss.frac` for this VR.

DICOM "DA" format A string of characters of the format yyyymmdd; where yyyy shall contain year, mm shall contain the month, and dd shall contain the day. This conforms to the ANSI HISPP MSDS Date common data type. Example:

1. 19930822 would represent August 22, 1993.

Notes: For reasons of backward compatibility with versions of this standard prior to V3.0, it is recommended that implementations also support a string of characters of the format yyyy.mm.dd for this VR.

Value

For "TM", a list structure containing two fields

`txt`	A text version of the time where colons have been inserted for readability.
`time`	Time in seconds from midnight.

for "DA", a simple character string.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org
http://en.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine

See Also

readDICOM

Examples

```
str2date("19930822")
str2time("112308")
```

---

swapDimension *Reslice Data Volume Using DICOM Header Fields*

Description

The input data volume (assumed to be three-dimensional) is re-sliced so that each slice is in the axial plane. Orientation is preserved so that orthographic viewing is standardized.

Usage

```
swapDimension(img, dcm, digits = 2)
```

Arguments

img	Multidimensional array (assumed to be three-dimensional only).
dcm	DICOM header/image object associated with the multidimensional array.
digits	Number of significant digits used in testing unique-ness of values in DICOM header fields.

Value

Multidimensional array with (potentially) permuted dimensions because of the reslicing operation. An additional attribute "pixdim" is provided in order to facilitate conversion from DICOM to NIFTI/ANALYZE.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

See Also

dicom2nifti, getOrientation

`writeHeader` *Write DICOM Table to ASCII File*

Description

A wrapper to `write.table` specifically for DICOM tables.

Usage

```
writeHeader(dtable, filename, ...)
```

Arguments

`dtable`	The DICOM table.
`filename`	Name of the file to be created.
`...`	Additional parameters to be passed to `write.table`.

Details

This function is a straightforward wrapper to `write.table`.

Value

None.

Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

References

Digital Imaging and Communications in Medicine (DICOM)
http://medical.nema.org

See Also

`write.table`

Index