# DICOM Handling

DICOM is the standard that includes medical image files. It is widely used by the manufacturers of medical image scanners and suppliers of hospital PACs systems for transmitting, viewing and archiving images.

The aspects discussed here are:

- Organisation of information and images in a DICOM file.
- The DICOM model of Study, Series, Instance
- "Classic" and "Enhanced" DICOM.
- FIle naming and organisation.
- Viewing images amd metadata.
- Scaling of data.
- Anonymisation and Protected Health Information.
- Discussion.

## Organisation of information and images in a DICOM file

It is vital that images are correctly associated with the patient to which they belong. In DICOM, the files contain both the image and associated meta-information, for example, date, patient name, age, position of image in patient, scanner details etc. The meta-information, or metadata, can be read from a DICOM file in MATLAB using `dicominfo`. Although the format is different, this is similar in concept to photographic tiff images that also store informatin about exposure, photo location etc.

## The DICOM model of Study, Series, Instance

The terms "Study", "Series" and "Instance" have special meanings and relate to the DICOM model of a patient visit. A "Study" roughly corresponds to the whole session for a patient in a scanner. During this session there may be multiple "Series" acquired and each Series will contain one or more image "Instances". For example, the images used in an initial survey acquisition may be one Series, then in MRI a T1-weighted volume might be another Series and a T2-weighted volume another Series.

Different Unique Identifiers (UID)s are assigned to every Instance, Series and Study. These are usually long strings of numbers and dots. Their benefit is that they allow software to keep track of images, scans, sessions and patients. Series usually also get a text name to help identification e.g. "Survey", "T1W"

## "Classic" and "Enhanced" DICOM

Throughout DICOM, it is useful to think of images as just 2D slices with meta-data about their position, but it is up to the user to assemble these slces together into a 3D volume or a time series. In Classic DICOM, there is one image per file, whilst in Enhanced DICOM, there are multiple images per file, usually one file for all the images in one Series. In MR, Enhanced DICOM also has more meta information about technical aspects of the scan. Enhanced DICOM has the advantages of more information and many fewer files, whereas the older Classic format can be read by all DICOM handling software.

## FIle naming and organisation

Although some software tries to name DICOM files in a meaningful way, in general the filename conveys no useful information. You should not sort files or infer meaning from the filenames - instead the meta-data inside the files needs to be inspected. In Classic format, a patient Study with multiple volumes and time frames can

result in thousands of images, each in a separate file and these files can be spread over multiple folders making handling difficult. In MATLAB the function `dicomCollection` can parse data efficiently. There is also an app called `dicomBrowser`.

```
% Here use in-built data examples
dataDirectory = fullfile(matlabroot,"toolbox/images/imdata/") ;

dc = dicomCollection( dataDirectory )
```

dc = 5×14 table

...

| | StudyDateTime | SeriesDateTime | PatientName | PatientSex | Modality |
|---|---|---|---|---|---|
| 1 s1 | [] | [] | "" | "" | "RTSTRUCT" |
| 2 s2 | 30-Apr-1993 11:27:24 | 30-Apr-1993 11:27:24 | "Anonymized" | "" | "CT" |
| 3 s3 | 03-Oct-2011 19:18:11 | 03-Oct-2011 18:59:02 | "" | "M" | "MR" |
| 4 s4 | 03-Oct-2011 19:18:11 | 03-Oct-2011 19:05:04 | "" | "M" | "MR" |
| 5 s5 | 30-Jan-1994 11:25:01 | [] | "Anonymized" | "" | "US" |

The variable `dc` is a Table and just selected columns can be viewed, e.g. to see selected columns and all rows:

```
dc(:,{'SeriesDateTime','SeriesDescription' ,'Rows','Columns','Frames'})
```

ans = 5×5 table

| | SeriesDateTime | SeriesDescription | Rows | Columns | Frames |
|---|---|---|---|---|---|
| 1 s1 | [] | "" | 0 | 0 | 1 |
| 2 s2 | 30-Apr-1993 11:27:24 | "" | 512 | 512 | 1 |
| 3 s3 | 03-Oct-2011 18:59:02 | "" | 512 | 512 | 1 |
| 4 s4 | 03-Oct-2011 19:05:04 | "" | 512 | 512 | 1 |
| 5 s5 | [] | "PS LAX MR & AI" | 430 | 600 | 10 |

To pick out one Series, choose its row as the first index into the Table. e.g.

```
dc("s2",:)
```

ans = 1×14 table

...

| | StudyDateTime | SeriesDateTime | PatientName | PatientSex | Modality |
|---|---|---|---|---|---|
| 1 s2 | 30-Apr-1993 11:27:24 | 30-Apr-1993 11:27:24 | "Anonymized" | "" | "CT" |

## Viewing Images and Meta-Data

For volume data, the MATLAB function `dicomreadVolume` can be used. This has the advantage that it will sort the slices into order (remmebr that listing files alphabetically does not guarantee they are in any particular order).

```
% You shoud be in the folder above the source and data folders for this to
% work (or edit the next line)
dataLocal = 'data/MathWorks_dicom_images' ;
dcbrain = dicomCollection( dataLocal )
```

```
Error using images.internal.dicom.CollectionLoader/describeSource
Could not open specified directory or file.
Error in images.internal.dicom.CollectionLoader (line 35)
            obj.describeSource(source, includeSubdir);
Error in dicomCollection (line 18)
loader = images.internal.dicom.CollectionLoader(source, recursive);
```

```
[V, spatial] = dicomreadVolume(dcbrain) ;
whos V
```

Note that V is of type `int16` because within DICOM files, the pixel data is stored as integers and `dicomreadVolume` does not apply any rescaling. Also, for grayscale images (not RGB), the size of the thrid dimension returned here is 1. To see a single slice, for example slice 10:

```
figure
imshow(squeeze(V(:,:,1,10)),[])
% squeeze removes dimensions that are size 1.
% So if V had size [256 256 1 20], then squeeze(V) wiil have size
% [256 256 20]
```

Function `dicomdisp` can be used to display all the meta-data on the screen, alternatively `dicominfo` will place the data in a structure, which can be used elsewhere in code.

```
filenames = dcbrain.Filenames ; % here returns a 1x1 cell array
thirdFile = filenames{1}(3) ; % {1} gives contents of the cell array,
% The itself is a 20x1 string array - take the 3rd file as an example

dinfoBrain = dicominfo( thirdFile ) ; % get the metadata for this file

disp("Patient age: " + dinfoBrain.PatientAge)
```

and to see all the information in the structure

```
dinfoBrain
```

## Scaling of data

Pixels are usually stored in DICOM as integers. If the underlying data is floating point or outside of the range of the integers used, then data might need to be linearly scaled using a `RescaleSlope` and `RescaleIntercept`. Typically this might be to get Hounsfield numbers in CT, or computed parameters in MR e.g. ADC diffusion values. In some DICOM files, the `RescaleSlope` and `RescaleIntercept` are replaced with `RealWorldValueMapping` parameters, which form a similar purpose but will not be discussed further here.

For the ankle CT data in the original dicomCollection

```
filenameAnkle = dc("s2",:).Filenames ; % Ankle was 2nd row. Here the output
is not a cell.
dinfoAnkle = dicominfo( filenameAnkle ) ;

disp("Rescale Slope: " + dinfoAnkle.RescaleSlope + ...
     "    Intercept: " + dinfoAnkle.RescaleIntercept )
```

Now, we need to get the pixel data (which will be integers), and apply the rescaling to get Hounsield units. As the file is just one slice, we cannot use `dicomreadVolume` and need to use `dicomread`

```
img = dicomread(filenameAnkle) ;
whos img

% img is int16, convert to double, otherwise risk loss of precision, and
% rescale to Hounsfield units
imgHounsfield = (double(img) * dinfoAnkle.RescaleSlope) +
dinfoAnkle.RescaleIntercept ;

figure
imshow(imgHounsfield,[]), colorbar
```

BEWARE when reading pixel data, or metadata from a file, that either may be read as an integer type, and if you multiply an integer with a doube in MATLAB it (annoyingly) returns an integer. Hence there is the risk of loss of precision. In the above, the `img` data had to be converetd to double before the rescale was applied.

## Anonymisation and Protected Health Information

In the brain data above, we could see the subject's age at the time of the scan, but the name is blank:

```
dinfoBrain.PatientAge
dinfoBrain.PatientName
```

It is likely that the Patient Name has been deliberately removed after the scan - a process often termed "anonymisation". This is a difficult topic that will be discussed in detail elsewhere. There are a few comments below.

## Discussion

Reading DICOM data can be fiddly, sometimes needing trial and error. For example, in the above the filenames were in a cell array in one example, and not in another. Writing DICOM is harder, and a lot harder to get fully correct. Many people convert DICOM to NIfTI format but this has two major disadvantages. Firstly, a lot of software that uses NIfTI does not correctly handle geometry and you can end up with images are have slices reversed, flipped or transposed and this is both unsafe clinically and makes comparison with other clinical images error-prone. Second, the meta-data is no longer with the image (as NIfTI has only a brief header) so it might be difficult or error-prone to correlate image data with other patient data, for example histology results.

Anonymisation is also a difficult topic. My current advice is that if possible, keep original, identifiable DICOMs direct from the scanner, in a safe location such as an encrypted hard drive, data safe haven or trusted research environment. Work on de-identified copies and try to avoid NIfTI. For de-identification, use quality software such as DicomCleaner. For saving of intermediate results, use `.mat` files and include both the processed data and a copy of the output of `dicominfo` that has been run on the de-identified data. If you have to use NIfTI, use a reliable convertor such as dcm2niix.

*David Atkinson.*

```
disp("Live Script last run: " + string(datetime("now")))
```