Duetto Toolbox: User Manual

Table of Contents

1. Introduction

Duetto is the name of the GE PET toolbox that houses MATLAB and MATLAB-callable C and C++ code that enables reconstruction and analysis of PET data. Duetto is compatible with many PET/CT scanners and the SIGNA PET/MR scanner. This document provides some documentation of the code architecture (not the algorithms themselves), describes the steps needed to perform an offline reconstruction, and describes some useful I/O functions.

Duetto uses an object-oriented paradigm. Algorithm parameters are organized into a number of different objects. The default values of these objects are populated based on the specified scanner type and reconstruction algorithm. Figure 1 shows the high-level steps of the toolbox.

Defaults are first populated into the userConfig structure. These default values can be overridden by the user, and other certain fields (described in the Parameter Classes section) can be added to the userConfig structure to override defaults in the objects. The raw data is then extracted from the provided 'raw' data folder. DICOM header values and userConfig parameters are used to initialize all objects.

The next main step is to generate the corrections files. These functions compute or model the system and physics effects, such as scanner geometry, detector deadtime/pile-up, attenuation, random events, and scatter events.

Raw data and the correction files are input into the image reconstruction module.

Finally, any post-processing is performed, such as transaxial or z-filtering. There is an option to write DICOM files to disk in addition to a 3D matrix file (in ".sav" or ".mat" format).
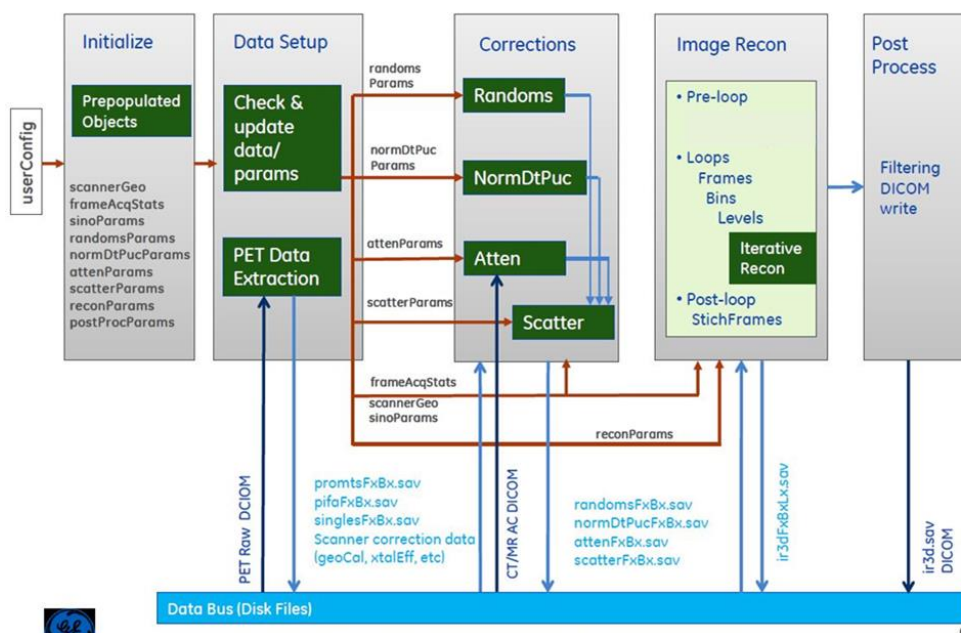

Figure 1. Duetto workflow

2. Getting Started with Duetto

First Steps and Important Notes

1. Supported MATLAB versions include 2017b to 2022b.

2. Directories and filenames should not contain any spaces. Directory names are case-sensitive.

3. GE list and raw data are compressed using proprietary filters. For processing list and raw data with raw data files version 10 (RDFv10), a script is available in Duetto to facilitate the decompression settings. For Linux users, run `start_toolbox_MATLAB.sh`, which will enable the RDFv10 decompression libraries and start MATLAB.

4. Run `initializeDuetto`, which will perform two crucial tasks:

    a. It will add all of the toolbox scripts to the MATLAB path.

    b. It will check for the location of the PET/MR MRAC package of executables and templates (for PETMR). The default location is at same level as the duetto directory. The default value can be changed by setting `userConfig.mracExtDir` in ptbUserConfig or in the user run script. The directory will then be automatically added to the MATLAB path.

5. The toolbox uses MATLAB compiled mex files, which are operating system and library dependent. Those included with the toolbox distribution should work for Linux operating systems. Compatibility with Mac and Windows OS is improving but may not yet be fully supported.

6. A license file, `Duetto_license`, is available in the Duetto root directory within a released package. A function, `ptbLicenseCheck`, can be used to show the number of days left to the license expiration date.

7. Example scripts to get started with Duetto are contained in the runScripts sub-folder.

8. Filenames for separate bed positions and time bins are labeled using
    *   f = frame, for multiple bed positions
    *   b = bin, for multiple time frames or bins

    Using the raw data files (RDFs) as an example, the first file would be `rdf_f1b1.rdf`. Multiple beds would increment as `rdf_f2b1.rdf`, etc., and multiple time bins would increment as `rdf_f1b2.rdf`, etc.

9. The userConfig structure can be saved as a .mat structure in order to view or use later. Note that Duetto must be loaded into the MATLAB path for Enums to be recognized.

Required Data and Image Files
Reconstruction using the toolbox requires data from the scanner, which includes:

*   PET raw data

- CTAC or MRAC images or PIFA images, which are used for attenuation correction. The CTAC and MRAC images are used to generate a PIFA image. The PIFA image is forward projected into the attenuation correction sinogram. If running MRAC for a phantom study or patient study with truncation completion, TOFNAC images may also be provided. It is recommended to reconstruct these on the scanner; TOFNAC reconstruction parameters are described in the Appendix.
  Note: Running MRAC requires the folder duettoMracFiles to be in the MATLAB path!

The PIFA images should be named as:

- `pifa_fMbN.pifa`
- `pifaIvv_fMbN.pifa` (PET/MR only)

where `M` is the bed position, and N is time bin.

The in-vivo PIFA is used only in PET/MR (`pifaIvv_fMbN.pifa`), and contains only the patient or PET phantom. The complete PIFA (`pifa_fMbN.pifa`) is the sum of the in-vivo PIFA and ex-vivo PIFA. The ex-vivo PIFA contains certain PETMR-supported attenuating media outside of the patient or phantom, such as the fixed coils and table. These are read from object templates in the supporting duettoMracFiles directory.

Both PET/CT and PET/MR utilize one PIFA per bed position, and they utilize the same PIFA for all dynamic time frames or gates within that bed position; this means that `N=1`. Symbolic file links pointing to the first PIFA will be generated for `2:N` time bins. Note that the user can also create their own PIFA files for each time frame; they just need to be given the correct name, incrementing `N` appropriately.

PIFA files can be obtained from the scanner within a short time after image reconstruction. They are deleted from the scanner after ~1 day for PET/CT and after a few weeks for PET/MR. A retro-reconstruction can be performed at any time to restore them.

Please see the Transferring Scanner Data section for detailed instructions on how to obtain raw files and PIFA files. The Data Organization section describes how the folders and files should be organized.

The PIFA, or AC map, represents the linear attenuation coefficient, commonly referred to as the µ value. It is a scaled version of the traditional units so that it is in a reasonable viewing range. The units are $100,000 \times$ the µ511-kev value in mm^{-1}. Therefore, air is 0 and water is approximately 960.

3. Transferring Scanner Data

PET/CT
Both the PET raw data (non-list file) series and CTAC DICOM series need to be transferred to the external server.

Raw data
Select the desired exam in the Exam Database, and highlight the PET raw data series in the Series Database. In a command window, execute the command:

```
dumpDB SERIES raw
```

This command will extract the PET raw data and calibration files into the `raw` directory. The `raw` directory is created in the current directory. The full path can also be specified in the dumpDB command, e.g.:

```
dumpDB SERIES /petLists/examXXXX/raw
```

The details of the files extracted are described in the Data Organization section.

Note that there is an upper limit on the size of the raw PET sinogram that can be dumped with dumpDB. For systems circa 2016 and newer, this limit is 800 MB per frame. For all older systems this limit is 200 MB per frame. If the frame file exceeds this size, the output raw DICOM file will be created with a file size of zero bytes. If the file exceeds the maximum file size limit, there is an alternative method to extract raw files from the scanner:

- Highlight the PET raw series + Static image file
- Use the terminal command to obtain the location of the DICOM header:

```
cat $SDC_SELECTION_FILE
```

- Copy the file corresponding to the raw series (if single frame, the file will be *.1.*).
  Note that the file name/number should be the same as the RPDC.1.img file that was zero bytes. Another check is that the filename numbers should be consecutive. If there is a UID error in running the toolbox, please double-check this step.
- Copy the corresponding SINO file. One way to find the SINO location is with the DICOM header tool. Do a Ctrl+F for "SINO."

Note that dumpDB can also be used with image series in the image database.

Scanner CTAC
Highlight the CTAC series in the Series Database and type:

```
dumpDB SERIES CTAC
```

This command will extract the CTAC DICOM image files into the `CTAC` directory.

Transfer both `raw` and `CTAC` directories to a directory on the computer where the toolbox will be run.

PET/MR

Raw data
Please see "Raw data" section on the previous page under PET/CT, as the process is the same.

Scanner PIFA
For a patient acquisition, the PIFA files are temporarily stored within the directory

```
/petRDFS/PIFA/MRAC_PIFA/<sub-folder 1>/<sub-folder 2>
```

The first sub-folder is from DICOM field (0020,000E) in the associated FAT MRAC series. This information is provided for completeness; * can be used in place of this folder in the path, so this DICOM field does not need to be retrieved.

For a phantom acquisition, the PIFA files are temporarily stored within the directory

```
/petRDFS/PIFA/MRAC_PIFA/999999/<sub-folder 2>
```

The second sub-folder must be retrieved, and the DICOM field can be found in the PET raw series header and is dependent on the type of exam:

- (0002,0003)
    - static
    - non-head gated with truncation completion (TC)
    - non-head dynamic with truncation completion (TC)
    - phantom acquisition
- (0009,1013)
    - head dynamic
    - non-TC dynamic
    - head gated
    - non-TC gated

The DICOM fields, either (0002,0003) or (0009,1013), can be found by looking at the PET raw header:

- Highlight the PET raw data in the Series Database
- Sort the files in the Image Database by "Frame Location"
- Highlight the first file in the Image Database associated with this PET raw data series
  (The data files have a non-blank "Frame Location" field)
- Click on "Tools" -> "Image Header"
- In the opened header, highlight the field between brackets next to the appropriate DICOM field.
- Highlighting this field will copy the contents to memory, and the contents can be pasted clicking the middle mouse button. The user can also use right click -> copy.

Check the contents of the folder by opening a command window and entering the path, as in the following example. The most common case is for patient data, where you can use * for sub-folder 1, and sub-folder 2 from DICOM field (0002,0003) or (0009,1013):

```
ls /petRDFS/PIFA/MRAC_PIFA/*/<middle mouse button>/
```

Clicking the middle mouse button will paste the highlighted field from the header into the command line. This command will list all the PIFAs associated with the selected bed station. If the PIFAs are not present, make sure that you are checking the header of a highlighted image series for the exam.

The PIFA path should contain a pair of PIFA files. For example, a bed station that was processed as a Lungs station might contain the following two files

- `Lungs.0.1.CompPifa.dat`
- `Lungs.0.1.InVivo.dat`

The product PIFA naming convention contains the anatomy, two numbers, and PIFA type:

- The prefix describes the anatomy ID assigned to the bed station. Options include Head, Lungs, Abdomen, and Pelvis.
- The first number is for the AC method
  - o 1: Partial Head
  - o 2: Atlas
  - o 3: ZTE
- The second number is for truncation completion
  - o 0: Truncation completion off
  - o 1: Truncation completion on
- The last string of characters describes the PIFA type
  - o CompPifa: complete PIFA (contains both in-vivo and ex-vivo objects, e.g. coils)
  - o InVivo: in-vivo PIFA (contains only the patient or phantom object)

The PIFAs for phantom studies will be prefaced with "UNKNOWN."

In the event that no files can be found or that the above path does not exist (double check to make sure PET raw data file + Image series is highlighted), this means that either the exam was restored from a USB drive or that no recent reconstruction with measured attenuation correction (MAC) was performed on the product. The PIFA files are deleted within a couple weeks after reconstruction. In either case, a new whole-body MAC reconstruction must be performed to re-generate the PIFAs. This can be done using the PetRecon/Replay GUI on the console. The PIFA files for a bed position in an exam will always be placed in the same location, and reconstruction for a given exam will re-use PIFA files once they have been generated.

Once the pair of PIFAs are located, copy them to the desired location using:

```
cp PIFApath/<InvivoFile> pifaIvv_fMbN.pifa
cp PIFApath/<CompleteFile> pifa_fMbN.pifa
```

where M = bed position and N = time bin.

For multi-bed studies (patient studies only), repeat the above steps to obtain the PIFA path for each bed position and copy the PIFA files to the desired location. Take care to match the filenames with the correct bed position number.

MATLAB PIFA

To generate the PIFA in MATLAB, the MRAC image series are required (FAT, WATER, In-Phase, Out-Phase). Additionally, for phantom reconstructions and patient reconstructions with truncation completion, a TOFNAC image is required. To reconstruct the TOFNAC image on the scanner, see the Appendix for information on the TOFNAC parameters. Generating the PIFA with MRAC algorithms requires the folder duettoMracFiles to be in the MATLAB path!

Highlight each of the MRAC series, one at a time, and type:

```
dumpDB SERIES MRAC/directory
```

where `directory` is either `FAT-N,` `WATER-N,` `InPhase-N,` or `OutPhase-N` with `N` being the station number.

For example, for a two-bed exam, highlight the Fat MRAC series for station 1, and type:

    dumpDB SERIES MRAC/FAT-1

then highlight the Fat MRAC series for station 2, and type:

    dumpDB SERIES MRAC/FAT-2

Repeat the above procedures for the Water and In-Phase series (using `WATER-N` and `InPhase-N`, respectively, instead of `FAT-N` in the command line).

For a multi-bed scan, highlight each of the TOFNAC series, one at a time, and type:

    dumpDB SERIES TOFNAC-N

where `N` is the station number. Repeat the above procedure for each of the reconstructed TOFNAC image series.

If the TOFNAC images are not provided, MATLAB will generate them. However, this adds reconstruction time (two iterations of TOFOSEM).

Transfer the `raw`, `MRAC`, and all `TOFNAC-N` directories to the server directory, which should contain the directory structure described in the Data Organization section. Directory names are case-sensitive!

4. Running Duetto

Data Organization

Required Files
The required files for image reconstruction are:

- The PET raw data files (`*.RPDC.N.img`), where each file contains header information and a sinogram for the `N` frame in the acquisition (e.g. one per bed position for a whole-body study, one per gating bin for a gated study, or one per time interval for a dynamic study)
- Three calibration files (`*.RPDC.N.img`, where the `N` is a number that is one larger than the last bed frame file), containing calibration information for the normalization, well-counter correction, and geometry calibration factors.
- Information for attenuation correction:
    - For PET/CT, the CTAC images are used to generate PIFA images, which are forward projected into the attenuation correction factor sinograms (ACFs).
    - For PET/MR, the PIFA image is generated from the MRAC series images. The MR images required are `FAT-N`, `WATER-N`, `InPhase-N,` `OutPhase-N`, and optional `ZTE-N` where

N is the bed position number. More information on running the MRAC module is provided within the Attenuation Correction parameters section.

o For PET/MR, another option is to directly provide the scanner PIFA file, as described in the Transferring Scanner Data section.

In both PET/MR options, the PIFAs will be forward projected in the toolbox to create the attenuation correction factor sinograms (ACFs).

The PIFA files generated by the toolbox will be saved in the current reconstruction folder.

Organization of Files and Folders

A folder, typically labeled with the exam number, should contain the following directories and files:

Option 1 (PET/CT only):
- Directory `raw` containing the PET raw data
- Directory `CTAC` with the CT DICOM images
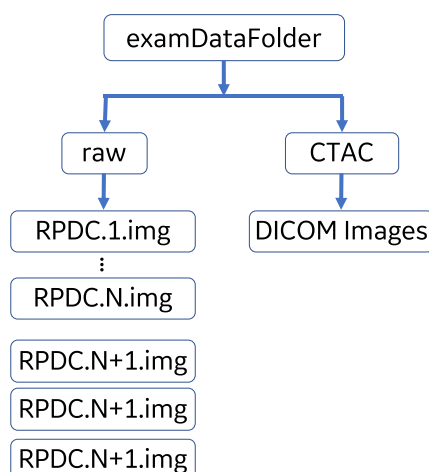
Option 2 (PET/MR, not recommended for PET/CT):
- Directory `raw` containing the PET raw data
- Complete PIFAs: `pifa_f1b1.pifa` to `pifa_fNb1.pifa`
- In vivo PIFAs (PET/MR only): `pifaIvv_f1b1.pifa` to `pifaIvv_fNb1.pifa`

PIFA files for multiple dynamic bins will be created using symbolic links within the reconstruction folder.
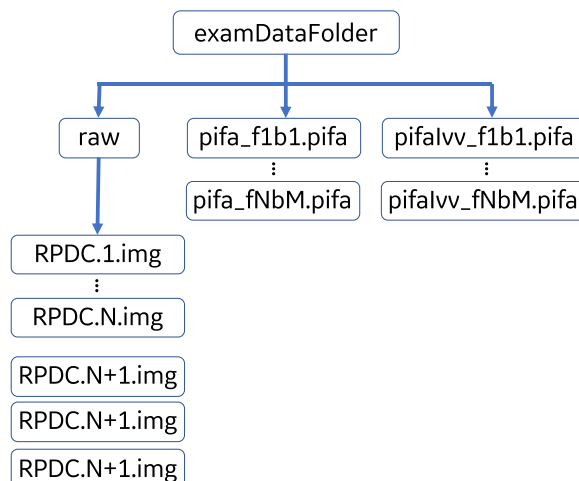
Option 3 (PET/MR only):

- Directory `raw` containing the PET raw data
- Directory `MRAC` with three sub-directories with the MR DICOM images:
    - `FAT-N`
    - `WATER-N`
    - `InPhase-N`
    - `ZTE-N` (optional)
        where `N` refers to the bed station number (the order in which the beds were acquired).
- Directory `TOFNAC-N` with PET TOFNAC DICOM images, where `N` refers to the bed station number. Only required for patient truncation completion and phantom acquisitions. Duetto will generate the TOFNAC image if not provided.
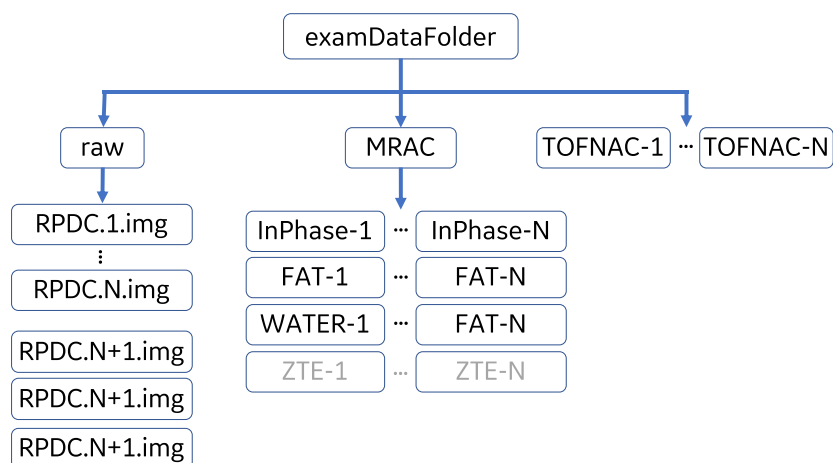
Recommendation for both PET/CT and PET/MR:

It is recommended to run each reconstruction from a new sub-directory within the examDataFolder, with a descriptive folder name of the desired reconstruction. Symbolic links can be used in the reconstruction sub-folder to point to the data files and prevent the need to duplicate large data files for every reconstruction. For example:
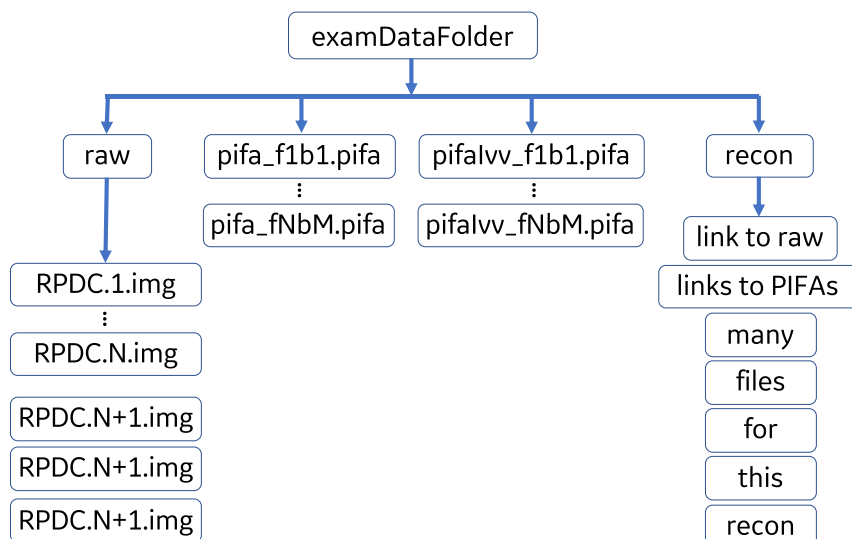
Scanner Options

The following scanners are supported on Duetto:

- SIGNA PET/MR
    - 'PETMR'
- PET/CT
    - 'D610'
    - 'D690'
    - 'D710'   (also Discovery MI DR)
    - 'COLUMBIAX', where X = 3, 4, or 5   (Discovery MI)
    - 'COMETX', where X = 2, 3, 4, or 5   (Discovery IQ)
    - 'DMI-DR'
    - 'OMNI' (Omni Legend)

The scanner name no longer needs to be defined by the user; this information is read from the data itself. However, COMET data must be flagged when setting up the configuration defaults using variable isComet; this is input to ptbUserConfig, as described in the Populate Parameter Defaults section.

Reconstruction Algorithm Options

Reconstruction algorithm options include:
- FFBP   Fourier-rebinning filtered back projection (FFBP)
- OSEM   Ordered-subsets Expectation Maximization (VPHD)
- OSEM-PSF   OSEM with point-spread-function modeling (VPHDS)

- TOFOSEM   Time-of-flight OSEM (VPFX)
- TOFOSEM-PSF   TOFOSEM with PSF modeling (VPFXS)
- BSREM   Block-sequential regularized EM (Q.Clear)
- TOFBSREM   Time-of-flight BSREM (TOF Q.Clear)

Q.Clear algorithms (BSREM and TOFBSREM) have multiple levels of reconstruction. BSREM runs a few initial iterations of OSEM, and TOFBSREM runs both OSEM and BSREM before the final TOFBSREM reconstruction. BSREM and TOFBSREM always use PSF so it is not included in the algorithm name.

Populate Parameter Defaults

User-definable parameters within Duetto are stored within the structure userConfig. The defaults are populated primarily using the specified reconstruction algorithm. The PET raw data directory, attenuation data directories, and working reconstruction directory are optional; they are assumed to be the current working directory if they are not defined.

PET/CT data acquired on the COMET scanner requires setting flag `isComet` = true;

Example #1:
```
reconAlgorithm = 'TOFOSEM-PSF';
userConfig = ptbUserConfig(reconAlgorithm);
```

Example #2:
```
reconAlgorithm = OSEM-PSF;
dataDir      = 'myDataDirectory';
petDataDir   = fullfile(dataDir,'raw');  % Optional, default=pwd
attenDataDir = fullfile(dataDir,'MRAC'); % Optional, default=pwd
workDir      = fullfile(dataDir,'myReconDirectory');
                                         % Optional, default=pwd
userConfig = ptbUserConfig(reconAlgorithm, petDataDir,
                           attenDataDir, workDir);
```

Example #3:
```
reconAlgorithm = 'BSREM';
dataDir      = 'myDataDirectory';
isComet      = true;                     % Optional, default=false;
userConfig = ptbUserConfig(reconAlgorithm, '', '', '', isComet);
```

Set User-Defined Parameters

Once the default parameters are populated in userConfig for the given scanner and reconstruction algorithm, the user can modify the fields to override the default values. A selection of parameters that can be defined are below.

General parameters:
```
userConfig.nParallelThreads % Max number of parallel threads
userConfig.workFileFormat   % Output files in .sav or .mat format
userConfig.mracExtDir       % Define full path to duettoMracFiles
userConfig.verbosity        % Set level of detail to print to screen
userConfig.filesToKeep      % Set level of intermediate files to
                              delete after reconstruction is complete
```

Specify whether corrections should be computed or not:
```
userConfig.computeAttenuationFlag % true or false
userConfig.computeRandomsFlag     % true or false
userConfig.computeScatterFlag     % true or false
```

Specify whether corrections should be applied or not:
```
userConfig.attenCorrFlag    % true or false
userConfig.randomsCorrFlag  % true or false
userConfig.scatterCorrFlag  % true or false
```
Note that scatter correction requires attenuation correction to be turned on.

Reconstruction parameters:
```
userConfig.nIterations % Number of iterations used for OSEM/TOFOSEM
userConfig.nSubsets    % Number of subsets used for OSEM/TOFOSEM
userConfig.keepIterationUpdates % Save intermediate iterations
userConfig.keepSubsetUpdates    % Save intermediate subsets

userConfig.nX          % Image matrix size (nY is auto updated)
userConfig.radialFov   % Size of the field of view (mm)
userConfig.beta        % Penalty strength for BSREM/TOFBSREM
userConfig.frames      % Specify which bed/time frames to process
userConfig.zFilter     % Specifies the [1 X 1] center kernel:
                         Light X=6, Standard X=4, Heavy X=2
userConfig.postFilterFwhm % FWHM of Gaussian post filter,
                            only applied to OSEM recons (mm)
```

DICOM options:
```
userConfig.writeDicomOutputFlag % 1 = write DICOM image
userConfig.dicomSeriesNumber    % Series number of DICOM images
userConfig.dicomSeriesDesc      % Series description of DICOM images
```

FFBP specific options:

```
userConfig.transaxialWindowType  % 'none', 'gaussian', 'hanning',
                                     'butterworth', and 'hamming' (the
                                     default method is 'none')
userConfig.transaxialWindowParam % Filter FWHM (mm). It would be
                                     adjusted if it is smaller than
                                     Nyquist sampling frequency (the
                                     Nyquist frequency is used as
                                     default)
userConfig.transaxialWindowOrder % Filter order for Butterworth only
                                     (the default value is 2)
```

Run Main Function

The corrections and reconstruction modules are called using:

```
reconImage = ptbRunRecon(userConfig);
```

1. The parameter classes are populated in `ptbInit,` which takes `userConfig` as the only input. These classes are detailed in the Parameter Classes section.

2. PET input data are parsed and unpacked from the input DICOM files in `ptbExtractPetData.` The PET raw data are extracted into files with the naming convention

    `rdf_fMbN.rdf`

    where frame (f) refers to the table position and bin (b) refers to the gating or dynamic time bin.

    A PET acquisition may take place over multiple bed positions, as in the case of a whole-body PET scan. A gated or dynamic PET scan will contain multiple time bins of data at each bed position. Gating may be performed over multiple bed positions, which will produce multiple time bins at multiple bed positions. For the different acquisition modes, data from a:

    - single static acquisition is extracted to `rdf_f1b1.rdf`
    - M-frame whole body acquisition is extracted to `rdf_f1b1.rdf` through `rdf_fMb1.rdf`
    - N-bin dynamic study, or a N-bin gated study at one table position, is extracted to `rdf_f1b1.rdf` through `rdf_f1bN.rdf`
    - M-frame whole body with N-bins of gated information is extracted to `rdf_f1b1.rdf` through `rdf_fMbN.rdf`

3. The corrections files are generated in `ptbGenCorrections`, which extracts the PET data from the raw data file (RDF), extracts calibration information, and computes the various correction files, including

    - An estimate of the detection probability for each line of response (LOR) in the data set, which is a combination of a count rate-independent portion (often referred to as "normalization") and a count rate-dependent portion ("deadtime").
        o The normalization file saved by Duetto includes the crystal efficiencies and geometry calibration factors.

- o   The deadtime/pile-up correction file saved by Duetto includes effects of photon pile-up in each detector crystal.

- An estimate of the random coincidences recorded on each LOR. The randoms are estimated from the singles.

- An estimate of the scattered coincidences recorded on each LOR, and for TOF imaging also in each time bin.

- An estimate of the attenuation losses for each line of response, which may be based on CT or MR image data.

PET/MR only: The corrections function also evaluates the data to determine if "Resilient Reconstruction" (or Fault Tolerant Reconstruction (FTR)) should be enabled.

The detector response or point spread function (PSF) is another correction that can be applied in the reconstruction, and this information is read from a system-dependent MATLAB.mat file. See the Appendix for details.

4. The reconstruction algorithm is called using `ptbPetRecon`.

The size of the reconstructed image will vary depending on the type of reconstruction:

- For a single- or multi-frame acquisition: the final 3D voxel array (x,y,z) representing the PET volume image is written to a file named `ir3d.sav`. For a multi-frame study, the image is stitched together from overlapping individual frame images and written to `ir3d.sav`.
- For a dynamic or gated study: a 4D array (x,y,z,n) representing a set of PET volume images, with one image for each frame of a dynamic study or one phase of a gated study.

After reconstruction, the overlapping frames of a whole-body study are combined, and, if requested, a post-reconstruction image filter is applied to the images.

The output image can also be written as a series of DICOM files, where there is one DICOM file for each (x,y) plane in the reconstructed PET image (i.e. one file per value of z in a static image volume, or one file per (z,n) pair in a dynamic or gated series). DICOM images are written to directory titled

    `duetto_<algorithm>_<FOV>_<nX>_<nIterations>`

with an extension of .sdcopen (not .dcm).

The PET image volume can be viewed using the function call:

    `dispOrthoSlices(reconImage)` or `ximage(reconImage)`

5. Lastly, there is an option to automatically clean-up intermediate files after the reconstruction is finished. The types of files to delete after running the reconstruction can be specified using

    `userConfig.filesToKeep`

where the value can be set to:

- PtbVerboseEnum.OFF   = Keep all files
- PtbVerboseEnum.VERBOSE   = Delete only extracted data (rdf, prompts, wcc)
- PtbVerboseEnum.DETAILED   = Delete intermediate files (norm, keyhole, intermediate images)
- PtbVerboseEnum.CONCISE   = Keep only final image, scatter, attenuation
- PtbVerboseEnum.TERSE   = Keep only final frames and stitched, post-processed image

Examples

Two example scripts and an example function can be found in the Duetto directory runScripts.

5. Output Files

Intermediate Files

Data extracted from the raw data files and the computed or estimated corrections files are saved as intermediate files. These include well-counter correction (WCC), normalization, deadtime/pile-up, randoms, attenuation, and scatter.

Many of these data (nonTOF) are saved as a sinogram matrix corresponding to the lines of response defined in the GE-standard PET detector coordinate system. Therefore, it is convenient to store them in the same orientation as a volumetric PET data set, which is as a three-dimensional array $p(u, v\theta, \phi)$, where u and v represent the transaxial and axial distance coordinates of the line of response, respectively, and ϕ and θ are the circumferential and axial angular coordinates of the LOR (the v and θ coordinates are combined because in a cylindrical PET geometry there are fewer available values of v as $|\theta|$ increases).

These arrays can be stored in "savefile" or MATLAB `.mat` file formats. The file naming convention for these files is of the form `filename_fXbX.sav`, where f = frame number (bed position) and b = bin number (time frame). It is noted that this naming convention may be confusing and opposite from intuition.

TOF files are saved in smaller intermediate file formats (instead of the full 4D matrix). TOF prompts are always saved in `.mat` format, as sparse files are nicely compressed in the MATLAB file format. The TOF scatter estimate is saved in a down-sampled form, as the full 4D matrix is memory-intensive, and up-sampled during the reconstruction loop. The TOF scatter estimate is saved in a 5D format, [crystal 1, radial bin, view, crystal 2], and can be up-sampled using the function `ptbUpSampleTofScatterSinogram.m`.

The default filenames for the intermediate files are defined in the function `PtbFrameFilenames.m`.

Functions to Read and Write Intermediate Files

.sav and .mat Files

The savefile `.sav` format can be read and written using functions `ptbReadSaveFile` and `ptbWriteSaveFile`.

Either savefile `.sav` or MATLAB `.mat` formats can be read and written using ptbReadFile and ptbWriteFile. Example function calls are:

```
data = ptbReadFile(filename, fileType);
ptbWriteFile(data, 'filename', fileType);
```

where filetype is "`mat`" or "`sav`" (with or without a "`.`" works).

.pifa Files

PIFAs can be read and written using

```
PIFA = ptbReadPifa('filename')
```

```
ptbWritePifa('filename',PIFA)
```

The PIFA variable is a structure with header information and the 3D PIFA matrix in the data field. Note that the unit of the PIFA files is inverse millimeters (1/mm). The attenuation of water will therefore be 0.096 1/mm.

DICOM Files

DICOM image data can be read using `ptbReadDicom`, and DICOM header information can be read using

```
dicominfo('dicom.img','dictionary','pet-dicom-dict.txt')
```

Additionally, the pet-dicom-dict.txt file is a text file and can be opened and searched for specific DICOM fields.

DICOM data can be written using

```
ptbWriteDicom(outDir, img3d, sliceLoc, dicomHdr, generalParams)
```

where `outDir` is the output directory; `img3d`, `sliceLoc`, and `dicomHdr` were originally obtained from `ptbReadDicom`; and `generalParams` is a structure that at a minimum contains the verbosity field, defined by `PtbVerboseEnum`.

Reconstructed Image Files

Reconstructed images from different stages of the reconstruction process are written to disk:

1. There is the option to write out an intermediate image at the end of each subset update and/or each iteration update. These images are written out in the format `ir3d_fXbX_lX_itXofX.sav`, where `f` = frame number (bed position) and `b` = bin number (time frame) , `l` = reconstruction level, and `it` = iteration.
2. After a number of scaling corrections are applied (such as WCC), the image is written out as `ir3d_fXbX.sav`.
3. The images of a multi-bed study are then stitched together and post-filtered (if applicable, only OSEM algorithms). The final image is saved to disk as `ir3d.sav`.

6. Parallel Processing

Parallel processing can be used if the MATLAB parallel processing toolbox is available.

Parallel processing is implemented by calling the built-in MATLAB function `parfor` in the following places:

- Extracting prompts data: data will be extracted in parallel for either one or multiple frames and bins.
- Estimation of the scatter correction
- Image reconstruction: reconstructions will be performed in parallel if multiple frames or bins.

Further parallel processing will be added as development time allows.

The user should specify the maximum number of parallel processing threads (to prevent saturating the memory) using `userConfig.nParallelThreads`. The user can force parallel processing to be off by setting this value to only 1 thread (0 also works).

For more information on Parallel processing and the Parallel Computing Toolbox, please visit: https://www.mathworks.com/products/parallel-computing.html

7. Advanced Options

PSF Modeling Options

The default PSF settings are based on the scanner and are modelled in sinogram space:

- sinoRadialFlag = true
- sinoRadialFilename = ''detectorResponseMatrix.<scanner name>.mat',
  stored in the `corrections/psf` folder.
- sinoAxialFlag = true;
- sinoAxialFactor = 3.7 or 4, depending on the scanner,
  where the value is the central weight of a 3-point kernel

PSF can also be modeled in image space, with these options:

- imageTransaxialFlag
- imageTransaxialFwhm_mm
- imageAxialFlag
- imageAxialFactor

Image-based PSF implementation is currently spatially invariant, whereas the sinogram radial kernel is spatially variant.

Hybrid PSF can also be implemented by specifying a combination of the sinogram and image-based parameters. The above parameters can be set using userConfig.<property name>.

Detector PSF Details

A standard shift-invariant 1D convolution can be expressed by a single vector, the impulse response. However, the sinogram-based PSF implementation is shift-varying, so a different smoothing kernel is applied for each radial position in the 1D projections. The PSF kernels are stored in the `corrections/psf` folder.

The size of the matrix is [`nRadialBins nRadialBins nCrystalsPerBlock`]. Using PETMR as an example, the size of the matrix is 357 x 357 x 16.

The 357 x 357 matrix defines a shift-varying smoothing kernel. Each row/column of the matrix holds a kernel centered on the diagonal. The center of the 357 x 357 matrix shows sharp kernels, while the outside shows blurry kernels. The increasing width of the kernels from the center to edge of the field of view is due to the position-dependent effect of depth of interaction. Note also that the rows of the 357 x 357 matrix sum to 1, so the counts are distributed differently, but the kernel is count preserving.

Just after forward projection, for a single projection, the PSF smoothing is applied:

```
filteredProj = detResponseConvMtx(:,:,k) * unfilteredProj;
```

where k is set to the appropriate kernel (explained below), and "`unfilteredProj`" is a vector of size 357 x 1. Before backprojection, the transpose of `detResponseConvMtx` is applied.

The sinogram projections are defined such that moving from one projection to the next moves around the ring one crystal at a time. For modules that have 16 crystals radially, many geometric components of the system repeat every 16 projection angles. In fact, the geometric calibration is saved with just 16 projection angles because it repeats. (The entire PET ring rotated by 1 module, which is 16 crystals, gives the same exact PET ring.)

Similarly, only 16 projections are needed for the PSF kernel:

> The 1st projection angle corresponds to `detResponseConvMtx(:,:,1)`

> The 2nd projection angle corresponds to `detResponseConvMtx(:,:,2)`

> …

> The 16th projection angle corresponds to `detResponseConvMtx(:,:,16)`

> The 17th projection angle corresponds to `detResponseConvMtx(:,:,1)`

> …

> etc.

User-Generated PseudoCT

During the course of the PIFA generation process for PET/MR, a pseudo-CT DICOM series is created for each bed station. The pseudo-CT is an image volume in Hounsfield Units that mimics CT data. The pseudo-CT is derived from the acquired MRAC series and is obtained by calling the `ptbMr2PseudoCt` function.

It is possible to use a user-generated (or modified) pseudo-CT by setting

> `userConfig.pseudoCtFlag = 0;`

This option will force the toolbox to bypass the call to `ptbMr2PseudoCt`, and the user-specified pseudo-CT folder(s) will be used in the generation of PIFA files.

The user-specified pseudoCT folders should be of the format

> `pseudoCT-N`, where N is the bed station number

The user must also specify the anatomy IDs for each bed position using

> `userConfig.anatomyIdManual = ['H','L','A','P'];`

In this example, there are four bed positions, containing Head, Lungs, Abdomen, and Pelvis.

The easiest way to provide a user-generated pseudo-CT DICOM image series is to start from an already existing toolbox-generated pseudo-CT and then modify it. The reason for this is that the necessary data orientation, format, and header information would be available for the new pseudo-CT DICOM image.

To obtain a set of toolbox-generated pseudo-CT DICOM image series for each bed station, run a full reconstruction with attenuation correction 'on'. The resulting pseudo-CT series can then be modified by using the `ptbReadDicom` and `ptbWriteDicom` functions.

Note: If any PIFA files (ending in `.pifa`) exist in the directory, then Duetto will skip the PIFA generation algorithm, and the existing PIFAs will be used instead in the generation of the attenuation correction files (acf).

8. Parameter Classes

Variables are divided into a number of parameter classes, which are briefly described here with a selection of the parameter fields within each structure.

Many of these parameters can be modified by the user by simply adding the field and parameter value to the userConfig structure, as described in the section Set User-Defined Parameters. The value will automatically be populated into any parameter class that contains that field.

frameStats

frameStats is populated from information in the PET DICOM header, and fields are not modifiable. Fields include:

- kFrame: current frame number
- kBin: current bin number
- nTotalFrames
- nTotalBins
- isotope
- halfLife
- positronFraction
- patientWeight
- patientPosition
- patientEntry: 0 = head first, 1 = feet first
- vqcTableShift, additional vqcYoffset due to table deflection
- examData: rdf.sharc_rdf_pet_exam_data;
- rdfConfig: rdf.sharc_rdf_config
- sysGeoData: rdf.sharc_rdf_sys_geo_data;
- acqStatsData: rdf.sharc_rdf_acq_params_data
- landmarkParams: rdf.sharc_rdf_acq_landmark_params
- scanParams: rdf.sharc_rdf_acq_scan_params
- edcatParams: rdf.sharc_rdf_acq_edcat_params
- gatedParams: rdf.sharc_rdf_acq_rx_gated_params
- acqStatsData: rdf.sharc_rdf_acq_stats_data
- deadTimeEventsData: rdf.sharc_rdf_deadtime_header.sharc_rdf_deadtime_events_data
- unitIntegDeadTime: rdf.unitIntegDeadTime;
- unitMuxDeadTime: rdf.unitMuxDeadTime;
- scanStartTime
- frameStartTime
- frameDuration
- decayFactor: note that images are decay-corrected back to the beginning of the exam
- rawDicomFilename
- rawDicomFrameStartTime
- rawDicomFrameNumber

- baseNames: populated based on frameFiles,a structure containing the names of the various data arrays required for reconstruction, named using the filecontents_fXbX.sav naming convention. These filenames can be modified to a user-specified name.

generalParams

The fields in generalParams are not module-specific. Fields include:

- scannerName: extracted from PET raw data
- workDir: input from the user, or assumed to be current working directory
- petDataDir: input from the user, or assumed to be current working directory
- attenDataDir: input from the user, or assumed to be current working directory
- nParallelThreads: maximum number of parallel recon threads
- workFileFormat: file formats in which intermediate files are saved ('mat' or 'sav')
- verbosity: how much detail is printed out to screen
- filesToKeep: option to automatically clean-up files after the reconstruction is complete

scanner

The fields in scanner are populated based on the scanner name, and they include definition of system and detector geometries. It is not advised, nor readily possible, to modify any of these parameters.

sinoParams

sinoParams contains details on the sinogram, and the parameters are not modifiable. Fields include:

- nU: number of radial bins
- nV: number of direct+oblique slices
- nPhi: number of views
- nZ: number of direct slices
- sV: size of direct slices (mm)
- radialRepositionFlag: whether data is in native geometry or radial repositioned space
- nUrr: number of radial bins in radial repositioned space
- nT: number of timing bins
- dT: size of each timing bin (ps)

corrParams

The correction parameters structure contains many sub-structures with parameters for each of the correction types.

The following flags (set to true or false) determine if the corrections are generated. If the correction files already exist, they will not be regenerated. The exception is if the attenuation correction files (ACFs) exist, but the PIFA files do not, the ACFs will be deleted and regenerated after new PIFA files are generated.

- computeNormFlag
- computeDtPucFlag
- computeAttenuationFlag
- computeScatterFlag

The user can use a different correction algorithm from the GE default by specifying the function name for that correction in any of the following variables:

- randomsUserFunc
- normUserFunc
- dtPucUserFunc
- attenUserFunc
- scatterUserFunc

Randoms (randomsParams)

Randoms are computed from the singles data, based on the publication:

> C.W. Stearns ; D.L. McDaniel ; S.G. Kohlmyer ; P.R. Arul ; B.P. Geiser ; V. Shanmugam. Random coincidence estimation from single event rates on the Discovery ST PET/CT scanner. IEEE NSS-MIC Conference Record 2003.

Normalization (normParams)

The normalization correction file includes the crystal efficiencies and geometry calibration factors. The parameters associated with this correction are not modifiable.

Deadtime/Pileup (dtPucParams)

The deadtime/pile-up correction is calculated based on pile-up correction crystal factors in the PET raw data and known detector characteristics. The parameters associated with this correction are not modifiable.

Attenuation (attenParams)

The attenuation parameters are assigned from the CTAC or MRAC parameters.

ctacParams

- attenDataDir: full path to CTAC DICOM images

MRAC

- attenDataDir: full path to MRAC DICOM images
- mracExtDir: full path to the location of MRAC executables and templates
- mracExtInfo: filenames of executables, phantom templates, non-patient AC images, config files
- pseudoCtDirBaseName: base directory name for pseudoCT-n = 'pseudoCT'. This directory will be created if pseudoCtFlag=1, it will be read in if pseudoCTflag=0

- pseudoCtFlag: run pseudoCT algorithm = 1 (default), use user-generated pseudoCT = 0; if using user-generated pseudoCT, user must define anatomyIdManual for each bed
- anatomyIdManual: options are H, L, A, P. Determined from AnatomyBoundaries, user can assign to override. Format is as one string, e.g. 'PPALL'.
- anatomyType: options are 'Head', 'Nhead' (HNU coil), 'Lungs', 'Abdomen', 'Pelvis'
- anatomyBoundaries: vector of SI locations of the bottom of the head, lung, and abdomen boundaries (mm), this is read from the PET DICOM header, but user can override
- tofNacDirBaseName: base name for TOFNAC-N directory (default = 'TOFNAC')
- tofnacFlag: generate TOFNAC image = 1, use user-supplied TOFNAC = 0; if user-supplied TOFNAC does not exist, then generate TOFNAC image. Note generating the TOFNAC image adds two iterations of TOFOSEM to the reconstruction.
- headMracMethodFlag: Head Atlas = 1, Partial Head = 2, ZTE = 3
- chunkwise: define a single anatomy ID per bed = 0, allow multiple anatomy IDs = 1 (default)
- truncComplete: perform truncation completion on body bed positions = 1 (default)
- fatWaterSepMode: continuous fat-water AC values = 'cont', four discrete values between fat-water = '4seg'
- jointEstimationFlag: not yet implemented in Duetto

Scatter (scatterParams)

Scatter parameters are populated into the structure scatterParams.

- edgeTailScalingLimit (default = 2.5)
- centerTailScalingLimit (default = 1.5)
- multipleScatterMode: 'FixedKernel' or 'ObjectDependentKernel' (default)
- calcDetEffFlag: detector efficiencies as a function of energy window; read from a file for 425-650 keV energy window = 0 (default), analytically calculate = 1

Resilient Recon/FTR (ftrParams)

Resilient reconstruction parameters are populated automatically based on the FTR detection algorithm.

reconParams

Reconstruction parameters are populated into the structure reconParams.

General reconstruction parameters:
- outputFilename: prefix of the intermediate and final reconstruction images (default = 'ir3d')
- frameOutputFilename: populated based on the working directory path and outputFilename
- outputFormat: file format for reconstructed images, 'mat' or 'sav' (default)
- reconFunction: populated based on input reconstruction algorithm
- tofFlag: populated based on the recon algorithm name
- nX: number of x and y image voxels
- radialFov: radial field of view (mm)

- lTarget: LR offset for off-centered reconstruction (negative = left)
- pTarget: AP offset for off-centered reconstruction (negative = anterior)
- nIterations: default is populated based on the algorithm; the value can only be overwritten for OSEM algorithms (with 1 recon level)
- startIteration: not yet implemented
- nSubsets: default is populated based on the scanner, can be overwritten
- subsetSelectionScheme: how to select subsets, 'contiguous' or 'distributed' (default)
- subsetAngleOffset: number of views by which to shift subset selection
- keepSubsetUpdates: keep intermediate subset update images (true or false)
- keepIterationUpdates: keep intermediate iteration update images (true or false)
- saveSensitivityImages: save sensitivity images calculated in TOF algorithms (true or false), as it is time-consuming to calculate these
- useSavedSensitivityImages: read in saved sensitivity images calculated in TOF algorithms (true or false), as it is time-consuming to calculate these
- fwdProjFunc: forward projector function name (default = 'ptbFdd')
- backProjFunc: back projector function name (default = 'ptbBdd')

Correction options (reconParams.corrOptions), options for all are true or false:
- decayCorrFlag
- durationCorrFlag
- randomsCorrFlag
- scatterCorrFlag
- attenCorrFlag
- normDtPucCorrFlag

PSF options (reconParams.corrOptions.psfOptions), are described in more detail above:
- sinoRadialFlag
- sinoRadialFilename
- sinoAxialFlag
- sinoAxialFactor
- imageTransaxialFlag
- imageTransaxialFwhm_mm
- imageAxialFlag
- imageAxialFactor

Parameters regarding reconstruction levels:
- nReconLevels: number of reconstruction levels. The image at the end of each level is used as the initial image in the next reconstruction level.
- reconLevels: description of each of the recon levels and parameters (iterations & subsets)
- levelIndex: current reconstruction level
- algorithmParams: populated based on the current recon level parameters
- initialImageFile: the user-input initial image filename should be of the form <initialImageFile>_fXbX.<outputFormat>   If not specified, the initial image is initialized as all

ones for the first recon level. The initial image for remaining recon levels is the last image from the previous recon level.

Post-processing parameters:

- postFilterFwhm: transaxial Gaussian post filter (mm), OSEM algorithms only
- zFilter: axial post filter, implemented as [1 N 1], OSEM algorithms only
- frameOverlap: number of slices overlap in the acquired bed positions
- writeDicomOutputFlag: write final reconstructed PET image to DICOM file (true or false)
- dicomSeriesNumber: assign a series number to the DICOM image (default = 801)
- dicomSeriesDesc:name of the DICOM series, pre-pended with "NON-DIAGNOSTIC'

Example to set 3 reconstruction levels:

```
userConfig.reconLevelList(1).reconAlgorithm =
     PtbReconAlgorithmEnum.OSEM;
userConfig.reconLevelList(1).reconFunction  = 'ptbOsem';
userConfig.reconLevelList(1).nIterations = 3;
userConfig.reconLevelList(1).nSubsets    = 28/4;

userConfig.reconLevelList(2).reconAlgorithm =
     PtbReconAlgorithmEnum.OSEM
userConfig.reconLevelList(2).reconFunction  = 'ptbTofOsem';
userConfig.reconLevelList(2).nIterations = 6;
userConfig.reconLevelList(2).nSubsets    = 28/2;

userConfig.reconLevelList(3).reconAlgorithm =
     PtbReconAlgorithmEnum.BSREM';
userConfig.reconLevelList(3).reconFunction  = 'ptbTofBsrem';
userConfig.reconLevelList(3).nIterations = 10;
userConfig.reconLevelList(3).nSubsets    = 28;

userConfig.psfFlag = true;
```

Note that reconParams.reconLevels.tofFlag will be set automatically based on the reconFunction.

If setting multiple levels, make sure:

- to remove reconLevelList field before defining user-defined reconLevelList.
- that none of the above fields exist outside of reconLevelList in userConfig. Fields can be removed using rmfield. For example:

```
fields = {'reconFunction','reconAlgorithm','nIterations'};
userConfig = rmfield(userConfig,fieldsToRemove);
```

keyholeParams

keyholeParams is used for targeted image reconstruction, where the reconstruction FOV is smaller than the maximum FOV. It is very similar to reconParams, with a few automatically modified parameters.

9. Unlisting List-mode Data

List data can be unlisted into sinograms into one frame (static exam) or multiple frames (dynamic or gated acquisition). Dynamic unlisting will divide the list file into time bins based on start and end time vectors. (Note that currently these time bins can NOT overlap.). Gated unlisting will sort the list events into a given number of gated bins, either by time or percent as shown in the examples below.

The steps for performing data unlisting and reconstructing are as follows:

1. Required files for unlisting and reconstructing can be dumped by highlighting the list series in the image database and executing the command:

   ```
   dumpDB SERIES LIST
   ```

2. For raw data files version 9 (RDFv9, e.g. PETMR list data), the LIST000N.BLF files must be decompressed before transferring from the scanner by executing the command (only works on the scanner):

   ```
   unglepl LIST000N.BLF
   ```

   The original list file will be replaced with the uncompressed list file; a new file will not be generated.
   To recompress the list file, execute the command (only works on the scanner):

   ```
   glepl LIST000N.BLF
   ```

   For Linux users, the RDFv10 list files do not need to be decompressed if the decompression filters are correctly set by running `start_toolbox_MATLAB.sh` before MATLAB is launched. For other OS users, the list files need to be decompressed or recompressed using GZIP on a scanner by executing the commands below:

   ```
   rdfRepack -r NONE LIST000N.BLF <output filename>
   ```

   ```
   rdfRepack -r GZIP LIST000N.BLF <output filename>
   ```

   [WARNING] The RDFv10 decompression can NOT be reverted and the modified files will not work on the scanner. Please make sure the original files are saved separately.

   [NOTE] `rdfRepack` can be used for decompressing both list and raw RDFv10 data.

3. The unlister requires the files:
   - `LIST0000.BLF to LIST000N.BLF`, where `N` corresponds to the frame number - 1
   - `RPDC.1.img to RPDC.N.img`, where `N` corresponds to the frame number

   Library errors may be solved by recompiling the mex files using the "`mexUnlister`" command.

4. An example MATLAB script is `runScript_unlister.m` in the unlist directory, along with examples below.

5. The output from the unlister will be a new directory, titled as unlist-<time-stamp>, with three files:
   - `i100000.RPDC.N`: header information copied from RPDC.X.img files

- `SINO000N:` the sinogram
- `unlistRx-bedN.txt:` parameters defined in rx structure and used in the unlisting

where `N` corresponds to the frame number (numbering starts from 0).

6. After unlisting, when performing image reconstruction in the toolbox, the three correction files must also be present (RPDC.N.img, where N is the number of frames + 1). As long as these files are located in the 'raw' folder (or there are valid symbolic links), the toolbox will know how to read it.

The SIGNA PET/MR has the option to write energy-mode list files. These files are 8-bytes per event instead of 6-bytes per event. Energy-mode list files are NOT saved by default. Scanner configuration files must be modified prior to acquisition; contact the GE Engineering team for more information. To unlist these files, the `listmodeWithEnergyFlag` must be set to true, and the energy limits `lowEnergyLim` and `highEnergyLim` must be specified.

Notes:
- Time vectors are in <u>milliseconds</u>.
- startMsecVec is with respect to the first timestamp in the list file. For example, if start on count rate is enabled, and the acquisition begins at 10 seconds, startMsecVec = 0 will actually start 10 seconds into the list file.
- Here are the criteria for setting up startMsecVec and endMsecVec:
    o startMsecVec and endMsecVec need to have the same total number of elements.
    o Time intervals for each bin are positive or zero so startMsecVec(i) <= endMsecVec(i)
    o Time intervals for each bin do not overlap so endMsec(i-1) <= startMsecVec(i).
    o The first startMsecVec needs to be the smallest/earliest time point in startMsecVec, and the last endMsecVec needs to be the largest one in the endMsecVec as the startMsecVec(1) and endMsecVec(end) would be used to define the total frame duration (not the bin duration).
- For gated unlisting, option 3, the % offset is with respect to the triggers in the list file (these are often defined to be 30% after peak inspiration; confirm the acquisition settings of your dataset).

Static

```
rx.listFilePath = '~/LST/LIST0000.BLF';

% Energy mode unlisting (if energy-mode list-file)
rx.listmodeWithEnergyFlag = 0;  % Set = 1 if want to use energy limits below
rx.lowEnergyLim  = 425;         % [keV]
rx.highEnergyLim = 650;         % [keV]

% Unlist parameters
rx.tofMode = 'tof';         % Options: nontof, tof
rx.unlistType = 'static';   % Options: static, dynamic, gated
rx.startMsecVec = 0;        % Unlist 6 minutes into 1 frame [msec]
rx.endMsecVec  = 360 * 1000;

% Call main function
data = UnlistMain(rx);
```

Dynamic

```
rx.listFilePath = '~/LST/LIST0000.BLF';

% Unlist parameters
rx.tofMode = 'tof';          % Options: nontof, tof
rx.unlistType = 'dynamic';   % Options: static, dynamic, gated
rx.startMsecVec = [0:30:330]' * 1000;  % Bin every 30 seconds for 6 minutes
rx.endMsecVec   = [30:30:360]' * 1000;  % [msec]

% Call main function
data = UnlistMain(rx);
```

Gated

Option 1

```
rx.listFilePath = '/localdata/list/LIST0000.BLF';

rx.unlistType = 'gated';
rx.tofMode    = 'nontof';

% To generate startMsecVec and endMsecVec values,
% each start/end row vector specifies one gated frame
% zero msec value corresponds to first time marker in list file (ref:
/HeaderData/ListHeader/firstTmAbsTimeStamp)
rx.startMsecVec = [ ...
     412  4413  8413 12414 16415 20416 24417 28418 32419 36419 40420 44421;
     712  4713  8713 12714 16715 20716 24717 28718 32719 36719 40720 44721;
    1012  5013  9013 13014 17015 21016 25017 29018 33019 37019 41020 45021;
    1312  5313  9313 13314 17315 21316 25317 29318 33319 37319 41320 45321];

rx.endMsecVec   = [ ...
     712  4713  8713 12714 16715 20716 24717 28718 32719 36719 40720 44721;
    1012  5013  9013 13014 17015 21016 25017 29018 33019 37019 41020 45021;
    1312  5313  9313 13314 17315 21316 25317 29318 33319 37319 41320 45321;
    1612  5613  9613 13614 17615 21616 25617 29618 33619 37619 41620 45621];

rx.gatedBinMode = 'bypass';
% 'bypass' implies no trigger rejection.
% In this case, caller is taking responsibility for the time intervals
% (no checking is done).

% [ numGatedFrames, numTimeIntervals ] = size(startMsecVec);

UnlistMain(rx);
```

Option 2

```
% All msec values are zero relative to the first timestamp in the list file
% Ref: HDF5 list file at '/HeaderData/ListHeader/firstTmAbsTimeStamp'
rx.startMsecVec = [ ...
     412  4413  8413 12414 16415 20416 24417 28418 32419 36419 40420 44421];
```

```
rx.endMsecVec = [ ...
    4413  8413 12414 16415 20416 24417 28418 32419 36419 40420 44421 48422];

rx.gatedBinMode = 'percent';
% gated phases reside at given percentages of each trigger interval

rx.gatedBinVec = [ 25.0 25.0 25.0 25.0 ];   % gated bin percentages
```
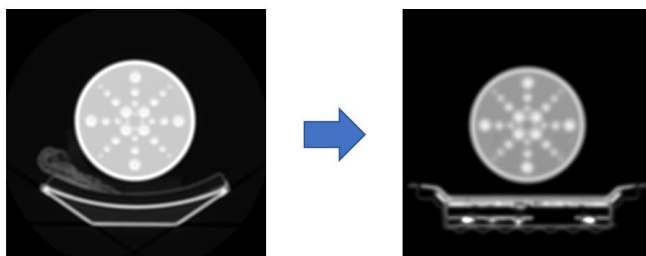
Option 3
```
rx.gatedBinMode = 'sppb';        % single phase percent binning
rx.gatedBinVec = [ 10.0 50.0 ];  % offset & width of single phase binning
```

Option 4
```
rx.gatedBinMode = 'time';             % gated with fixed time interval
rx.gatedBinVec = [ 300 300 300 300 ]; % 4 bins with 300 msecs in each bin
```

10. CT to PIFA Helper Tool

This experimental tool with a graphical user interface (GUI) has been developed to enable the attenuation correction (AC) of a PET phantom that does not have a built-in template of the attenuation map in the SIGNA PET/MR. In order to perform AC with the phantom, CT DICOM images must be acquired, and then they must be converted into an attenuation image (as PIFA format), which Duetto uses during the PET reconstruction.

Acquired CT image   Converted to TOFNAC

Requirements

- Duetto in MATLAB R2018b or newer
- Files for the offline PET image reconstruction with Duetto (see "Organization of Files and Folders" in Chapter 4 for options and details)
  - PET RAW sinogram data and correction files (RPDC.*.img)
  - TOFNAC 128x128 (optional, if not present, Duetto will auto-generate them)
  - System generated PIFA (in case of Option 2 in Organization of Files and Folders section)
  - MRI to perform "MRAC" (in case of Option 3 in Organization of Files and Folders section)
- CT DICOM
  - Set up a phantom precisely the same as PET/MR scan, in Head-First-Supine orientation (HFS)
  - Use of a hard, flat tabletop and/or supporting legs are recommended

- o Use a fairly high kVp and mA, e.g. 120 kVp/300 mA
- o Z-axis coverage must be greater than 250 mm, i.e. the length of one PET/MR bed
- o Reconstructed slice interval must be 2.78 mm, i.e. the interval of PET slices in PET/MR
- o FOV = 60cm is recommended, but smaller FOV can be readjusted
- o Image matrix = 128x128 is recommended
- o Reasonable slice thickness to depict the structural detail e.g. 2.0 mm

It is recommended that PET/MR scan be performed with the "LaBestia" phantom selected as the built-in template. It may be possible to instead choose an anatomy, e.g. pelvis, and proceed with MRAC; however, the resulting MRAC series (Water, Fat, and In-Phase with 2-point Dixon scan) could be full of artifacts, resulting in a PIFA that is difficult to manipulate with the Phantom CTAC Tool.

Before running this tool, run a PET image reconstruction for the appropriate PET/MR dataset in Duetto. **Make sure that PET image reconstruction completes without an error**. The two PIFA files (*pifa_f1b1.pifa* and *pifaIvv_f1b1.pifa*) in the main "examDataFolder" will be replaced by this tool.

Creating PIFA from CT DICOM

Run the script **runScript_ctacToPifa** in MATLAB to start the Phantom CTAC Tool. The folder "SampleData" contains TOFNAC and CT images of a phantom as well as the original PIFA files that can be used to familiarize yourself with the tool.

STEP 1: Select 89 slices of Axial CT images

The file selection dialog box will pop up to import TOFNAC images followed by CT DICOM images. Follow the instructions and select each folder to load images inside. The GUI window will show the TOFNAC and Axial CT side by side, as shown below. Find a matching pair of TOFNAC and CT by flipping through slices using the pull-down menus. Once you have found the identical pair, click "ACCEPT" and 89 slices that are needed for AC would be automatically selected for the next step.

Then you will be asked to type in the FOV as well as the peak voltage of these CT images in the MATLAB command window.
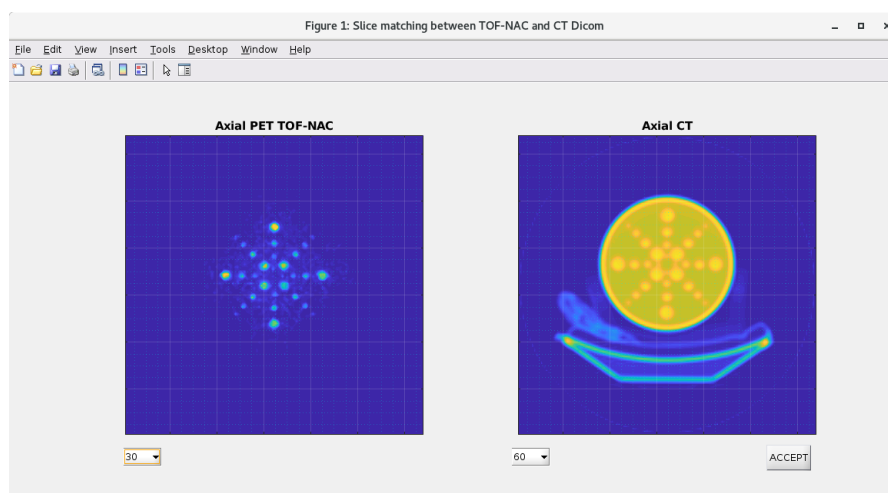
## STEP 2: Mask only the scanned object in PET/MR

A GUI will appear showing an axial CT slice (starting with slice #1). You need to create a mask around the scanned object by clicking the outline. Then right-click on the closed mask and select "Create Mask" (or double-click to skip the menu box). An example is shown below. The GUI will then progress through each axial CT slice, and you must repeat the mask process for all 89 slices.
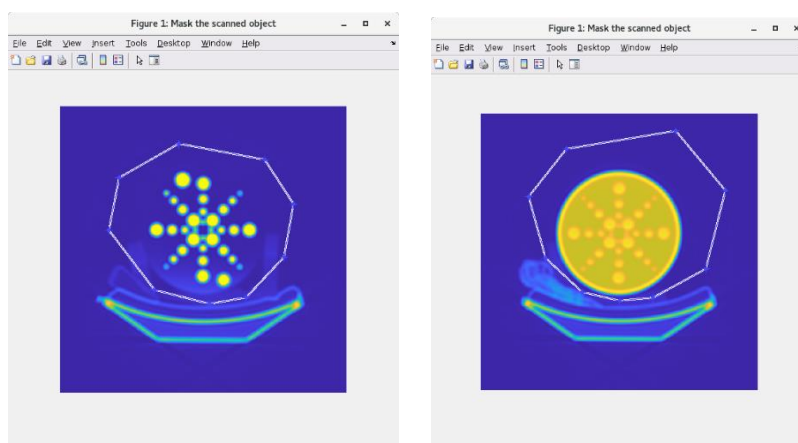
STEP 3: Manually adjust the in-plane alignment of TOFNAC and CT

The GUI shows TOFNAC PET and the masked CT images side by side. Select an axial slice that has good features for matching the location between TOFNAC and CT images. Then adjust the in-plane alignment by shifting CT image. The images can be shifted left/right and up/down by the +1 and +10 slice buttons next to the CT image. You can reset the shifts by clicking "RESET." When the CT and TOFNAC images are aligned, click "ACCEPT" to move to the next step. This alignment only needs to be done for one of the slices (not all of the slices like in the previous step).

CT images will be converted to mu-values using the internal lookup table provided in Duetto.
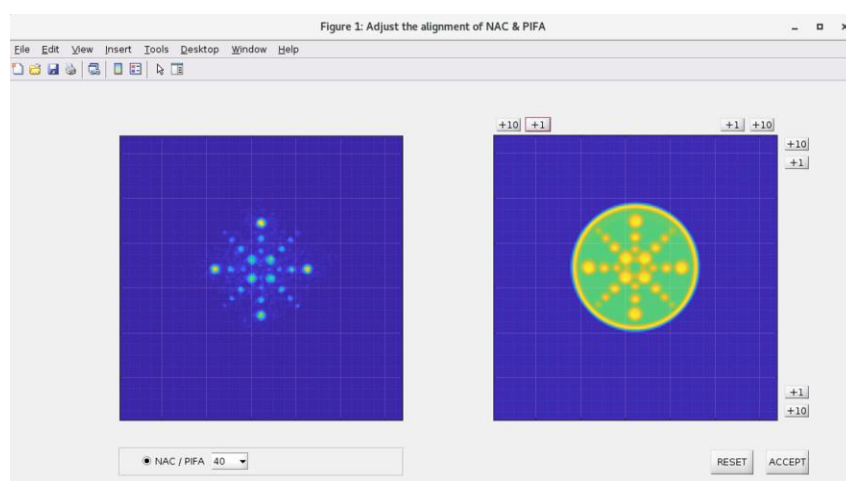
STEP 4: Prepare new PIFAs

The file selection dialog box pops up in order to select a pair of PIFA files to be duplicated and overwritten. Follow the instructions and first select in-vivo PIFA e.g. *pifaIvv_f1b1.pifa* followed by the complete PIFA with hardware e.g. *pifa_f1b1.pifa*. The new PIFA files based on CT DICOM will be saved as "*pifaIvv_new.pifa*" and "*pifa_new.pifa,*" respectively in the same directory where the original PIFA files exist.

STEP 5: Repeat Duetto image reconstruction with new PIFAs

There are two sets of PIFA files in *examDataFolder* now: the original pair (*pifaIvv_f1b1.pifa* and *pifa_f1b1.pifa*) and the new pair (*pifaIvv_new.pifa* and *pifa_new.pifa*). You can use the new PIFA during the next PET reconstruction by specifying userConfig parameters i.e.

> userConfig.pifa = 'pifa_new'
> userConfig.pifaIvv='pifaIvv_new'

Erase all intermediate files <u>except</u> these PIFA files and repeat PET image reconstruction. You should see the progress status like the one below in MATLAB Command Window.

```
Computing PIFA images and attenuation sinograms using ptbComputeMrac
        Using existing PIFA file pifa_f1b1.pifa
        Linking PIFAs from first time bin to remaining time bins
        Making ACF (CTAC) acf_f1b1.sav
        Making ACF (CTAC) acfIvv_f1b1.sav
Computing scatter sinograms using ptbComputeScatter
        =================================================
        Estimating scatter file scatter_f1b1.sav
```

Quick Inspection of PIFA

The script ptbCtacToPifaDisplay is provided in order to visually inspect PIFA files. When this function is run, the file selection dialog box appears to select a PIFA. You can change the plane and the slice to be displayed and adjust the window level manually using the slider bars at the bottom of the GUI.
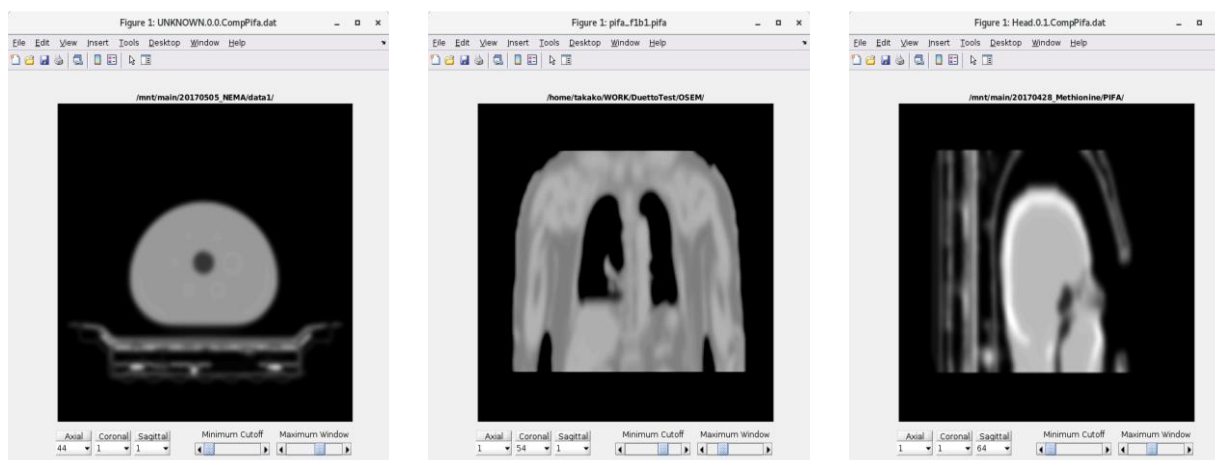
DICOM Fields

The following are common DICOM fields related to the PET image reconstruction settings:

- Recon method: (0054,1103) (toolbox recon algorithm)
- Iterations: (0009,10B2)
- Subsets: (0009,10B3)
- Q.Clear beta: (0009,10F8)
- DFOV: (0018,1100)
- Matrix size: (0015,103C)
- MRAC truncation completion: (0023,1031)
- Corrections applied: (0028,0051)
- SharpIR (PSF): (0015,103D) (also in field 0054,1103, where "S" indicates PSF)
- PET ir_z_filter_flag: (0009,10DB) (use z-axis filter or not)
- PET ir_z_filter_ratio: (0009,10DC) (corresponds to "z-axis filter" in UI, described above)
- PET post_filter flag: (0009,10BA) (use transaxial Gaussian filter or not)
- PET post_filter_parm: (0009,10BB) (corresponds to "Filter Cutoff (mm)" in scanner UI, details are described above)

Appendices

Appendix 1: Data Conventions

There are two aspects of MATLAB array indexing that must be considered:

1. The first element in MATLAB arrays are indexed beginning with 1 in each dimension, while many other parts of the software and system design (e.g. crystal-to-sinogram mapping algorithms) use the C-language convention of numbering from zero. For example, when the detector surface is interpreted as a 2-D array of crystals, the data from crystal (0,0) from block #0 in module #0 appears in the MATLAB array of crystal elements at position (1,1).

2. Multidimensional MATLAB arrays are ordered from the fastest-changing to slowest-changing index variable, which is consistent with other languages' ordering of array indices and with our conventional descriptions of multidimensional arrays. However, MATLAB's interpretation of a two-dimensional array is more like the linear algebra description of a matrix: a fast-changing row index followed by a slower- changing column index. This is different than the horizontal raster-scan convention used in many other places, including DICOM image storage. This makes it easy to confuse an image array and its transpose.

    The convention adopted in the toolbox is that every image array will be interpreted as a raster-scanned image I(x,y[,z]), where, in a typical gravity-down image presentation, "x" represents a horizontal index.

Appendix 2: TOFNAC Generation

This step will create a TOFNAC (TOF non-attenuation corrected) image series for each individual station to be used for the Truncation Completion step during the offline PIFA generation process. Note: head stations using the Head Atlas method for PIFA creation do not require a TOFNAC image series since Truncation Completion is automatically disabled for this case.

Select the desired exam in the Exams Database, and highlight the PET raw data from the Series Database. Next, highlight one of the RDFs associated with this PET raw data in the Images Database (the RDFs are the entries that have a non-blank "Frame Location" field). Click on PetRecon/Replay.

In the PetRecon/Replay GUI, enable Recon 2 and disable both Recon 1 and Recon 3. With the Recon 2 tab highlighted, edit the following recon parameters:

> Matrix Size = 128
> DFOV (cm) = 60
> R/L center (mm) = 0
> A/P center (mm) = 0
> Recon Option -> Attenuation Type = None
> Recon Option -> Well Counter = Activity
> Recon Option -> WCC File = Default
> Recon Option -> Normalization = Default
> Recon Option -> Scatter = Off
> Recon Option -> Randoms = Singles
> Recon Option -> Deadtime = On
> Recon Type -> Recon Method = VUE Point FX
> Recon Type -> Sharp IR = Off
> Recon Type -> Cardiac 3D = Off
> Recon Type -> Z-Axis Filter = Standard
> Recon Type -> Filter Cutoff (mm) = 4
> Recon Type -> Subsets = 28
> Recon Type -> Iterations = 2
> No. Bed Positions = 1
> Series Description = TOFNAC-1

Launch the reconstruction.

If this is a multi-bed study, keep the No. Bed Position = 1, and edit the "Recon Start Location" entry to be equal to the current value in the "Recon End Location" entry. This will modify both entries and essentially tell the GUI to reconstruct the second bed station only. Edit the "Series Description" to be "TOFNAC-2", and launch the reconstruction.

Repeat the above procedure until all bed stations have been reconstructed.

Appendix 3: "Home-made" attenuation correction file (PIFA)

The clinical MRAC algorithm is not designed for generation of attenuation correction files (PIFAs) from phantoms or non-patient objects. For commonly scanned phantoms, such as the 20 cm flood or NEMA-IQ phantom, there are built-in attenuation image templates on the scanner. These templates are automatically registered to the current acquisition.

However, for a non-standard phantom, the user must generate their own attenuation map. Below is a general approach that can be taken.

The most important thing is to make sure the phantom is <u>parallel to the Z-axis</u> in both the PET/CT and PET/MR scans.

Choose any scan mode that allows AC
- When scanning on PET/MR, select either patient or phantom AC mode. A good option to select is the fillable WCC phantom from the phantom list. Do not select "other" or "none" in the phantom AC list, as they will not allow AC recon.
- Make sure that an attenuation-corrected reconstruction has been completed on the host computer to generate the PIFAs.
- Create a 60 cm 128x128 TOFNAC image set, which will be used to line up the PET/MR image with the CT image.

Grab the PIFAs from the host computer
Access the PIFAs of the raw data frame of interest. See steps in Section 3 above on how to find the path to the PIFA files. Copy these files offline into a MATLAB environment. There will be a "CompPifa" file that contains the patient/phantom plus coils, cradle, etc. There will also be a "InVivo" file that contains patient/phantom only (no coils, etc)

CT Scan of phantom
- Take a CT scan of phantom. It is recommended to use 140 kVp and fairly high mA to avoid any issues.
- It is recommended to generate 60 cm recons, which will make resampling, etc, a little simpler. Similarly, reconstructing with a slice interval of 2.78 mm will make things simpler as well. (The PIFAs are 128x128 with 60 cm diameter, 2.78 slice interval.)

In MATLAB
- Load the CT images into MATLAB.
- Do some thresholding/image processing to strip away everything except the phantom. Get rid of the cradle, etc. Leave a little buffer at the outside of the phantom (a few mm). Remember that the background (everything outside the phantom) should be -1000 HU.
- Align the masked CT images with the 60 cm PET TOFNAC, resampling the CT images appropriately to 128x128 transaxially over 60 cm DFOV and with a slice interval of 2.78 mm. If the phantom was aligned parallel to the Z-axis in both PET/CT and PET/MR, then this is simply a potential translation in the three dimensions and in-plane rotation (about the Z-axis).

- Convert the CT values into mu-values (attenuation coefficient). Assuming that the transformed CT images (which now align with the PET/MR images) is assigned to the 3D matrix variable `"ctImagesTransformed"`, the mu values can be converted with the following lines:

    ```
    LUT = ptbBuildCtacLut(140, 0, which('ctacConvScale_rev5.cfg'));
    pifaFromCtImages = LUT.table(int16(round(ctImagesTransformed)-
    LUT.floor+1));
    ```
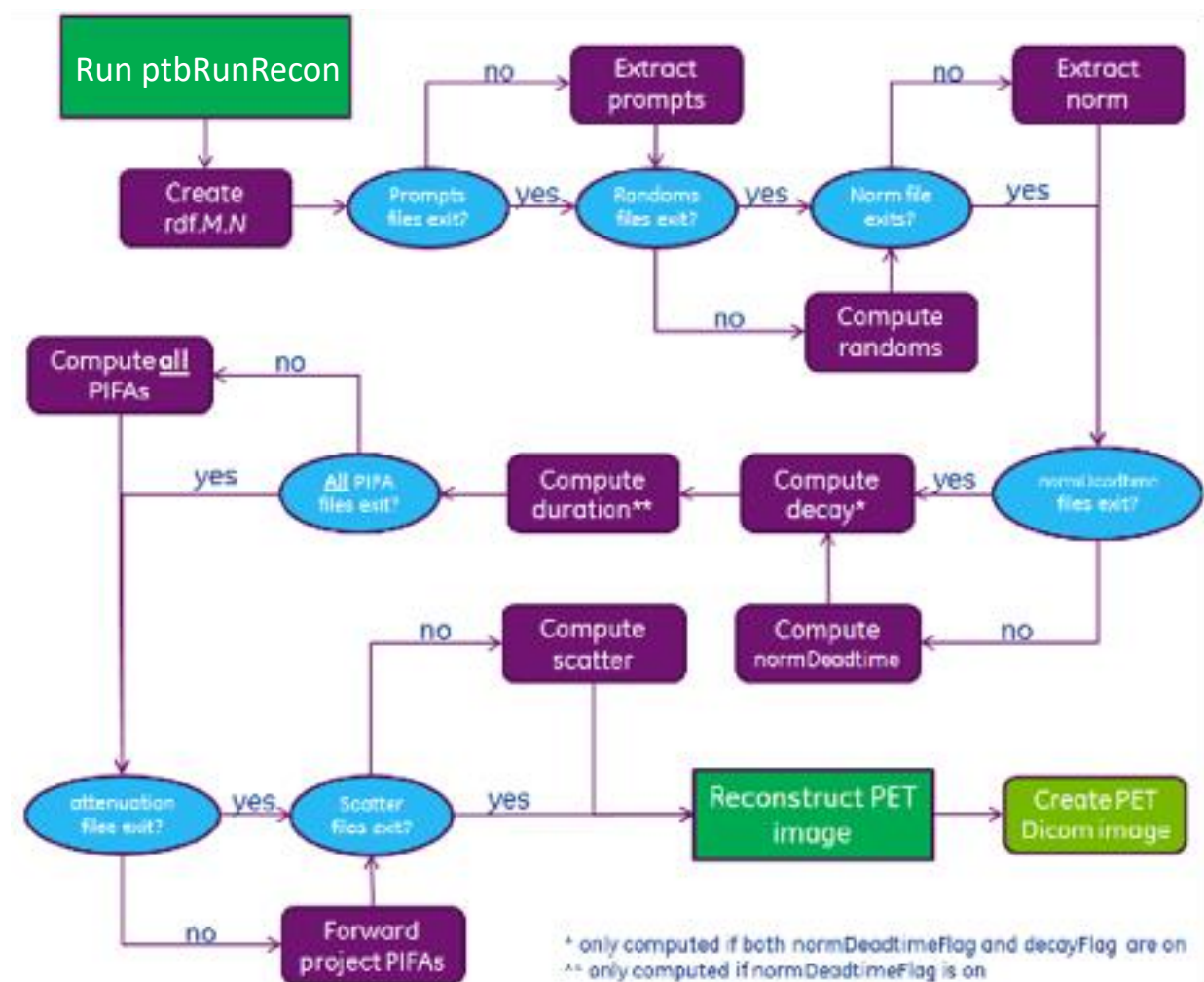
    Note: Replace 140 with the actual kVp with which the CT was acquired.
- Smooth the PIFAs with a Gaussian filter of about 10 mm FWHM in each dimension. This reduces negative impact from imperfect registration. This smoothing is performed on the patient PIFAs as well, as can be seen in the smoothed AC Map series.
- The orientation of the PIFA data is explained in the attached "PIFA Orientation.pptx" file. A check of the PIFA orientation will be easier on phantoms that have more features, such as the Hoffman phantom. the Hoffman phantom has tons of features, so this will be easy to check in the AC Map Dicom series on the scanner.
- Read in the PIFAs from the host computer using ptbReadPifa. Subtract the in vivo PIFA file from the CompPifa file. This gives only the non-patient/non-phantom components.
- Add the newly created phantom mu map to the non-patient/non-phantom components to make the new CompPifa file. Write the complete PIFA to disk using ptbWritePifa.
- The phantom portion is the in vivo PIFA. Write only the phantom portion to disk using ptbWritePifa.
- Give these modified PIFA files the same name as the original PIFA files.

Put them Back on the Host Computer
- Replace the original PIFA files with these modified PIFAs in the same location on the host computer.
- Perform an AC recon.
- ***Most important: View the associated AC Map DICOM series to make sure that everything is aligned appropriately!***
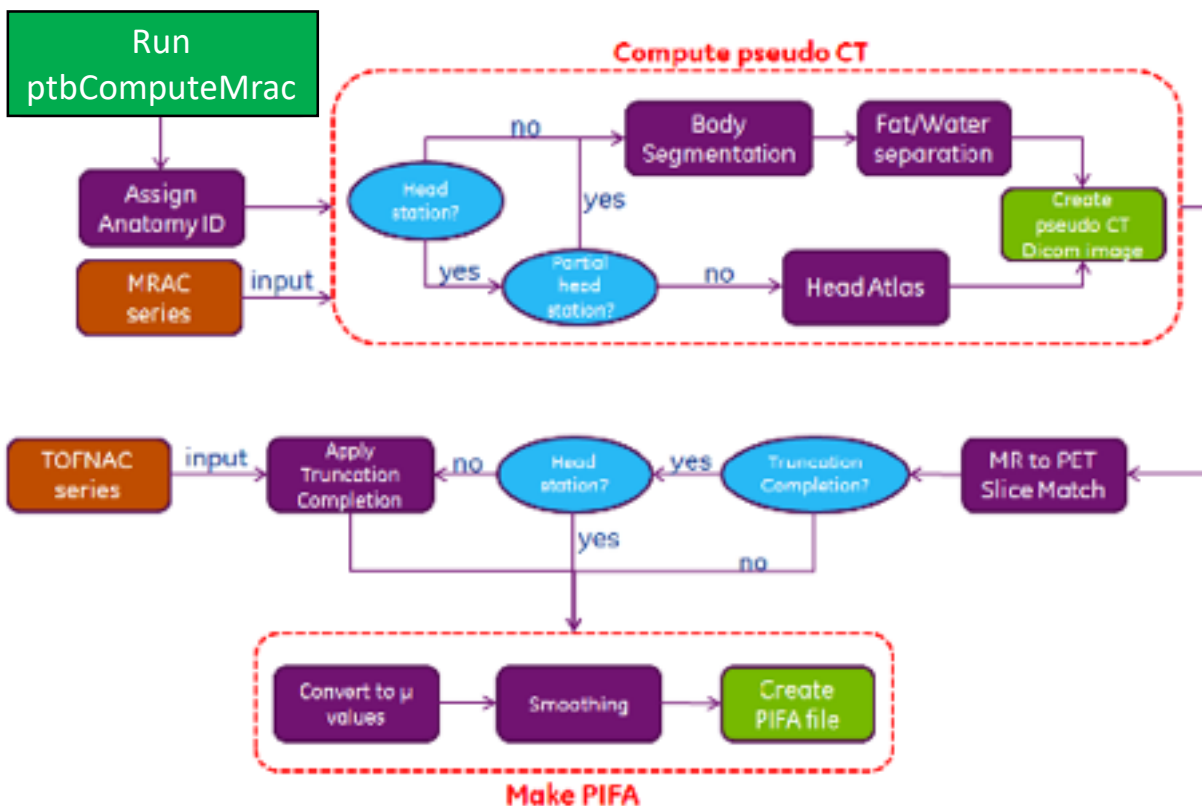
Appendix 4: Duetto Flowcharts

* only computed if both normDeadtimeFlag and decayFlag are on
** only computed if normDeadtimeFlag is on

The following figure shows the in-vivo patient MRAC PIFA creation steps:

Appendix 5: Running Duetto on Windows and Mac

It is recommended running Duetto on Linux systems as most of the testing and evaluating processes are performed on Linux systems. Duetto is still in its testing phase on Windows and Mac platforms. Users may run Duetto on those operation systems with the limitations below:

- PIFAs cannot be generated from MRAC images on Windows and Mac, they must be extracted from the scanner or generated on a Linux system.
- List and raw data are compressed by GE proprietary filters. Although the depression filters are available for Linux systems, those filters are not all available for Windows and Mac. In Windows and Mac, RDFv9 raw data can be decompressed by Duetto (not list data). For this reason, all list data files must be decompressed on a PET console before they can be unlisted on Windows and Mac systems (please see Section 9 for decompression details). For RDFv10, users need to decompress raw data using `rdfRepack` on a RDFv10 console before processing using Duetto.
- For Windows, Duetto source files must be extracted in a path without any spaces, e.g. C:\Users\research\Documents\duetto\.

Appendix 6: Disclaimer and Release Notes

Disclaimer

The toolbox is certified against particular product software versions. A range of image reconstruction scenarios are tested by comparing reconstructed images between the toolbox and product, and any differences are within a small tolerance. While a wide range of scenarios are tested, the testing is not comprehensive, and therefore there may be cases where results between the toolbox and product images will be different. It is advisable to compare images between toolbox and product when image matching is imperative for the desired study.

Release Notes

Duetto_v01.06_Feb2019

- Code changes
  - None; equivalent to the PET/MR MP26_M3 release of the petRecon toolbox.
- Certification against scanner release versions
  - PET/MR MP26_M3 release (software petmr_k1.99)

Duetto_v02.03_Sep2019

- Code changes
  - User no longer needs to specify "scannerName"; order of inputs to ptbUserConfig is modified
  - Improved handling of recon levels
  - Improvement in parallel processing via userConfig.nParallelThreads
  - PET/MR recon improvements:
    - Deadtime correction constants
    - Q.Clear beta scaling for fast decaying radioisotopes
  - ZTE improvements in the neck region
  - Ability to read RDF10 list data format (PET/CT only)
  - Added a function to sort DICOM files into directories based on their series description
  - Some improvements to DICOM header fields written to reconstruction images
- Ability to mex and run the Unlister on Mac and Windows platforms
- Certification against scanner release versions
  - PET/MR MP26 R02 release (software petmr_k1.142)
  - PET/CT DMI (3, 4, and 5 rings) software version 175

Duetto v02.06 Mar2020

- Modification of correction options from Enums (e.g. NONE, PRE, LOOP) to flags (true/false) where true indicates "in the loop" correction
- Modification of PSF options, including addition of image-space PSF option
- Addition of "CT to PIFA" helper tool
- General fixes and improvements

Duetto v02.07 May2020

- Addition of FFBP reconstruction

Duetto v02.11 Sep2020

- PET/MR MP26 R02 release
- MRAC rigid transformation updates

Duetto v02.13 Nov2020

- Addition of a license control tool
- Updated RDFv10 decompression filter libraries for compatibility with DMI Gen 2
- Certification against scanner release versions:
  - o DMI (3, 4, and 5 rings) and DMI-DR software version digital.9

Duetto v02.15 Apr 2021

- Performance Improvement of reading DICOM files
- Certification against scanner release versions:
  - o DMI and DMI-DR software version dmi.20

Duetto v02.16 Nov 2021

- Unlist performance improvement
- Enabling Windows and Mac supports
- Certification against scanner release versions:
  - o DMI and DMI-DR software version dmi.20 (same as in v02.15)

Duetto v02.17 Dec 2021

- Minor bug fixes.

Duetto v02.18 Jan 2023

- Supporting Omni Legend 3-ring and 6-ring reconstruction
- Scan Data Manager data conversion
- Enabling cardiac 3D filtering
- A new image display function – ximage
- Supporting RDFv8 reconstruction
- Improvement of DICOM headers for dynamic reconstructions
- Certification against scanner release versions:
  - o DMI and DMI-DR software version dmi.20
  - o Omni Legend software version omni.27