Package 'divest'

October 21, 2024

Version 1.1.1

Date 2024-10-20

Title Get Images Out of DICOM Format Quickly

Maintainer Jon Clayden <code@clayden.org>

Depends R (>= 3.5.0)

Imports Rcpp, RNifti

LinkingTo Rcpp, RNifti

Suggests jsonlite, tinytest, covr

Description Provides tools to sort DICOM-format medical image files, and convert them to NIfTI-1 format.

License BSD_3_clause + file LICENCE

URL https://github.com/jonclayden/divest

BugReports https://github.com/jonclayden/divest/issues

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Jon Clayden [aut, cre] (<https://orcid.org/0000-0002-6608-0619>),
Chris Rorden [aut],
Martin J Fiedler [cph],
Cong Xu [cph],
Pascal Gloor [cph]

Repository CRAN

Date/Publication 2024-10-20 22:40:02 UTC

Contents

divest.capabilities   *Report package capabilities*

Description

This function summarises the capabilities of the package as compiled for the current machine, analogously to the base function capabilities(). It identifies the support available for various input and output formats.

Usage

```
divest.capabilities(what = NULL)
```

Arguments

what   A character vector of components to extract, or NULL, the default, which indi-
                cates the full set of capabilities.

Value

A named logical vector, indicating whether plain JPEG, JPEG-LS and JPEG2000 DICOM transfer syntaxes are supported by the current build of the package, and also whether zlib is available to produce compressed NIfTI output files from convertDicom().

Author(s)

Jon Clayden code@clayden.org

See Also

readDicom()

Examples

```
divest.capabilities()
```

---

fromBidsJson *Conversion to and from BIDS JSON*

Description

Functions to convert to and from BIDS JSON format for image metadata. They are wrappers around functions from the jsonlite package, with the additional ability to convert between divest's tag naming convention and the BIDS equivalent. The differences are mostly in capitalisation, and the units used for magnetic resonance echo, repetition and inversion times.

Usage

```
fromBidsJson(source, rename = FALSE)

toBidsJson(source, path = NULL, rename = FALSE)
```

Arguments

source	A list containing metadata (see [imageAttributes](#)) or, for fromBidsJson, a string containing literal JSON or the path to a file containing it.
rename	Logical value. If TRUE, element names are also converted to or from the BIDS convention; otherwise this is just a conversion between an R list and a JSON string.
path	For toBidsJson, the path to write the JSON output to. If NULL, the default, the JSON text is returned in an object.

Value

fromBidsJson returns a list of image attributes. toBidsJson returns a character vector if path is NULL, otherwise nothing.

Author(s)

Jon Clayden code@clayden.org

References

More information about metadata captured by the BIDS format can be found at https://bids.neuroimaging.io or in the paper cited below.

K.J. Gorgolewski, T. Auer, V.D. Calhoun, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments (2016). Scientific Data 3:160044. doi:10.1038/sdata.2016.44.

---

imageAttributes   *Extended image attributes*

Description

These functions extract and replace medical image attributes that go beyond the core metadata associated with the NIfTI-1 file format.

Usage

```
imageAttributes(x)

imageAttributes(x) <- value
```

Arguments

x   An R object, generally an image object like those returned by readDicom.

value   A list of new image attributes to replace any existing ones.

Details

The DICOM format can encapsulate copious amounts of metadata about the scan and the patient, which can be useful for more advanced or research-focussed post-processing methods. Some of this information is extracted during the DICOM-to-NIfTI conversion process and stored in additional named attributes; the imageAttributes function returns a list of just these extended attributes. The replacement form allows this metadata to be modified or removed. These functions currently only act on objects inheriting from the niftiImage class.

Value

A list of image attributes, or a modified object with these changed. These are essentially all attributes except those used for basic niftiImage objects by the RNifti package.

Note

Attributes may include sensitive or identifiable information such as a patient's name, sex, date of birth, etc., if this was included in the original DICOM files. These functions make no attempt to identify or anonymise this metadata, and so this must be handled by the user if necessary.

Author(s)

Jon Clayden code@clayden.org

References

More information about metadata captured by the BIDS format can be found at `https://bids.neuroimaging.io` or in the paper cited below.

K.J. Gorgolewski, T. Auer, V.D. Calhoun, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments (2016). Scientific Data 3:160044. doi:10.1038/sdata.2016.44.

Examples

```
path <- system.file("extdata", "raw", package="divest")
images <- readDicom(path, interactive=FALSE)
imageAttributes(images[[1]])
```

---

readDicom   *Read one or more DICOM directories*

Description

These functions are R wrappers around the DICOM-to-NIfTI conversion routines provided by dcm2niix. They scan directories containing DICOM files, potentially pertaining to more than one image series, read them and/or merge them into a list of `niftiImage` objects.

Usage

```
readDicom(path = ".", subset = NULL, flipY = TRUE, crop = FALSE,
  forceStack = FALSE, verbosity = 0L, labelFormat = "T%t_N%n_S%s",
  depth = 5L, interactive = base::interactive(), output = NULL)

convertDicom(path = ".", subset = NULL, flipY = TRUE, crop = FALSE,
  forceStack = FALSE, verbosity = 0L, labelFormat = "T%t_N%n_S%s",
  depth = 5L, interactive = base::interactive(), output = path)

sortDicom(path = ".", forceStack = FALSE, verbosity = 0L,
  labelFormat = "T%t_N%n_S%s/%b", depth = 5L, nested = NA,
  keepUnsorted = FALSE, output = path)

scanDicom(path = ".", forceStack = FALSE, verbosity = 0L,
  labelFormat = "T%t_N%n_S%s", depth = 5L)
```

Arguments

path   A character vector of paths to scan for DICOM files. Each will examined in turn. The default is the current working directory. `readDicom` (only) will accept paths to individual DICOM files, rather than directories. Alternatively, for `readDicom` and `sortDicom`, a data frame like the one returned by `scanDicom`, from which file paths will be read.

subset	If path is a data frame, an expression which will be evaluated in the context of the data frame to determine which series to convert. Should evaluate to a logical vector. If path is a character vector, scanDicom is called on the path(s) first to produce the data frame. If this is specified, and does not evaluate to NULL, the read will be noninteractive, irrespective of the value of the interactive argument.
flipY	If TRUE, the default, then images will be flipped in the Y-axis. This is usually desirable, given the difference between orientation conventions in the DICOM and NIfTI-1 formats.
crop	If TRUE, then dcm2niix will attempt to crop excess neck slices from brain images.
forceStack	If TRUE, images with the same series number will always be stacked together as long as their dimensions are compatible. If FALSE, the default, images will be separated if they differ in echo, coil or exposure number, echo time, protocol name or orientation.
verbosity	Integer value between -2 and 3, controlling the amount of output generated during the conversion. A value of -1 will suppress all output from dcm2niix except warnings and errors; -2 also suppresses warnings.
labelFormat	A [sprintf](#)-style string specifying the format to use for the final image labels or paths. See Details.
depth	The maximum subdirectory depth in which to search for DICOM files, relative to each path.
interactive	If TRUE, the default in interactive sessions, the requested paths will first be scanned and a list of DICOM series will be presented. You may then choose which series to convert.
output	The directory to write converted or copied NIfTI files to, or NULL. In the latter case, which isn't valid for sortDicom, images are converted in memory and returned as R objects.
nested	For sortDicom, should the sorted files be created within the source directory (TRUE), or in the current working directory (FALSE)? Now soft-deprecated in favour of output, which is more flexible.
keepUnsorted	For sortDicom, should the unsorted files be left in place, or removed after they are copied into their new locations? The default, FALSE, corresponds to a move rather than a copy. If creating new files fails then the old ones will not be deleted.

Details

The scanDicom function parses directories full of DICOM files and returns information about the acquisition series they contain. readDicom reads these files and converts them to (internal) NIfTI images (whose pixel data can be extracted using as.array). convertDicom performs the same conversion but writes to NIfTI files by default, instead of retaining the images in memory. sortDicom renames the files, but does not convert them.

The labelFormat argument describes the string format used for image labels and sorted files. Valid codes, each escaped with a percentage sign, include a for coil number, b for the source file base name, c for image comments, d for series description, e for echo number, f for the source directory,

i for patient ID, j for the series instance UID, k for the study instance UID, l for the procedure step description, m for manufacturer, n for patient name, p for protocol name, q for scanning sequence, r for instance number, s for series number, t for the date and time, u for acquisition number, v for vendor, x for study ID and z for sequence name. For `sortDicom` the label forms the new file path, and may include one or more slashes to create subdirectories. A ".dcm" suffix will be added to file names if no extension is specified.

Value

`readDicom` and `convertDicom` return a list of `niftiImage` objects if `output` is `NULL`; otherwise (invisibly) a vector of paths to NIfTI-1 files created in the target directory. The `scanDicom` function returns a data frame containing information about each DICOM series found. `sortDicom` is mostly called for its side-effect, but also (invisibly) returns a list detailing source and target paths.

Author(s)

Jon Clayden code@clayden.org

Examples

```
path <- system.file("extdata", "raw", package="divest")
scanDicom(path)
readDicom(path, interactive=FALSE)
```

Index