

## Introduction

The program imports modified or simulated SPECT projection data and CT slices into the DICOM format to enable their use and reconstruction on clinical workstations. This program was developed as part of work package 3 of the MRTDosimetry project, which aimed at producing a reference database for SPECT CT imaging.

## Workflow

The general workflow of the program is described in Figure 1. The user provides the SPECT and CT DICOM files of a SPECT examination to be used as templates (1) and a set of images (simulated SPECT projections and CT reconstructed slices (2)). The user also selects the workstation to be used for the reconstruction (3). The program modifies specific DICOM tags in the templates, according to the original SPECT system and the specified workstation, and replaces the SPECT projections and the CT reconstructed slices of the templates with the provided ones. The user can then perform the reconstruction of the newly created DICOM files on the specified workstation following usual practice (5).

## Availability

The script was written in Python (Python 3.6.1, Python Software Foundation, USA). It is freely available as a Python script (.py) and as a Windows standalone executable file (.exe) created using PyInstaller. Both formats of the program (.py and .exe) are callable from the command prompt.

The following libraries are necessary to run the Python script: dicom, pathlib, os, SimpleITK, numpy, argparse and sys.

The list of the DICOM tag modifications (including the import of the projection data and the reconstructed CT slices) can be found in Table 1; if the reconstruction is to be performed on a Hermes workstation, additional modifications are performed, as reported in Table 2. More details on the DICOM modifications and on the program working can be found in the report of the program development available on demand.

## Disclaimer

This program was written in the framework of the European Metrology Programme for Innovation and Research- (EMPIR-) Metrology for clinical implementation of dosimetry in molecular radiotherapy (15HLT06 MRTDosimetry).

The program is distributed **without any warranty...TO BE COMPLETED**



## Usage

The usage of the program is described below; optional arguments are reported in brackets. Usage of the .exe and the .py versions are identical, except that “python” should be added at the beginning of the command line when the .py version is used. The order of the arguments has no importance but they must be preceded by “-x” or “--x\_full\_name”.

usage .exe:

```
usage: DICOM_modify.exe [-h] -m MODEL -w WORKSTATION -i INPUT_FOLDER -e  
      IMAGE_EXTENSIONS [IMAGE_EXTENSIONS ...] [-o OUTPUT_FOLDER]
```

usage .py:

```
python "DICOM_modify.py" [-h] -m MODEL -w WORKSTATION -i INPUT_FOLDER -e  
      IMAGE_EXTENSION [-o OUTPUT_FOLDER]
```

The description of the program arguments is reported below:

Arguments:

- h, --help show this help message and exit
- m MODEL, --model MODEL  
"Brightview XCT", "Discovery 670", "Infinia Hawkeye4",  
"Optima 640", "Symbia T2", "Symbia Intevo Bold"
- w WORKSTATION, --workstation WORKSTATION  
"Hermes", "Jetstream", "Syngo", "Xeleris"
- i INPUT\_FOLDER, --input\_folder INPUT\_FOLDER  
Path to the folder containing the original DICOM files  
and the simulated images "C:\Users\..."
- e IMAGE\_EXTENSIONS [IMAGE\_EXTENSIONS ...], --image\_extensions IMAGE\_EXTENSIONS  
[IMAGE\_EXTENSIONS ...]  
List of file extensions of the simulated images (".mhd", ".png", etc.)
- o OUTPUT\_FOLDER, --output\_folder OUTPUT\_FOLDER  
Optional: Path to the folder where the modified images are to be saved "C:\Users\..."

## Specifications

### Folder structure

A specific folder structure has to be created to input the data (Figure 2). All input files have to be stored in the same folder. Within this folder, the original SPECT and CT files have to be stored in two separated, dedicated subfolders (input\_folder\SPECT and input\_folder\CT, respectively). Simulated SPECT and CT images, if used, also have to be stored in dedicated folders (input\_folder\sim\_SPECT and input\_folder\sim\_CT, respectively).

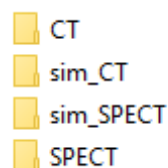


Figure 2: Folder structure for data input

A folder where to save the processed files can also be provided. However, this is not necessary. A specific output folder will be created if none is provided.

If no SPECT (no CT) DICOM files are provided in the subfolders, the program will modify the CT (SPECT) files, without generating an error message.

#### Input of the DICOM files and selection of the SPECT CT system

The program was tested with DICOM files generated by five SPECT CT systems: Siemens Symbia T2 and Symbia Intevo Bold (UKW); GE Discovery 670 (LUND) and GE Optima 640 (RSCH); Philips Brightview XCT. The program can handle files from several examinations performed on the same system in one run.

Selection of the SPECT CT system is only possible among the five tested systems. The program does not actually check which SPECT CT system generated the DICOM files, but only relies on the name input by the user, and modifies the tags of the DICOM files accordingly.

#### Input of simulated projection data and CT slices

The user should provide as many simulated projections or CT slices as contained in the DICOM files. The program does not check if the resolution of the provided images matches the original DICOM images, nor does it modify the DICOM tags related to the image characteristics.

The simulated SPECT projection data can be provided in one file or in as many files as there are input SPECT DICOM files. The same rule applies to CT reconstructed slices. For example, if 6 SPECT DICOM files each containing 120 projections are to be used as templates, one file containing 720 simulated projections or six files containing 120 simulated projections can be input indifferently. If projections are provided in different files, the program uses the projection data from the first file in alphabetical order; if several projections are provided in a single file, the order of the projections is then used as provided. Therefore, the user should make sure that the projections are provided in the same order as stored in the DICOM files. This means that if six DICOM files and six simulated image files are provided, the first image ranked following the alphabetical order will be used with the first-ranked DICOM file, and so on.

More than one type of images can be used in a single call to the program. Currently, the program has only been tested with “.mhd” and “.tif” images. Other grayscale image types should be usable if their characteristics match the images of the DICOM templates, and if they are recognised by – simple – ITK, which is used for image processing.

#### Selection of the workstation for reconstruction

The program was tested on three vendor's workstations (Jetstream, Syngo and Xeleris) to reconstruct examinations generated on SPECT CT systems from the respective vendor (Philips, Siemens and GE); one vendor-independent workstation (Hermes) was tested to reconstruct examinations generated on SPECT CT systems from all vendors. Selection among those four workstations is possible. In the current program version, the selection of any of the three vendor's workstation is equivalent and results in identical output; the selection of Hermes workstation delivers different results.

## Step-by-step

0) Construct the input folder structure as described in Figure 2.

1) Start the Windows command prompt by typing “cmd” in the search window.

2) Set the path to the disk and directory containing the .exe (py) file

Example: .>C:

.> cd "C:\Users\jdoe\Desktop\data"

3) Display the help message:

Example: .> DICOM\_modify.exe -h

### 3') Run the executable .exe

Example: modifying files from a GE Optima 640 system using “.mhd” images to be reconstructed on a Xeleris workstation. The input folder is “input” and no output folder is provided:

```
.> DICOM_modify.exe -m "Optima 640" -w "Xeleris" -i "C:\Users\jdoe\Desktop\input" -e "mhd"
```

Example: modifying files from a Philips Brightview XCT system using “.tif” images to be reconstructed on a Jetstream workstation. The input folder is “data” and the output folder is “results”:

```
.> DICOM_modify.exe -m "Brightview XCT" -w "Jetstream" -i "C:\Users\jdoe\Desktop\data" -e ".tif" -o "C:\Users\jdoe\Desktop\results"
```

### 3'') Run the python script (.py)

Example: modifying files from a GE Optima 640 system using “.tif” and “.mhd” images to be reconstructed on a Hermes workstation. The input folder is “input” and no output folder is provided:

```
.> python "DICOM_modify.py" -m "Optima 640" -w "Hermes" -i "C:\Users\jdoe\Desktop\input" -e ".mhd" ".tif"
```

Example: modifying files from a Discovery 670 system using “.tif” images to be reconstructed on a Xeleris workstation. The input folder is “data\_in” and the output folder is “results”:

```
.>python "DICOM_modify.py" -m "Discovery 670" -w "Xeleris" -i "C:\Users\jdoe\Desktop\data_in" -e ".tif" -o "C:\Users\jdoe\Desktop\results"
```



Table 2: Additional DICOM modifications for image import and reconstruction on Hermes workstation

		GE			Siemens			Philips	
		Optima 640	Discovery 670	Infinia Hawkeye4	Symbia T2	Intevo Bold (Homogeneity- corrected)	Intevo Bold (Advanced)	Brightview XCT	
	(0011,100D)	Private	Range of the energy windows replaced with "No scatter windows"						
	(0011,1012)	Private	Description of the energy windows replaced EM1"						
	(0011,1030)	Private							
	(0011,1050)	Private							
	(0028,0008)	Number of frames		New number of frames					
	(0035,1001)	Private				Description of the energy windows replaced with "Simulated EM1"			
	(0054,0010)	Energy window vector		Number of encoded values shortened to new number of frames (originally equal to number of frames)					
	(0054,0011)	Number of energy windows							
	(0054,0012)	Energy window information sequence		Only first sequence kept (originally one sequence for each energy window)					
	(0054,0020)	Detector vector		Number of encoded values shortened to new number of frames (originally equal to number of frames)					
	(0054,0050)	Rotation vector							
	(0054,0090)	Angular view vector							
	(0055,1012)	Private		Only first sequence kept (originally one sequence for each energy window)					