Package 'tractor.base'

July 2, 2014

Version 2.5.0

Date 2014-03-17

Title A package for reading, manipulating and visualising magnetic resonance images

Author Jon Clayden

Maintainer Jon Clayden <code@clayden.org>

Depends R (>= 2.12.1), graphics, grDevices, methods, stats, utils

Imports reportr

Enhances oro.nifti

Description The tractor.base package consists of functions for working with magnetic resonance images. It can read and write image files stored in Analyze, NIfTI, MGH and DICOM formats (DICOM support is read only), generate images for use as regions of interest, and manipulate and visualise images.

Encoding UTF-8

LazyLoad yes

License GPL-2

URL http://www.tractor-mri.org.uk

NeedsCompilation no

Repository CRAN

Date/Publication 2014-03-18 16:43:20

R topics documented:

`DicomMetadata-class` *Class* `"DicomMetadata"`

Description

This class represents DICOM metadata, which typically contains detailed information about the scan parameters and subject.

Extends

Class `"SerialisableObject"`, directly.

All reference classes extend and inherit methods from `"envRefClass"`.

Fields

`source`: Object of class `character`.

`tags`: Object of class `data.frame`.

`tagOffset`: Object of class `integer`.

`dataOffset`: Object of class `integer`.

`dataLength`: Object of class `integer`.

`explicitTypes`: Object of class `logical`.

`endian`: Object of class `character`.

Class-Based Methods

`getTags()`: Retrieve the data frame containing all available tags, which has columns `"groups"`, `"elements"`, `"types"` and `"values"`. See the DICOM standard for more information.

`getTagValue(group, element)`: Retrieve the value of the tag with specified group and element numbers, usually given in hex.

`getTagOffset()`: Retrieve the byte offset of useful tags in the file.

`getDataOffset()`: Retrieve the byte offset of the data in the DICOM file.

`getDataLength()`: Retrieve the length of the data part of the DICOM file in bytes.

`getSource()`: Retrieve the name of the source file.

`getEndianness()`: Retrieve the endianness of the file: `"big"` or `"little"`.

`nTags()`: Retrieve the number of tags stored in this object.

The following methods are inherited (from the corresponding class): serialise ("SerialisableObject"), methods ("SerialisableObject")

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

Examples

```
showClass("DicomMetadata")
```

dictionary *DICOM tag dictionary*

Description

This data set provides information about DICOM tags, including descriptions and data types. It is primarily for internal use.

Usage

```
dictionary
```

Format

A data frame, with rows corresponding to known tags.

emptyMatrix *The empty matrix*

Description

The empty matrix is a standard matrix of dimensions 0 x 0. It is intended to be used as a placeholder where a matrix is required but no information is stored.

Usage

```
emptyMatrix()
is.emptyMatrix(object)
```

Arguments

`object`	Any object.

Value

`emptyMatrix` returns the empty matrix, equivalent to `matrix(NA,0,0)`. `is.emptyMatrix` returns `TRUE` if its argument is identical to the empty matrix.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

`equivalent` *Test two numeric vectors for equivalence*

Description

Test two numeric vectors for equivalence.

Usage

```
equivalent(x, y, signMatters = TRUE, ...)
```

Arguments

`x`	The first numeric vector.
`y`	The second numeric vector.
`signMatters`	Logical value: if FALSE then equivalence in absolute value is sufficient.
`...`	Additional arguments to `all.equal`, notably `tolerance`.

Details

This function is a wrapper for `isTRUE(all.equal(x,y,...))`, but with the additional capability of doing sign-insensitive comparison.

Value

`TRUE` if all elements of `x` match all elements of `y` to within tolerance, ignoring signs if required. `FALSE` otherwise.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

`all.equal`

Examples

```
equivalent(c(-1,1), c(1,1))  # FALSE
equivalent(c(-1,1), c(1,1), signMatters=FALSE)  # TRUE
equivalent(1:2, 2:3, tolerance=2)  # TRUE
```

execute *Find or run an external executable file*

Description

The execute function is a wrapper around the system function in base R, which echoes the command being run (including the full path to the executable) if the reportr output level is Debug. locateExecutable simply returns the path to an executable file on the system PATH.

Usage

```
execute(executable, paramString = NULL, errorOnFail = TRUE, silent = FALSE, ...)
locateExecutable(fileName, errorIfMissing = TRUE)
```

Arguments

executable, fileName	Name of the executable to run.
paramString	A character string giving the parameters to pass to the executable, if any.
errorOnFail, errorIfMissing	Logical value: should an error be produced if the executable can't be found?
silent	Logical value: should the executable be run without any output?
...	Additional arguments to system.

Value

For execute, the return value of the underlying call to system. For locateExecutable, the location of the requested executable, or NULL if it could not be found.

Note

These functions are designed for Unix systems and may not work on Windows.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

system

getColourScale *Functions for working with colour scales or palettes*

Description

The getColourScale function can be used to obtain a standard or customised colour scale for use in the package's image visualisation functions. A graded palette of colours between two or more key colours can be obtained using interpolatePalette.

Usage

```
getColourScale(n)

interpolatePalette(colours, n, ...)
```

Arguments

n	For getColourScale, a number, colour name or list (see Details). For interpolatePalette, a single integer specifying the length of the interpolated palette.
colours	A vector of colours to interpolate between, using any format recognised by colours.
...	Additional arguments to colorRamp.

Details

Colour scales can be specified in any of three ways. Firstly, by a single number, representing a predefined colour scale. Currently valid values are 1 (greyscale, black background), 2 (red to yellow heat scale, red background), 3 (blue to red rainbow scale, blue background), 4 (blue to white to red diverging scale, white background), 5 (white to red, white background) and 6 (white to blue, white background). Secondly, a single colour name can be given (see colours); in this case the background will be black. This is useful for binary images. Thirdly and most flexibly, a list with two named elements can be given: colours, a vector of colours representing the colour scale, perhaps created using rgb; and background, a single colour representing the background.

Value

For getColourScale, a list with elements

colours	A character-mode vector representing the colours in the scale, usually of length 100. This can be passed as a colour scale to R's plotting functions.
background	A single character string representing the background colour.

This can be passed as an argument to the package's image visualisation functions.

The interpolatePalette function returns a character-mode vector representing the colours in the interpolated scale.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

colours, rgb, colorRamp

Examples

```
getColourScale(1)

interpolatePalette(c("red","yellow"), 10)
```

---

implode *Create a character string by concatenating the elements of a vector*

Description

Create a character string by concatenating the elements of a vector, using a separator and optional final separator.

Usage

```
implode(strings, sep = "", finalSep = NULL, ranges = FALSE)
```

Arguments

strings	A vector, which will be coerced to mode character.
sep	A unit length character vector giving the separator to insert between elements.
finalSep	An optional unit length character vector giving the separator to insert between the final two elements.
ranges	Logical value. If TRUE and strings can be interpreted as integers, collapse runs of consecutive numbers into range notation.

Value

A character vector of length one.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

paste

Examples

```
implode(1:3, ", ")  # "1, 2, 3"
implode(1:3, ", ", " and ")  # "1, 2 and 3"
implode(1:2, ", ", " and ")  # "1 and 2"
implode(1:3, ", ", ranges=TRUE)  # "1-3"
```

MriImage-class *Class* "MriImage"

Description

This class represents an MRI image. An object of this class is made up of some voxel data, stored as a sparse or dense numeric array, and some metadata, such as the file it was read from, the voxel dimensions, and so on. The group generic functions Math, Ops and Summary are defined for this class, as are methods for coercing to and from a standard R array.

Extends

Class "SerialisableObject", directly.

All reference classes extend and inherit methods from "envRefClass".

Methods

[signature(x = "MriImage", i = "ANY", j = "ANY"): Index methods for the MriImage class. These are wrappers around the corresponding method for the array or SparseArray classes, depending on the underlying data type.

[signature(x = "MriImage", i = "ANY", j = "missing")

[signature(x = "MriImage", i = "missing", j = "ANY")

[signature(x = "MriImage", i = "missing", j = "missing")

`[<- signature(x = "MriImage", i = "ANY", j = "ANY")`: Replacement methods for the `MriImage` class. These are wrappers around the corresponding method for the `array` or `SparseArray` classes, depending on the underlying data type.

`[<- signature(x = "MriImage", i = "ANY", j = "missing")`

`[<- signature(x = "MriImage", i = "missing", j = "ANY")`

`[<- signature(x = "MriImage", i = "missing", j = "missing")`

Fields

`imageDims`: Object of class `integer`.

`voxelDims`: Object of class `numeric`.

`voxelDimUnits`: Object of class `character`.

`source`: Object of class `character`.

`origin`: Object of class `numeric`.

`storedXform`: Object of class `matrix`.

`reordered`: Object of class `logical`.

`tags`: Object of class `list`.

`data`: Object of class `MriImageData`.

Class-Based Methods

`initialize(imageDims=NULL, voxelDims=NULL, voxelDimUnits=NULL, source="", origin=NULL, storedXform=`: Create a new object of this class.

`summarise()`: Retrieve information about this object. This method is usually only called implicitly by the `"show"` method.

`apply(...)`: Apply a function to margins of the image, as with the `apply` function in the base package.

`getData()`: Retrieve the array of voxel values.

`getDataAtPoint(...)`: Retrieve the value of the voxel at the location specified by `c(...)`. Returns `NA` if the location is out of bounds.

`getDimensionality()`: Get the dimensionality of the image.

`getDimensions()`: Get the number of pixels or voxels in each dimension.

`getFieldOfView()`: Get the field of view.

`getMetadata()`: Retrieve a version of the object with the `data` field set to `NULL`.

`getNonzeroIndices(array=TRUE, positiveOnly=FALSE)`: Get the locations of nonzero voxels in the image.

`getOrigin()`: Get the coordinates of the image origin, in voxels.

`getSource()`: Get the source file name associated with the image.

`getSparseness()`: Retrieve the proportion of image pixels or voxels which are nonzero.

`getStoredXformMatrix()`: Get the xform matrix that was read with the image, if any. This will be the empty matrix (see `emptyMatrix`) if none is stored.

`getTag(key)`: Get the tag value associated with the specified key.

`getTags()`: Get the list of tags associated with the image.

`getVoxelDimensions()`: Get the size of the voxels in each dimension.

`getVoxelUnits()`: Get the spatial and temporal units relating to the voxel dimensions.

`setOrigin(newOrigin)`: Set the image origin.

`setSource(newSource)`: Set the source file associated with the image.

`isEmpty()`: `TRUE` if no data is stored with the image.

`isInternal()`: `TRUE` if no source file is associated with the image.

`isReordered()`: `TRUE` if the image data has been reordered to LAS.

`isSparse()`: `TRUE` if the image data is stored as a `SparseArray` object, `NA` if there is no data stored with the image, and `FALSE` otherwise.

The following methods are inherited (from the corresponding class): fields ("SerialisableObject"), serialise ("SerialisableObject"), methods ("SerialisableObject")

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

The `SerialisableObject` class, which this class extends. Also the group generic functions `Math`, `Ops` and `Summary` (in the `methods` package); `newMriImageWithSimpleFunction` and `newMriImageWithBinaryFunction`.

Examples

```
showClass("MriImage")
```

---

neighbourhoodInfo *Class representing a cuboidal neighbourhood in an image*

Description

This class represents a cuboidal region of an image, with a centre and a fixed voxel width.

Usage

```
createNeighbourhoodInfo(width, dim = 3, centre = rep(0, dim))
```

Arguments

width	An integer voxel width. Must be odd.
dim	An integer giving the dimensionality of the neighbourhood. Currently must be 3.
centre	A numeric vector giving the centre voxel of the neighbourhood. Must have exactly dim elements.

Value

createNeighbourhoodInfo returns a list with class "neighbourhoodInfo" and elements

width	Copied from the width argument.
dim	Copied from the dim argument.
centre	Copied from the centre argument.
vectors	dim x width^dim matrix whose columns give the locations of each point in the neighbourhood.
innerProducts	A square, symmetric matrix of inner products between every location in the neighbourhood and every other.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

---

newDicomMetadataFromFile

Read a DICOM file into a DicomMetadata object

Description

This function reads a DICOM file into a DicomMetadata object. Only DICOM files from magnetic resonance scanners are supported.

Usage

```
newDicomMetadataFromFile(fileName, checkFormat = TRUE, dictionary = NULL,
    stopTag = NULL, ignoreTransferSyntax = FALSE)
```

Arguments

fileName	The name of a DICOM file.
checkFormat	If TRUE, the function will check for the magic string "DICM" at byte offset 128. This string should be present, but in reality not all files contain it.
dictionary	A tag dictionary to use when reading the file. If NULL then the built-in dictionary will be loaded and used.
stopTag	An integer vector giving the group and element numbers (in that order) of a DICOM tag, or NULL. If not NULL, the function will stop parsing the DICOM file if the specified tag is encountered. This can be used to speed up the process if a specific tag is required.
ignoreTransferSyntax	If TRUE, any transfer syntax stored in the file will be ignored, and the code will try to deduce the transfer syntax using heuristics. This may occasionally be necessary for awkward DICOM files, but is not generally recommended.

Value

newDicomMetadataFromFile returns a DicomMetadata object, or NULL on failure.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

The DICOM standard, found online at http://dicom.nema.org/. (Warning: may produce headaches!) Also dictionary, and newMriImageFromDicomDirectory for information on how to create MriImage objects from DICOM files.

---

newMriImageFromDicomDirectory

Functions for reading images from DICOM files

Description

Functions for reading images from a directory of DICOM files.

Usage

```
readDicomDirectory(dicomDir, readDiffusionParams = FALSE, untileMosaics = TRUE)
newMriImageFromDicomDirectory(dicomDir, readDiffusionParams = FALSE,
    untileMosaics = TRUE)
```

Arguments

dicomDir	Character vector of length one giving the name of a directory containing DICOM files.
readDiffusionParams	Logical value: should diffusion MRI parameters (b-values and gradient directions) be retrieved from the files if possible?
untileMosaics	Logical value: should Siemens mosaic images be converted into 3D volumes? This may occasionally be performed in error, which can be prevented by setting this value to FALSE.

Value

A list containing elements

image	An MriImage object.
bValues	Diffusion b-values, if requested. Will be NA if the information could not be found in the files.
bVectors	Diffusion gradient vectors, if requested. Will be NA if the information could not be found in the files.

Note

These two functions are equivalent, but readDicomDirectory is preferred for brevity. The newMriImageFromDicomDirecto function is likely to be deprecated in a future release.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

DicomMetadata, MriImage, sortDicomDirectory.

newMriImageFromFile *Working with MRI images stored in NIfTI, Analyze and MGH formats*

Description

Functions for reading, writing, locating, copying and removing MRI images stored in NIfTI, Analyze and MGH formats.

Usage

```
readImageFile(fileName, fileType = NULL, metadataOnly = FALSE, volumes = NULL,
    sparse = FALSE, mask = NULL, reorder = TRUE)
writeImageFile(image, fileName = NULL, fileType = NA, overwrite = TRUE)

newMriImageFromFile(fileName, fileType = NULL, metadataOnly = FALSE,
    volumes = NULL, sparse = FALSE, mask = NULL, reorder = TRUE)
writeMriImageToFile(image, fileName = NULL, fileType = NA, overwrite = TRUE)

identifyImageFileNames(fileName, fileType = NULL, errorIfMissing = TRUE)
imageFileExists(fileName, fileType = NULL)
removeImageFilesWithName(fileName)
copyImageFiles(from, to, overwrite = FALSE, deleteOriginals = FALSE)
symlinkImageFiles(from, to, overwrite = FALSE, relative = TRUE)
```

Arguments

`fileName, from, to`	File names, with or without appropriate extension.
`image`	An `MriImage` object.
`fileType`	A character vector of length one, giving the file type required or expected. If this option is missing, the file type used for writing images will be taken from the `tractorFileType` option. See Details.
`metadataOnly`	Logical value: if `TRUE`, only metadata are read into the object.
`volumes`	An optional integer vector specifying a subset of volumes to read (generally to save memory). If given, only the requested volumes in the 4D file will be read.
`sparse`	Logical value: should the image data be stored in a `SparseArray` object?
`mask`	An optional `MriImage` object representing a mask, outside of which the image to be read should be considered to be zero. This can be used to save memory when only a small part of a large image is of interest. Ignored if `sparse` is not `TRUE`.
`reorder`	Logical value: should the image data be reordered to LAS? This is recommended in most circumstances.
`overwrite`	Logical value: overwrite an existing image file? For `writeMriImageToFile`, an error will be raised if there is an existing file and this is set to FALSE.

errorIfMissing Logical value: raise an error if no suitable files were found?

deleteOriginals
Logical value: if TRUE, copyImageFiles performs a move rather than a copy.

relative Logical value: if TRUE, the path stored in the symlink will be relative (e.g. "../some_dir/some_image.nii") rather than absolute (e.g. "/path/to/some_dir/some_image.nii")

Details

NIfTI and Analyze are related formats for storing magnetic resonance images. NIfTI is a more recent extension of Analyze, and contains more specific information about, for example, the orientation of the image. Its use is therefore recommended where possible. MGH format is used by the popular image processing package FreeSurfer. These formats use a number of different file extensions, but the details are abstracted away from the user by these functions.

TractoR does not allow for files with the same basic name using multiple Analyze/NIfTI/MGH formats in a single directory (e.g. "foo.nii" AND "foo.img"), and these functions will produce an error if multiple compatible files exist.

Suitable values for fileType (and the tractorFileType option, which is used as a default) are ANALYZE, NIFTI, NIFTI_PAIR (the two-file NIfTI format), MGH, ANALYZE_GZ, NIFTI_GZ, NIFTI_PAIR_GZ and MGH_GZ. The latter four are gzipped versions of the former four. NIFTI_GZ is recommended unless there is a need for one of the others. This is the default value for the tractorFileType option, but that can be changed using a call to options, or by setting the TRACTOR_FILETYPE environment variable before loading the tractor.base package.

Since multiple files may be involved, copying, moving or symlinking images is not trivial. copyImageFiles and symlinkImageFiles are wrappers around the standard functions file.copy and file.symlink which handle this complexity.

Value

readImageFile returns an MriImage object. imageFileExists returns TRUE if an existing file with the specified name exists (all file extensions are checked), and FALSE otherwise. removeImageFilesWithName returns the result of unlink applied to all relevant files. writeImageFile and identifyImageFileNames return a list with the following elements, describing the identified or written files:

fileStem The file name without extension.

headerFile The full header file name.

imageFile The full image file name.

format The format of the files ("Nifti", "Analyze" or "Mgh"). Not returned by writeMriImageToFile.

copyImageFiles and symlinkImageFiles are called for their side effects.

Note

The functions readImageFile and writeImageFile are equivalent to newMriImageFromFile and writeMriImageToFile, but preferred for brevity. The latter are likely to be deprecated in a future release.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

The NIfTI-1 standard (http://nifti.nimh.nih.gov/nifti-1) and MriImage.

---

newMriImageWithData *Functions for creating MriImage objects from data*

Description

Functions for creating MriImage objects from data, including other images.

Usage

```
newMriImageWithData(data, templateImage = nilObject(), imageDims = NA,
    voxelDims = NA, voxelDimUnits = NA, origin = NA, tags = NA)

newMriImageByExtraction(image, dim, loc)
extractDataFromMriImage(image, dim, loc)
newMriImageByMasking(image, mask)
newMriImageByThresholding(image, level, defaultValue = 0)
newMriImageByTrimming(image, clearance = 4)
newMriImageByReordering(image)

newMriImageAsShapeOverlay(type = c("cross", "block"), baseImage, ...)
generateImageDataForShape(type = c("cross", "block"), dim, background = 0,
    centre = NA, width = NA)

newMriImageWithSimpleFunction(image, fun, ...)
newMriImageWithBinaryFunction(image1, image2, fun, ...)
```

Arguments

data	An array of voxel data.
templateImage	An optional MriImage object, to be used as a metadata template.
imageDims, voxelDims, voxelDimUnits, origin, tags	Metadata for the new image object. These values override any from the metadata object or data array.

image, image1, image2	MriImage objects.
dim, loc	For newMriImageByExtraction, the dimension and location along that dimension for which data should be extracted. For generateImageDataForShape, dim is the dimensions of the image. newMriImageAsShapeOverlay takes this from the baseImage.
mask	An array of mode logical indicating which voxels are in the mask. Must have the same dimensions as the image.
level	A numeric value specifying the threshold level.
defaultValue	The value of the final image in voxels which are below threshold.
clearance	The number of voxels' clearance left around a trimmed image.
type	The shape type to generate. A "block" is a cubic region of the image; a "cross" is the central line of the cube in each dimension.
baseImage	The MriImage to use as a base for the overlay.
background	The voxel value outside the shape.
centre, width	The centre and width of the shape.
fun	A function object, taking one or two numeric array parameters, as appropriate.
...	For newMriImageAsShapeOverlay, further parameters to generateImageDataForShape. And for newMriImageWithSimpleFunction and newMriImageWithBinaryFunction, further parameters to fun.

Details

All of these functions use data from arrays or MriImage objects to create a new MriImage object. newMriImageWithData is the basic function for creating an object from its constituents: an array of voxel values and some metadata (and/or a template image).

newMriImageByExtraction reduces the dimensionality of the source image by one, by extracting a single "line" of data along one dimension. (An array, rather than an MriImage object, is returned by extractDataFromMriImage.) newMriImageByMasking modifies the data by masking out unwanted voxels, and newMriImageByThresholding by thresholding. newMriImageByTrimming trims empty space from the edges of an image, reducing the dimensions of the image and thus avoiding the storage of lots of zeroes. newMriImageByReordering reorders the image data (and corresponding metadata) to the LAS convention, an operation which is usually performed when an image is read from file. newMriImageAsShapeOverlay creates an image which contains a simple shape. newMriImageWithSimpleFunction and newMriImageWithBinaryFunction modify the image data by applying an arbitrary function to it. Any function that can be applied to numeric arrays, and expects one or two arguments, respectively, is suitable for fun.

Value

An MriImage object.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

MriImage

`newMriImageWithDataRepresentation` *Convert between image data storage conventions*

Description

This function returns a version of its first argument using the specified data representation.

Usage

```
newMriImageWithDataRepresentation(image, representation = c("dense","coordlist"))
```

Arguments

`image`	An `MriImage` object.
`representation`	A character string specifying the required data representation.

Value

A version of `image` with using the specified data representation, i.e. an `array` (with `representation="dense"`) or a `SparseArray` (with `representation="coordlist"`).

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

MriImage, SparseArray, array

newSparseArrayWithData *Create a SparseArray object*

Description

This function creates a SparseArray object from its constituent parts.

Usage

```
newSparseArrayWithData(data, coordinates, dims)
```

Arguments

data	A vector of (nonzero) array elements.
coordinates	A matrix with as many rows as data has elements, containing the coordinates of each nonzero element in the array.
dims	The dimensions of the array.

Value

A SparseArray object.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

nilObject *The nil object*

Description

The nil object is an empty object of class SerialisableObject. It can be used as a placeholder where such an object of this class, or one of its subclasses, is required. It serialises to the empty list.

Usage

```
nilObject()
is.nilObject(object)
```

Arguments

object	Any object.

Value

nilObject returns the nil object. is.nilObject returns TRUE if its argument is identical to the nil object, or if it is equivalent in the sense of serialising to an identical result.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

SerialisableObject

---

path manipulation *Functions for file name and path manipulation*

Description

Functions for expanding file paths, finding relative paths and ensuring that a file name has the required suffix.

Usage

```
ensureFileSuffix(fileName, suffix, strip = NULL)
expandFileName(fileName, base = getwd())
relativePath(path, referencePath)
```

Arguments

fileName	A character vector of file names.
suffix	A character vector of file suffixes, which will be recycled if shorter than fileName.
strip	A character vector of suffixes to remove before appending suffix. The intended suffix does not need to be given here, as the function will not append it if the specified file name already has the correct suffix.
base	If fileName is a relative path, this option gives the base directory which the path is relative to. If fileName is an absolute path, this argument is ignored.
path, referencePath	Character strings representing file paths.

Value

The ensureFileSuffix function returns the specified file names with the requested suffixes appended. expandFileName returns the full path to the specified file name, collapsing ".." elements if appropriate. relativePath returns the specified path, expressed relative to referencePath.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

path.expand performs some of what expandFileName does.

---

printLabelledValues *Pretty print labelled information*

Description

This is a simple function to print a series of labels and associated data values, or key-value pairs.

Usage

```
printLabelledValues(labels, values, outputLevel = OL$Info, leftJustify = FALSE)
```

Arguments

labels	A character vector of labels.
values	A character vector of values. Must have the same length as labels.
outputLevel	The output level to print the output to. See setOutputLevel, in the reportr package.
leftJustify	Logical value: if TRUE the labels will be left justified; otherwise they will be right justified.

Value

This function is called for its side effect.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

setOutputLevel for the reportr output level system.

---

promote *Promote a vector to a single-column or single-row matrix*

Description

The promote function promotes a vector argument to a single-column or single-row matrix. Matrix arguments are returned unmodified.

Usage

```
promote(x, byrow = FALSE)
```

Arguments

x	A vector or matrix.
byrow	Logical value: if TRUE, a vector will be promoted to a single-row matrix; otherwise a single-column matrix will result.

Value

A matrix version of the x argument.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

matrix

SerialisableObject-class
Class "SerialisableObject"

Description

This reference class extends the standard "envRefClass" class, adding a function for simple serialisation of the data fields of an object, and one for finding all of the methods available for an object. A serialised object may be deserialised using the deserialiseReferenceObject function.

Extends

All reference classes extend and inherit methods from "envRefClass".

Class-Based Methods

serialise(file): If the file argument is missing, serialises the object to a list containing the (named) fields of the object, with its "originalClass" attribute set to the class name of the original object. If the file argument is supplied, this object is saved to the corresponding file name using the standard save function.

fields(): Retrieve a character vector describing the fields in the object.

methods(): Retrieve a character vector describing the methods available to the object.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

Examples

```
showClass("SerialisableObject")
```

serialisation *Reference object serialisation and deserialisation*

Description

Rather than using R's save and load functions directly for reference objects, TractoR uses the SerialisableObject class and these functions to save and load objects. The main difference is that this approach stores only the data in the object, and not the functions which operate on them. This helps backward compatibility when new member functions are added.

Usage

```
serialiseReferenceObject(object, file = NULL)

isDeserialisable(object, expectedClass = NULL)
deserialiseReferenceObject(file = NULL, object = NULL, raw = FALSE)
```

Arguments

object	For serialiseReferenceObject, a list or object inheriting from SerialisableObject. For other functions, an object in (raw) serialised form. See Details.
expectedClass	A class name which the object is expected to inherit. Any class is acceptable if this parameter is NULL.
file	A file name to deserialise from.
raw	If TRUE, the raw serialised object is returned; otherwise the object is converted back to its original class.

Details

The serialiseReferenceObject function, or the serialise member function of the SerialisableObject class can be used to create and/or save a version of an object which contains a hierarchical representation of the data embedded in it. These serialised objects are standard R lists, with an "originalClass" attribute describing the class of the original object. The deserialiseReferenceObject function can be used to deserialise them.

Note that this should generally NOT be used as the primary mechanism for saving and loading MriImage objects. Saving to standard NIfTI/Analyze format is usually preferable, and can be done using writeImageFile.

Value

isDeserialisable returns TRUE if the object is deserialisable and inherits from the specified class. deserialiseReferenceObject returns a raw or reconstituted object after deserialisation.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

SerialisableObject, save, load, writeImageFile.

---

sortDicomDirectory *Sort a directory of DICOM files into series*

Description

This function sorts a directory containing DICOM files into subdirectories by series number (DICOM tag 0x0020,0x0011) series time (0x0008,0x0031), subject name (0x0010,0x0010) and/or scan date (0x0008,0x0020). Each unique identifier, together with its description, will be used as the name for a new subdirectory of the specified top-level directory, and all relevant files will be copied into that subdirectory. Duplicate file names are disambiguated if necessary.

Usage

```
sortDicomDirectory(directory, deleteOriginals = FALSE, sortOn = "series",
    useSeriesTime = FALSE)
```

Arguments

directory	A length-1 character vector giving the directory to search for DICOM files. Subdirectories will also be searched.
deleteOriginals	A single logical value. If TRUE, then the source files will be deleted after being copied to their new locations, making the operation a move rather than a copy. Nothing will be deleted if the copy fails.
sortOn	The string "series", "subject" or "date", or any combination in the order desired. This will be the basis of the sort, which will be nested if more than one type is specified.
useSeriesTime	Logical value. If TRUE, use the series time rather than number as the identifier. This can be more reliable if the files are from multiple scan sessions.

Value

This function is called for its side effect.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

newMriImageFromDicomDirectory for reading DICOM files into an MriImage object.

SparseArray-class *Class* "SparseArray"

Description

This class represents an array with any number of dimensions, in which a significant proportion of entries are zero. The coordinates of nonzero entries are stored along with their values, with all remaining entries assumed to be zero. Methods are provided to index into the array in the standard way, using matrix or vector indices; and for coercing between SparseArray objects and standard (dense) arrays.

Extends

Class "SerialisableObject", directly.

All reference classes extend and inherit methods from "envRefClass".

Methods

[signature(x = "SparseArray", i = "ANY", j = "ANY"): Index method for the SparseArray class.

[<- signature(x = "SparseArray", i = "ANY", j = "ANY"): Replacement method for the SparseArray class.

Fields

data: Object of class ANY.

coords: Object of class matrix.

dims: Object of class integer.

Class-Based Methods

`initialize(...)`: Create a new object of this class.

`aperm(perm)`: Permute dimensions of the array.

`apply(margin, fun, ...)`: Apply a function to the margins of the array. This function should be used in preference to `apply` from the base package, since the latter will convert the `SparseArray` to a dense array first, with a potentially major memory cost.

`flip(dimsToFlip)`: Flip the image along the specified dimension(s).

`getDimensions()`: Retrieve an integer vector giving the dimensions of the array.

`getDimensionality()`: Retrieve an integer vector of length 1 giving the number of dimensions in the array.

`getCoordinates()`: Retrieve the matrix of coordinates of nonzero entries, with each row corresponding to a nonzero entry.

`getData()`: Retrieve the vector of nonzero data values.

`setCoordinatesAndData(newCoords, newData)`: Update the coordinate matrix and data vector explicitly.

`setDimensions(newDims)`: Change the dimensions of the array.

The following methods are inherited (from the corresponding class): serialise ("SerialisableObject"), methods ("SerialisableObject")

Note

Using the current "coordinate list" storage convention, there is rarely any data size benefit to storing an array in this form unless it is at least 75% sparse. Since `SparseArray` is a reference class, however, there may be some benefit in terms of avoiding duplication of the array on modification.

Indexing with logical or character vectors is currently not supported.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

`array`. The implementation of indexing owes lot to the `slam` package.

Examples

```
showClass("SparseArray")
```

`threadSafeTempFile` *Obtain thread-safe temporary file names*

Description

This function is a wrapper around `tempfile`, which creates temporary file names whose path contains the process ID of the calling process. This avoids clashes between threads created by functions such as `mclapply` (in the "parallel" package), which can easily occur with the standard `tempfile` function.

Usage

```
threadSafeTempFile(pattern = "file")
```

Arguments

`pattern`	Character vector giving the initial part of each file name.

Value

A character vector of temporary file names. No files are actually created.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

`tempfile`

`vector functions` *Miscellaneous vector functions*

Description

These functions provide the (Euclidean) length of a vector, the vector cross product or angle between two vectors.

Usage

```
vectorLength(vector)
vectorCrossProduct(a, b)
angleBetweenVectors(v1, v2)
resolveVector(len, ...)
```

Arguments

`vector, v1, v2`	Numeric vectors of any length.
`a, b`	Numeric 3-vectors.
`len`	The expected length of the vector.
`...`	Elements of the vector, to be concatenated together.

Value

For `vectorLength`, the Euclidean norm or length of the specified vector, given by `sqrt(sum(vector^2))`. For `vectorCrossProduct`, the vector cross product of the two specified vectors; and for `angleBetweenVectors`, the angle (in radians) between the two specified vectors.

The `resolveVector` function concatenates the values given in `...`, and if the result is a vector of length `len` then it is returned. If not, `NULL` is returned.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

`crossprod` for the matrix cross product.

viewImages *A simple interactive viewer for MriImage objects*

Description

The viewImages function provides a simple interactive viewer for MriImage objects. 3D and 4D images may be used.

Usage

```
viewImages(images, colourScales = NULL, point = NULL, interactive = TRUE,
    crosshairs = TRUE, orientationLabels = TRUE, infoPanel = defaultInfoPanel,
    ...)

defaultInfoPanel(point, data, imageNames)
```

Arguments

images	An MriImage object, or list of MriImage objects.
colourScales	A list of colour scales to use for each image, which will be recycled to the length of images. See getColourScale for details. The default is to use greyscale.
point	For viewImages, a length 3 integer vector giving the initial location of the crosshairs, in voxels. For defaultInfoPanel, the current location of the crosshairs.
interactive	A single logical value. If TRUE, the plot is interactive.
crosshairs	A single logical value. If TRUE, the crosshairs are displayed.
orientationLabels	A single logical value. If TRUE, orientation labels are displayed.
infoPanel	A function with at least three arguments, which must plot something to fill the bottom-right panel of the viewer after each change of crosshair location. The three mandatory arguments correspond to the current location in the image, the image values at that location, and the names of each image. The defaultInfoPanel function is a valid example.
...	Additional arguments to infoPanel.
data	A list giving the data value(s) at the current crosshair location in each image displayed.
imageNames	A character vector giving a name for each image displayed.

Value

These functions are called for their side effects.

Note

The defaultInfoPanel function is not intended to be called directly. It is a simple example of a valid value for the infoPanel argument to viewImages.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

getColourScale

---

visualisation *Visualise MriImage objects*

Description

Visualise MriImage objects noninteractively using an R graphics device. See viewImages for an interactive alternative.

Usage

```
createCombinedGraphics(images, modes, colourScales, axes = 1:3,
    sliceLoc = NULL, device = c("internal","png"), alphaImages = NULL,
    prefix = "image", zoomFactor = 1, filter = "Mitchell", windowLimits = NULL,
    clearance = NULL, nColumns = NULL)

createProjectionGraphic(image, axis, device = c("internal","png"),
    colourScale = 1, add = FALSE, file = NULL, zoomFactor = 1,
    filter = "Mitchell", windowLimits = NULL)

createSliceGraphic(image, x = NA, y = NA, z = NA, device = c("internal","png"),
    colourScale = 1, add = FALSE, file = NULL, zoomFactor = 1,
    filter = "Mitchell", windowLimits = NULL)

createContactSheetGraphic(image, axis, device = c("internal","png"),
    colourScale = 1, add = FALSE, file = NULL, zoomFactor = 1,
    filter = "Mitchell", windowLimits = NULL, clearance = NULL, nColumns = NULL)
```

Arguments

images	A list of MriImage objects.
image	A single MriImage object.

modes	A character vector of the same length as images, each element being "slice" or "projection" (or abbreviations), indicating which type of visualisation should be applied to each image.
colourScale, colourScales	A single colour scale definition, or a list in the plural case. See getColourScale.
axis, axes	A vector of axes along which slice/projection images should be created. 1 is left-right, 2 is anterior-posterior, 3 is superior-inferior.
x, y, z	Integer vectors, each of length 1. Exactly one of these must be specified to indicate the plane of interest.
sliceLoc	Like x, y and z, except that a point in 3 dimensions is specified. Must not be NA for each axis requested.
device	Either "internal" for display on the default graphics device, or "png" for creating PNG format image file(s). Abbreviations are fine.
alphaImages	A list of MriImage objects to be used as transparency masks. Must be the same length as images if not NULL. NULL values in the list indicate no mask.
prefix, file	A file name or prefix (to which "axial", "coronal" or "sagittal" will be added) to be used when device is "png".
zoomFactor	Factor by which to enlarge the image. Applies only when device is "png".
filter	Interpolation filter to be used by ImageMagick.
windowLimits	Numeric vector of length 2 giving the limits of the colour scale, or NULL for limits matching the range of the image data. Passed as the zlim argument to image.
clearance	Number of voxels' clearance to leave around each slice image in the contact sheet. Passed to newMriImageByTrimming.
nColumns	Number of slices per row in the contact sheet grid. If NULL, the function will aim for a square grid.
add	Overlay the graphic on a previous one. Used only when device is "internal".

Details

These functions create 2D visualisations of 3D images by slicing or maximum intensity projection.

NB: When the device option is set to "png", ImageMagick is required by these functions.

Value

These functions are called for their side effects.

Author(s)

Jon Clayden

References

Please cite the following reference when using TractoR in your work:

J.D. Clayden, S. Muñoz Maniega, A.J. Storkey, M.D. King, M.E. Bastin & C.A. Clark (2011). TractoR: Magnetic resonance imaging and tractography with R. Journal of Statistical Software 44(8):1-18. http://www.jstatsoft.org/v44/i08/.

See Also

See `viewImages` for an interactive alternative, and `getColourScale` for details of how colour scales are specified. Also `image`, which is used as the underlying plot function.

Index