

File Formats of Neuroimaging

John Muschelli

Johns Hopkins Bloomberg School of Public Health

January 25, 2015

DICOM (Digital Imaging and Communications in Medicine) format

- Standardized way of representing images
- Usually how data is given either from scanner or hospital PACS (picture archiving and communication system) system
- 2 integral pieces: Image data (in pixels) and header (meta-data (data about data))
 - Think of a JPEG and a text file

The pixel data in a DICOM file is a matrix (fixed number of rows and columns).

One DICOM file represents one "slice" of the brain.

The `oro.dicom` package is good for reading in this data.

```
library(oro.dicom)
slice = readDICOM('Example_DICOM.dcm')
class(slice)
[1] "list"
```

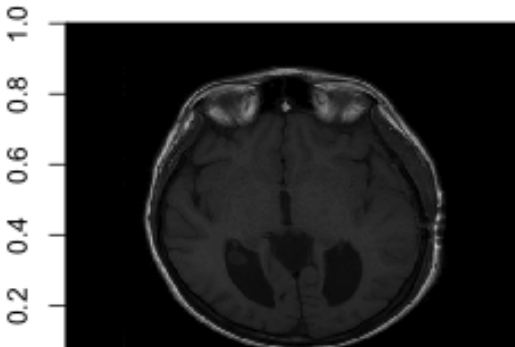
The output is a list with 2 elements: the DICOM header (`hdr`) and image (`img`) information, both of which are lists. Each element of `hdr` has a `data.frame`, and the elements of `img` are matrices:

```
names(slice)
[1] "hdr" "img"
class(slice$hdr)
[1] "list"
class(slice$hdr[[1]])
[1] "data.frame"
class(slice$img)
[1] "list"
class(slice$img[[1]])
[1] "matrix"
```

Display DICOM Image

We can display the data using the `image` command.
(We transpose the data using `t()` so the image faces “up” instead of “right”.)

```
image(t(slice$img[[1]]), col=gray(0:64/64))
```



What about the header?

There are many fields, for example pixelSpacing, which is the dimensions (x and y) of a pixel in millimeters (mm):

```
hdr = slice$hdr[[1]]  
hdr[hdr$name == 'PixelSpacing', "value"]  
[1] "0.46875 0.46875"
```

Multiple DICOM files

We have discussed only one slice of the brain. What about multiple slices? If you pass a directory into `readDICOM`, it will read in all DICOM files in that directory.

```
all_slices = readDICOM('T1/')
```

Now that we have multiple files read in, we can convert it to a 3-dimensional (3D) array, where you can think of the array as stacking each slice (which is a matrix) on top of each other. If each DICOM is a piece of paper, the 3D array is a stack of paper. The way we store this 3D array is in the NIfTI (Neuroimaging Informatics Technology Initiative) format.

	DICOM	NIfTI
File extension:	.dcm	.nii or .nii.gz (compressed)
Each file is a:	slice of the brain	3D image of brain
Header information:	Many fields, protected health information, hospital-related meta-data	Image meta-data, no patient information
Different Images	Different Folders	Different Files (can be same directory)

We can convert this list of header information and image information to a `nifti` object (an R object) with the `dicom2nifti` command:

```
nii = dicom2nifti(all_slices)
Error in nim@pixdim[2:4] <- attr(img, "pixdim"):
could not find function "checkAtAssignment"
dim(nii); class(nii)
Error in eval(expr, envir, enclos): object 'nii' not
found
Error in eval(expr, envir, enclos): object 'nii' not
found
```

We can see that this `nii` object is indeed a `nifti` object and has 3 dimensions.

Now that we have multiple files read in, we can convert it to a 3-dimensional (3D) array, where you can think of the array as stacking each slice (which is a matrix) on top of each other. If each DICOM is a piece of paper, the 3D array is a stack of paper. The way we store this 3D array is in the NIfTI (Neuroimaging Informatics Technology Initiative) format.

	DICOM	NIfTI
File extension:	.dcm	.nii or .nii.gz (compressed)
Each file is a:	slice of the brain	3D image of brain
Header information:	Many fields, protected health information, hospital-related meta-data	Image meta-data, no patient information

For Philips scanners, files from the scanner are PAR/REC and not DICOM. `dcm2nii` can convert to NIfTI.

NIfTI format was based on ANALYZE format where the header and image were in separate `.hdr` and `.img` files. This is an older format and we will not use this way of storing data because 1)

NIfTI can have one file with both header and image information, and 2) Can be stored as compressed `.nii.gz` files.

NRRD (Nearly Raw Raster Data) is another format similar to NIfTI. Much of the neuroimaging software can read in both NRRD and NIfTI files, but NIfTI is much more common.