1. **Before running the code**: make sure that mex file of L-BFGS-B is compiled:

Go to \GUI\_3DFilterVer--\Subcodes\L-BFGS-B-C-master\Matlab and run “compile\_mex”

1. Please make sure that relative order of various sub folders are not changed.
2. This code can also be used to convert raw 4D (or 3Dt) data set to nifty format. For that, just select raw data folder and then press the button: “Raw Dicom 🡪 Nifti (No MWI Analysis)”. You can also process directly from your raw data.
3. **Preprocess**: The same code can be used to to convert “**raw dicom data**” to **T2 relaxometry (3Dt) “nifty file**”. It also saves it as \*Avg\_T2RelaxometryData.mat”, which also contains TE vector beside 3Dt data. It also makes a nifty mask as well. The mask is not very accurate; but, it removes non brain part quite well. If **you need accurate mask, please make it manually: either in “nifty”or “roi” format (described later)**. The “.roi” format is what you get, when you use MriCro to make the mask. All nifty files could be find in the subfolder named: “PreprocessedData\_N\_Analysis”
4. MWI-analysis: Regarding running MWI-analysis, there are three options
5. Pass raw dicom files and the code will generate 3Dt T2 relaxometry data in nifty file (also \*.mat file), make a nifty mask and then will process. The mask is not very accurate; but, it remove non brain data quite well. In this case, the folder should be the raw dicom folder.
6. If 3Dt T2 relaxometry data already exist in nifty or mat format. In this case, the folder should be “PreprocessedData\_N\_Analysis”.
7. **Some features**
   1. Use this code to convert raw dicom images to 4D (3dt) T2 relaxometry nifty image.
   2. The conventional regularization uses “Prasloski’s method”. I made this change during the revision of my NeuroImage paper. I still use L-curve based method to choose temporal regularization.
   3. The code follows my NeuroImage paper closely: https://doi.org/10.1016/j.neuroimage.2018.05.026 Received
   4. **Virtual cores vs physical core**: For **conventional regularization**, a speed up can be achieved by processing on all virtual cores (= 2 x number of physical cores). This speed up is not possible for spatial regularization and hence, maximum matlab pool workers allocation is restricted by “number of physical cores”.
   5. Up to six data sets can be queued for processing.
8. **Default processing options**: Once you change default values and close the GUI, new values will be saved. If you would like to **return to default value**, then delete “state” file contained in outer folder.
9. **T2 Scale & FAE scale**: Please feel free to change T2 characteristics as per your choice.
10. **Resolution & Nominal FA**s: This should be input based on your experiment.
11. **Data Selecting Window**: It would have been ideal to solve for entire data at once. However, that would not be feasible given limited computational resources. So, I am using window size of 12x12x16 with overlap regions of 4 voxel wide.
12. **Regul. Tag**: There are three options: Temporal-Only, Spatial-Only, Both. You will be using “Both” mostly.
13. **Mask Option**: You can choose automatic mask option or you can make mask manually and save it as nifty or roi file. Please make sure that the name of file should contain the string: “mask”.
14. **Stimulated Option**: Use “WITH FAE correction” ONLY. Other option is not working.
15. **Selection Echo Tag**: I will recommend choosing “AllEchoes”, though you can use “AllEchoes\_Except\_1st”.
16. **muS\_Optimization**: I would recommend using “Optimize muS”. However, if you have enough experience and have idea of what muS should be, then you can choose “Use set value”. That will create a new field “Spatial Alpha-1” (same as Alpha\_s1 in the paper), where you can input your desired inputs. Of course, for that I have to

**For the time being, please use “Optimize muS”.**

Please let me know if you would like to “Use set value” option, then I have to make some additional changes to the code.

1. **1s Diff Op**: Use “3D” option if you have 3Dt data. Else choose “2D”.
2. **# Iterations**: choose values between 4-6. Though I have been using 6 iterations,
3. **Myelin cutoff**: Choose as per your choice.
4. Final Result: The final result would be saved as: “---Recur\_?.mat”. You can read these mat files:

Myelin\_Cutoff = [5e-3; 40e-3];

MyScaleMax = 0.3; %.22;

cd(Dir);

S1 = load('-------------Recur\_6.mat');

T2 = S1.T2\_Scale;

%T2\_dist\_4D1 = S1.T2\_dist\_4D\_Temporal; % Temporally regularized solution

T2\_dist\_4D1 = S1.T2\_dist\_Spa\_4D; % Spatially regularized solution

FAErrorMap\_3D0 = S1.FAErrorMap\_3D; % Flip angle inhomogeneity map

muT\_Map\_3D = S1.muT\_Map\_3D;

% Calculate MWF-map

MWF\_Spa3D0 = sum(T2\_dist\_4D1(:,:,:,find(T2 >= Myelin\_Cutoff(1) &T2 <= Myelin\_Cutoff(2))),4)./(sum(T2\_dist\_4D1,4)+eps);