# Imgtools User and Contributor Guide

Imgtools is a collection of MATLAB files for processing medical image data, primarily MRI data in DICOM format. Much of it has been developed in spare time or without resources for extensive testing. The code must not be used clinically and it comes with no warranty.

## Code

A Git repository is used for code development. Until March 2017 this was kindly hosted by Assembla (under the name imtools) but was then switched to GitHub and renamed imgtools (with the extra ‘g’ to distinguish the two). Users either have access to the source code in the GitHub repository, or, a zip file that is released periodically. Both the imgtools repository and the zip file exclude files that need to be separately downloaded (e.g. NODDI, PROPACK or MIRT toolboxes).

MATLAB needs to know where the imgtools files are located. You only need to do this once, but edit your startup.m file as explained in README.txt. This file also lists the other toolboxes you may need to download.

A typical configuration would be:

HOME/matlab/imgtools - either the repository, or the unzipped files

HOME/matlab/external - downloaded external files that are required.

A separate document explains the use of the GIT repository.

## Example Usage

dinfo = datparse(dselect) ;

[vol, mat\_prop] = d2mat(dinfo,{‘slice’},’op’,’fp’) ;

q = myanalysis(vol, mat\_prop) ;

The dselect function is for MultiFrame DICOMs and lists available files. The output can be passed automatically to datparse as above. For SingleFrame DICOMs, just use dinfo = datparse ; select any one of the files of interest and then the single folder containing the DICOM files and possible sub-directories of DICOM files.

The dinfo structure holds a minimal set of data (e.g. filenames, slice positions, temporal information, TE, TR). It can be saved and reloaded later.

d2mat takes the dinfo structure and produces a MATLAB matrix vol and structure with properties. The first two dimensions of the matrix are always image rows and columns, the user specifies the subsequent dimensions, e.g. slice, b-value, flip angle, etc. d2mat can be hard to use as it expects to find in the input files exactly one value to place in every element of the output matrix. So, if you specified just ‘slice’ and in the input structure there were two images for each slice, say for two flip angles, it will fail. There is some guidance and examples provided in the file. The more you sort and restrict the input data, the fewer problems you are likely to have. If you are not too bothered about a nice matrix, you can use the ‘frame’ option for d2mat to just read the frames of data into a volume that has dimensions [ny nx nframe].

## Design Strategy

* Code execution is a three step process:
  + Parse DICOM, PAR/Rec or other files to generate a MATLAB structure that holds the key information (TE, TR, patient geometry etc). The code here can be specific to various flavours of DICOM, manufacturer etc. This step can be slow to execute, but if the structure is saved, the step needs only to be run once.
  + Process data, e.g. calculate T1. Minimise in this code any manufacturer dependency.
  + Save data.
* The structure holding key information follows the DICOM standard for geometry (e.g. uses a LPH coordinate system and the ImageOrientationPatient and ImagePositionPatient tags) and DICOM-style names for parameters.
* Approximate priority order:
  + Numerical accuracy in algorithms and use of correct parameters from data.
  + Ease of use and understanding.
  + Ease of re-use by others.
* Where possible, embed help and examples in the code files, rather than in separate documents that are harder to maintain and easily loose synchronisation with code.
* Error checking: Add error checking to help the novice user or to catch issues that will cause numerical errors later in the code. Too much error checking can make code difficult to read or test – focus on areas where errors are likely to be made, or where numerical issues could have serious consequences.
* Reduce hard coding of data sizes, file and directory names. Make use of MATLAB’s uigetfile etc for getting file names and their file name processing functions (fileparts, filesep etc) to maintain portability across platforms. Any hard-coding should be done using variables set at the start of the file. For example if you had a = zeros([128 128]); in the code, replace with np=128; at the top of the code and a=zeros([np np]); within the code. If possible, get np from the data itself e.g. np = size(vol\_in,1) ;
* **Clinical Use**. We try to avoid legal liability but we want our code to be useful. Be careful about giving out code that writes DICOMs because a) it is very hard to get this writing correct in every respect, b) DICOMs are “trusted” to be correct. It is very difficult for a clinician or physicist at another site to re-implement the code so you should assume that the code will leak and be used at least as part of clinical research. Think carefully about the likely accuracy of numerical data. If necessary, put in error checks to prevent the code running for situations you have not tested or designed for. Always be alert to potential inadvertent left-right swaps that are hard to spot. For geometry processing, understand the issues, do not experiment by transposing matrices and swapping signs until it “looks right”.

## Key Features

Useful features of the code include:

* Checking and automatic re-ordering of slices. This uses only the ImagePositionPatient and ImageOrientationPatient tags and assigns a slice number itself. It does not use the manufacturer’s slice number which can be ambiguous.
* Ability to read data from multiple DICOM series at once.
* No need to sort DICOM files first (though it can help). For SingleFrame DICOMs, the user can supply the series numbers to help in the data reading, but does not need to split into multiple folders.
* Can apply re-scaling to the data (especially useful for Philips data).
* Allows user to restrict by a range of slices, series etc.
* Allows user to downsample on reading in.
* Limited checking of FrameOfReferenceUID so a bed or patient repositioning may be ignored. **This can be dangerous if the patient was moved because the geometry will no longer be consistent.**
* Ability to re-slice one dataset into the geometry of another (see dreslice).

Geometry and Reslicing

The DICOM LPH convention is used throughout. For Philips PAR/REC input files, dgeom converts from the offsets and angulations to DICOM-style IPP and IOP. Note this code is reverse engineered and should not be relied upon. Generally within the functions, a structure called geom contains some of the geometry information. The function dgeomextractcreates a geom structure from a dinfo structure. This function also sets the XData and YData for use by the roianalviewing package and some information for MATLAB’s newer image registration functions. Historically the GUI eshow uses a vector vdims to specify sizes (in mm) of each dimension of a voxel. vdims is a field of a geom structure.

**Geometrical Reformatting.**

dreslice allows the user to interpolate one data set at the geometry of another. For example a high resolution T2 image could be interpolated to the same slice positions as a diffusion image. Using the ‘PixelSpacing’,’input’ option, the input resolution can be preserved. More details are in the function help.

## Anonymisation.

In general I recommend DicomCleaner from David Clunie for anonymisation.

paranon is a GUI that will remove the patient name from the filename and within a Philips PAR file.

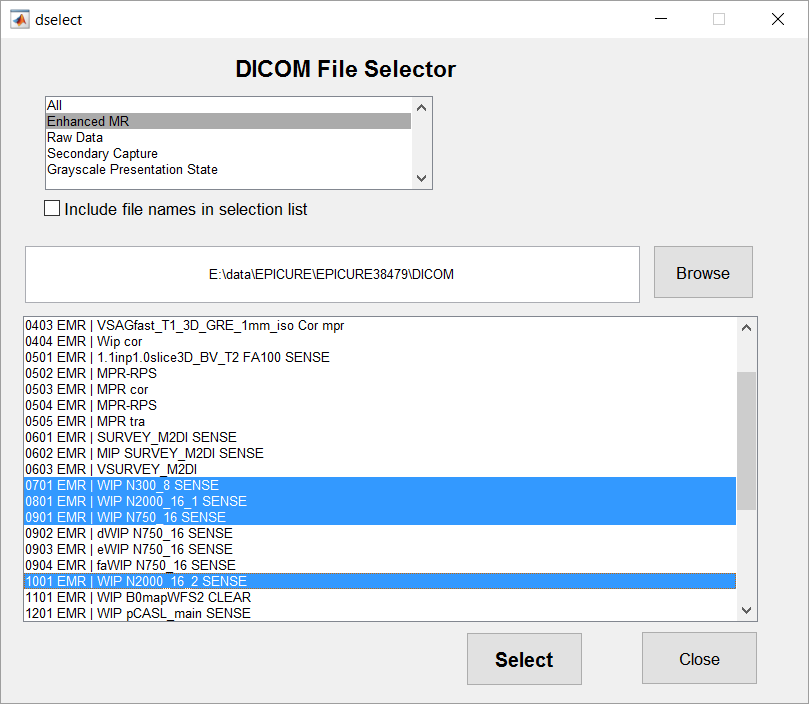
dicoman will keep only the DICOM tags specified in dinfostrip (for SingleFrame DICOM only). This is an aggressive anonymisation that may not be compatible with all subsequent use, though the study UIDs are replaced in an intelligent way and the files can at least be read with kPACS.

## Renaming

DICOM files are often given meaningless names by medical equipment. **dicoman** can add scan, slice and time point numbers to file names. For MultiFrame DICOM **dmfrename** will rename files to add the scan name and number. Note it can freeze or run out of memory when the file contains a lot of images (e.g. QUASAR ASL with 7000+ files). With the availability of dselect that produces a summary, dmfrename is nolonger used often.

## Diffusion Data

DICOM files from the scanner may contain individual diffusion weighted images and/or processed images such as the mean diffusivity, ADC, FA etc. Reading from a MultiFrame DICOM file needs some care as there may be more diffusion weighted than b=0 images present in the file. For data acquired over more than one series (e.g. the shells of a NODDI acquisition), it is important to preserve the scaling. In the example below, we have 4 series with data.



dinfo = datparse(dselect) with the above selection puts data from 4 series into dinfo. There are a number of ways to handle this data with some examples in the help of d2mat. To restrict the output from d2mat, you can use ‘series’, ‘bv’, or ‘ddty’ among others. A ‘ddty’ of 1 means diffusion directionality (an image from one diffusion weighting gradient), this will not be 1 for computed mean diffusivity images.

An alternative is to use dmf2d. This can be slow, but if you restrict the slices, you can easily give it a try. See the examples in the help for dmf2d.

## Reading Philips Raw Data

This makes extensive use of the code supplied to the Philips community forum. This part of the code is currently (October 2017) in the imgtools-external repository. Broadly, the process is to read the Philips .sin file and the k-space using e.g.

[dataB0, infoB0] = main\_loadLABRAW([],'verbose',true,'correct\_nus',false,'EPIphasecorrection', false);

and then reconstruct to the image domain using, e.g.

[ datat, geomt ] = explore\_recon(dataB0, infoB0,'MPS\_only',true) ;

The ‘explore’ name means it is code in development that will eventually be converted to a recon function. Throughout, the spatial referencing is handled in a DICOM-like fashion, defining the image plane using a point in the plane (ImagePositionPatient at top left pixel) and ImageOrientationPatient as two vectors in the plane pointing along the image row and column directions. The through-slice direction is defined here as the cross product of the first and second IOP vector. After the FT to image domain, the IPP, IOP and other parameters in the geom structure are updated for all in plane transformations, e.g. cropping, rotating, as part of the dinplanet function.

FT to the image domain here uses the *forward* FT (which unfortunately is the opposite of the convention used in most of my older code) and the functions are given names such as k2ir with the ‘r’ denoting reconstruction. The data has readout (FE) as the first dimension which can be confusing as this is labelled ‘x’, even though it is the first dimension in MATLAB (i.e. the row dimension). The readout is always the first dimension of the data array until the in plane transformations used to rotate/flip into radiological viewing orientation.

Immediately after the FT, the orientations (IOP) are given by the M and P directions. The IPP is calculated from the ‘origin’ and this point is taken to have the position determined by the offcentres applied in acquisition (including acquisition phase and frequency changes that result in an offset at the time of FT). For EPI with slope sampling, there is no M or P offcentre applied in acquisition – it is expected to be done after recon by shifting the pixels. In the Philips recon, there is a ‘mistake’ because the offcentre applied by shifting is the user-specified offset, rounded to the nearest reconstruction voxel, but the number supplied to the DICOM IPP is not rounded.

For non-EPI, the PPE code appears to suggest that a half-pixel shift is added to the user-specified offsets in M and P when the recon size is even. I assume this is to put the “middle” of the array at a voxel boundary. This extra acquisition offset is assumed in the imgtools code when it computes the IPP for non-EPI data.

The code has been tested on phantom data sets for combinations of phase encode direction, orientation and fat shift direction. The testing functions are generally in the Philips folder and called check\_\*.m or explore\_\*.m

## Features / Bugs / Future Work

data\_process and dreslice separately check the slices – should be done using one function.

PAR reading may not read and process all diffusion data since the upgrade for Diffusion Directionality in the Multiframe DICOM reader.

The handling of vdims by eshow and its storage in dinfo and geom is confusing.

DICOM writing is not perfect and the handling of RescaleSlope when writing DICOMs may not be strictly correct.