Process MOLLI and shMOLLI data

Reads a directory of shMOLLI DICOM images and processes the data to produce a look-locker corrected T1 map. Negative values are cropped to 0.

Two possible methods for processing the data are available.

1. Fast (default). Uses some numerical methods to solve exponential equations of the correct form. May be slightly less accurate, but significantly faster.
2. Slow. Uses scipy curve fitting algorithm to fit the exponential curve. More accurate, but much slower.

The difference between the two methods is very little as demonstrated in the figure below.

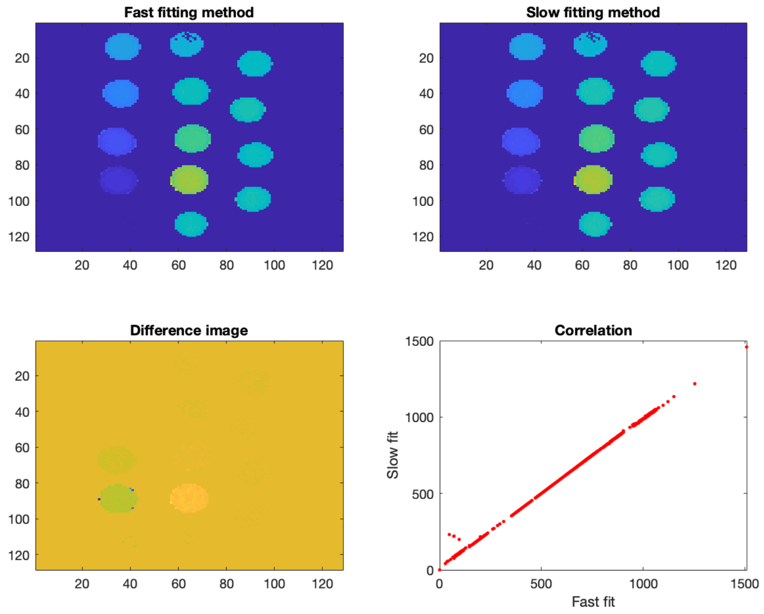

Figure 1: fast_slow comparison

Any pixels/voxels where the max value is less than 100 is skipped. NB: For this version, data is processed top-left to bottom-right in the image. As images are often empty around the edges, this means the the processing is 'fast' to start with as each pixel is basically skipped, then slows down as the true data is processed, then speeds up again at the end.

Usage

As module

```
> pip3 install sh_molli
> sh_molli.py -i <inputfolder> -o <outputfilename> -p <plot_flag> -m <method> -d <dicom_tag
```

In code

```
> python3
> import sh_molli.sh_molli as sh
> im.process_folder(path, method='fast', dcmtag=1)
```

```
process_sh_molli_series.py -i <inputfolder> -o <outputfilename> -p <plot_flag> -m <method> -
    intputfile is a path to a folder containging DICOM images from a shMOLLI or MOLLI series
        NB: likely to fail if other files are in the directory
    outputfilename should be a string of the format:
            filename.EXT where EXT is any extension understood by PIL
        NB: careful of bit-depth of output. TIFF is OK.
    method {fast, slow}
        fast = numerical methods fit    - less accurate
        slow = scipy.curve_fit method   - slower
    dicom_tag {0, 1, 2}
        0 = TriggerTime
        1 = InversionTime
        2 = ImageComments
```