Multi-Parameter Mapping Toolbox User Guide

Introduction

This MATLAB toolbox contains a few program files (.m files) to perform fast multi-parameter mapping.

At present, it can map up to 5 parameters including proton-density, T1, T2*, T2, and magnetization transfer from a minimum of 3 FLASH and 4 bSSFP sequences, (with the correct input bias fields or B1 maps). If you wish to implement MPM in your own imaging lab, you will have to optimize these sequences on your own scanner, and then prepare the reconstructed images to be processed by this pipeline. If you have a Philips scanner with the ReconFrame software patch installed (Gyrotools, LLC, http://www.gyrotools.com/products/gt-software.html), I may be able to provide you some additional scripts to prepare the pre-processed images from the raw data (see the contact info below). Otherwise, you will have to prepare your own image datasets from dicom images, or whichever file format your MRI scanner outputs. If this toolbox proves useful in publishing a paper, please cite at least one of these two publications:

Jutras, J.-D., K. Wachowicz, N. De Zanche. (2016) Analytical corrections of banding artifacts in driven equilibrium single pulse observation of T2 (DESPOT2). Magnetic Resonance in Medicine 76(6):1790-1804. DOI: 10.1002/mrm.26074

Jutras, J.-D., K. Wachowicz, G. Gilbert, N. De Zanche. (2016) SNR Efficiency of combined bipolar gradient echoes: Comparison of three-dimensional FLASH, MPRAGE and multiparameter mapping with VFA-FLASH and MP2RAGE. Magnetic Resonance in Medicine (Early View). DOI: 10.1002/mrm.26306

I wanted to include a number of example datasets so users can test the toolbox quickly and check that it works on their system; however, the 3D datasets tend to occupy a lot of memory (~ 1GB) and so GitHub will not let me include them on their site. Therefore, you can download a sample phantom datasets from my public dropbox folder (~700MB), and test it as the fourth example shown at the end of the Sample Examples section: https://www.dropbox.com/sh/355bbuom2zt0qho/AACphqU1UkOk_IEcrt_0vfUia?dl=0 If you would like me to send you more example datasets via a Gmail account (a shared link on google drive), see my contact information at the end.

Installation Instructions

To install the MPM toolbox, save the *MPM_tools* folder in your Matlab path, as well as the two subfolders *BCFCM_v1* and *NifTI_20140122*. The BCFCM is a bias-field corrected fuzzy-c-means classification algorithm[1] used to generate the mask associated with *T1app* and *M0app* mha files. You

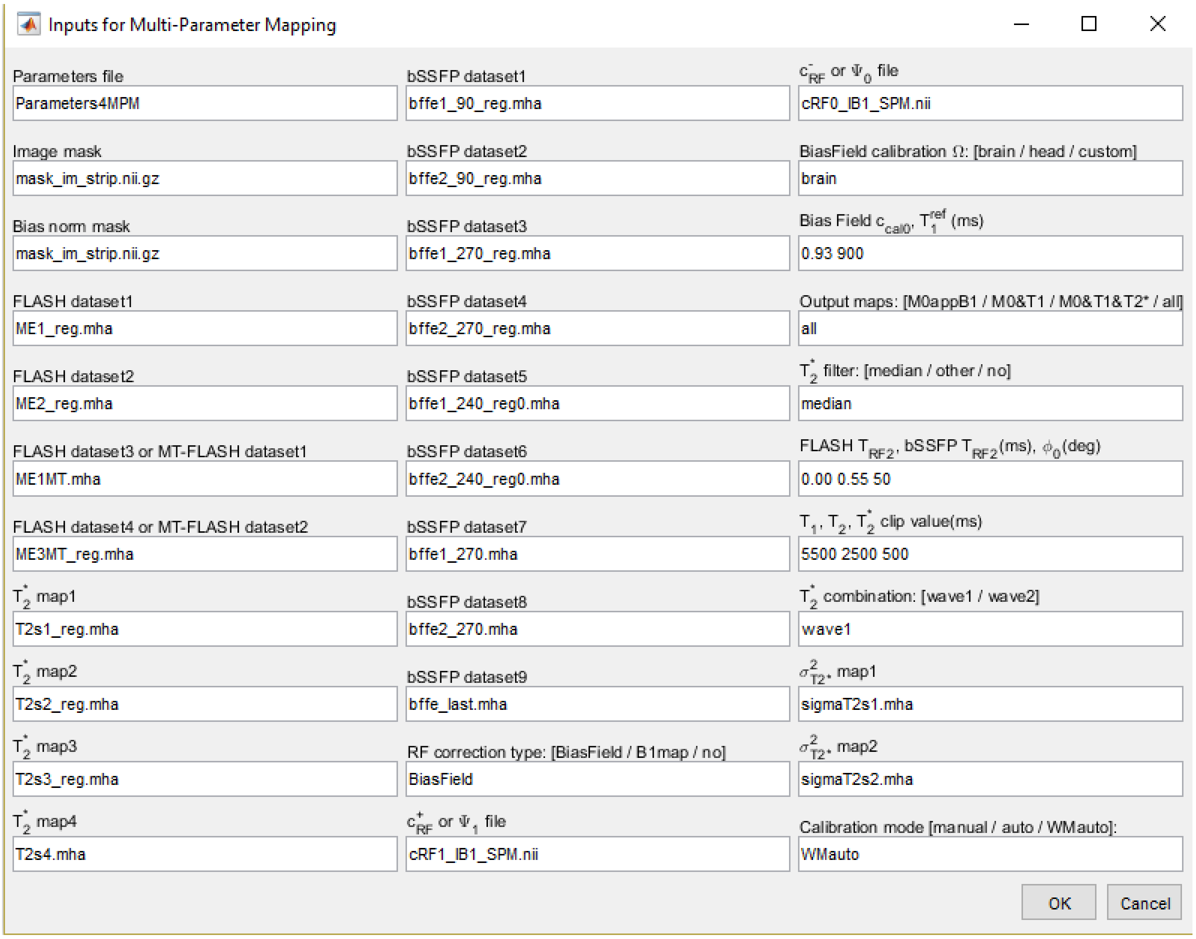
[1] Unfortunately, I am quite positive that the author of this Matlab program did not correctly implement the BCFCM classification method proposed by the original authors of the paper. The FCM algorithm seems to diverge after a few iterations and the bias field it generates contains too much image contrast to be the correct one. By the time I realized this, I had been using this BCFCM program for a while. It still gives you a type of FCM-looking classification, but do not trust it or use it for important classification/segmentation studies. Also, do not use more than 5 iterations as input.

may have to compile the BCFCM code to run on your own system by typing "*mex BCFCM3D.c*", etc, in MATLAB. Note that those two subfolders have their own separate copyrights and authorships.

Running MPM_main_v1 with sample data provided

The main program file that performs the post-processing is *MPM_main_v1.m* and contains about 1200 lines of code in Matlab. It only takes ~20-45 sec to curve-fit everything (excluding the waiting time of two user g-inputs required if you set the calibration mode to "manual"), because all operations are linear and performed on 3D arrays. The program contains a large number of options that must be specified, depending on what parametric maps you wish to obtain and exactly how many raw images you choose to include in the curve-fit. If a required file is missing or misnamed, you will get an error. Moreover, if your data is not correctly or consistently scaled, you may not get an error, but your final parametric maps will be wrong. The datasets must all have been co-registered and resampled consistently prior to being curve-fitted. I recommend using 3D slicer to do the image registration prior to curve-fitting, but SPM or FSL should work as well. There are two functions included within the folder to write and read mha image files (*mha_read_volume* and *mha_write_volume*) generated from Matlab, but you could also use another software like FSL and nifti image files. The program accepts *.mha*, *.nii* or *.nii.gz* images files to do the processing.

When you run MPM_main_v1, the following window will pop up as shown in



Parameters file
Parameters4MPM

Image mask
mask_im_strip.nii.gz

Bias norm mask
mask_im_strip.nii.gz

FLASH dataset1
ME1_reg.mha

FLASH dataset2
ME2_reg.mha

FLASH dataset3 or MT-FLASH dataset1
ME1MT.mha

FLASH dataset4 or MT-FLASH dataset2
ME3MT_reg.mha

T_2^* map1
T2s1_reg.mha

T_2^* map2
T2s2_reg.mha

T_2^* map3
T2s3_reg.mha

T_2^* map4
T2s4.mha

bSSFP dataset1
bffe1_90_reg.mha

bSSFP dataset2
bffe2_90_reg.mha

bSSFP dataset3
bffe1_270_reg.mha

bSSFP dataset4
bffe2_270_reg.mha

bSSFP dataset5
bffe1_240_reg0.mha

bSSFP dataset6
bffe2_240_reg0.mha

bSSFP dataset7
bffe1_270.mha

bSSFP dataset8
bffe2_270.mha

bSSFP dataset9
bffe_last.mha

RF correction type: [BiasField / B1map / no]
BiasField

c_{RF}^{+} or Ψ_1 file
cRF1_IB1_SPM.nii

c_{RF}^{-} or Ψ_0 file
cRF0_IB1_SPM.nii

BiasField calibration Ω: [brain / head / custom]
brain

Bias Field c_{cal0}, T_1^{ref} (ms)
0.93 900

Output maps: [M0appB1 / M0&T1 / M0&T1&T2* / all]
all

T_2^* filter: [median / other / no]
median

FLASH T_{RF2}, bSSFP T_{RF2}(ms), ϕ_0(deg)
0.00 0.55 50

T_1, T_2, T_2^* clip value(ms)
5500 2500 500

T_2^* combination: [wave1 / wave2]
wave1

$\sigma^2_{T2^*}$ map1
sigmaT2s1.mha

$\sigma^2_{T2^*}$ map2
sigmaT2s2.mha

Calibration mode [manual / auto / WMauto]:
WMauto

Figure *1*. The first entry contains the name of a *.mat* parameter file *Parameters4MPM*, which contains an *mpm_par* structure array. The *mpm_par* contains all the information on how the data was prepared prior to curve-fitting as well as some scan parameters. The scan parameters in the sample data were obtained (in part) using the ReconFrame patch developed by GyroTools, LLC, Switzerland, http://www.gyrotools.com/products/gt-software.html which is a special package and toolbox that can be purchased only for Philips scanners. The advantage of using ReconFrame is that you can take the raw data files from the scanner and reconstruct them offline using MRecon, which is a Matlab toolbox that also comes along with ReconFrame. ReconFrame and MRecon avoid the cumbersome experience of dealing with dicom files, where each image file does not usually contain more than a single echo and slice. Moreover, you can go back to square one and choose to reconstruct new images from the raw data using different parameters, such as the amount of filtering or zero-padding, etc. A short explanation is listed for each parameter below:

mpm_par.

no_slices=180 Number of slices (3rd or z index) of each *.mat* image array.

fov=[240 240 170] Field-of-view in mm.

B1_corr='no' Whether a B_1 map was acquired and saved as the *c* array under *Parameters4MPM*. If "no," the *c* array will only contain ones.

filter_strength1=[0.3500 0.3500 0.3500] Ringing filter strength applied to the FLASH images (between 0 and 1). Lower values gives better PSF but poorer SNR.

calculation='DESPOT1&2' Specifies if only "DESPOT1" (VFA T1) or if both DESPOT1 and DESPOT2 scans were performed (i.e. "DESPOT1&2").

filter_strength2=[0.3500 0.3500 0.3500] Ringing filter strength (ranging from 0 to 1) for the DESPOT2 or bSSFP images.

N_T1FFE_ang=2 Number of DESPOT1/VFA angles acquired. Can range from 2 to 4.

N_bFFE_ang=2 Number of DESPOT2 flip angles acquired. Can allow up to 4 if using two phase offsets (see next parameter), up to 3 if using 3 phase offsets and only 2 if using 4 phase offsets.

N_offsets=2 Number of DESPOT2 phase offsets acquired. Can range from 2 to 4.

Anatomy='Brain' Anatomy of the images, usually "brain".

coil_type='SENSE' Coil type that was used, such as an 8-channel head SENSE array, or a birdcage coil.

T2s_clip=250 Clip value (in ms) of the T_2^* maps already curve-fitted.

MT_scan='yes' Whether an MT-FLASH dataset was acquired. If yes, you must enter the *ME1MT.mha* and *ME3MT.mha* file names in the correct boxes in place of a 3rd and 4th VFA dataset.

last_echo=[12 8 12] Specifies how many echoes were incorporated in the MERGE combination for M_0/T_1 (1st entry), for MT_{sat} (2nd entry) and for the T_2^* maps (3rd entry). Later echoes tend to be more motion-sensitive, so if the subject moved his/her head more during the examination, the PSF in the final T1/M0 maps can be improved by not using all the echoes in the T_1, M_0 and MT_{sat} maps (e.g. using only the first 6 echoes instead of all 12). This is an important parameter that must not be wrongly specified, or the M_0 map will be affected.

file_ext='mha' File extension of the prepared datasets.

reg_strength=1.4100 Regularization strength of the SENSE reconstruction. Not really used anywhere at present.

TR1=31.5000 TR (in ms) of the first FLASH dataset (PD-weighted).

TE1=[2.3005 4.6509 7.0013 9.3517 11.7021 14.0525 16.4029 18.7533 21.1037 23.4541 25.8045 28.1549] Echo times (in ms) of the first FLASH dataset (PD-weighted).

ang1=6.5000 Flip angle (in deg) of the first FLASH dataset (PD-weighted).

TE2=[2.3000 4.6504 7.0008 9.3512 11.7016 14.0520 16.4024 18.7528 21.1032 23.4536 25.8040 28.1544] Echo times (in ms) of the second FLASH dataset (T1-weighted).

TR2=31.5000 TR (in ms) of the second FLASH dataset (T1-weighted).

ang2=35 Flip angle (in deg) of the second FLASH dataset (T1-weighted).

TE3=[2.3005 4.6509 7.0013 9.3517 11.7021 14.0525 16.4029 18.7533] Echo times (in ms) of the third FLASH dataset (T1-weighted or MT-weighted).

TR3=31.5000 TR (in ms) of the third FLASH dataset.

ang3=6.5000 Flip angle (in deg) of the third FLASH dataset.

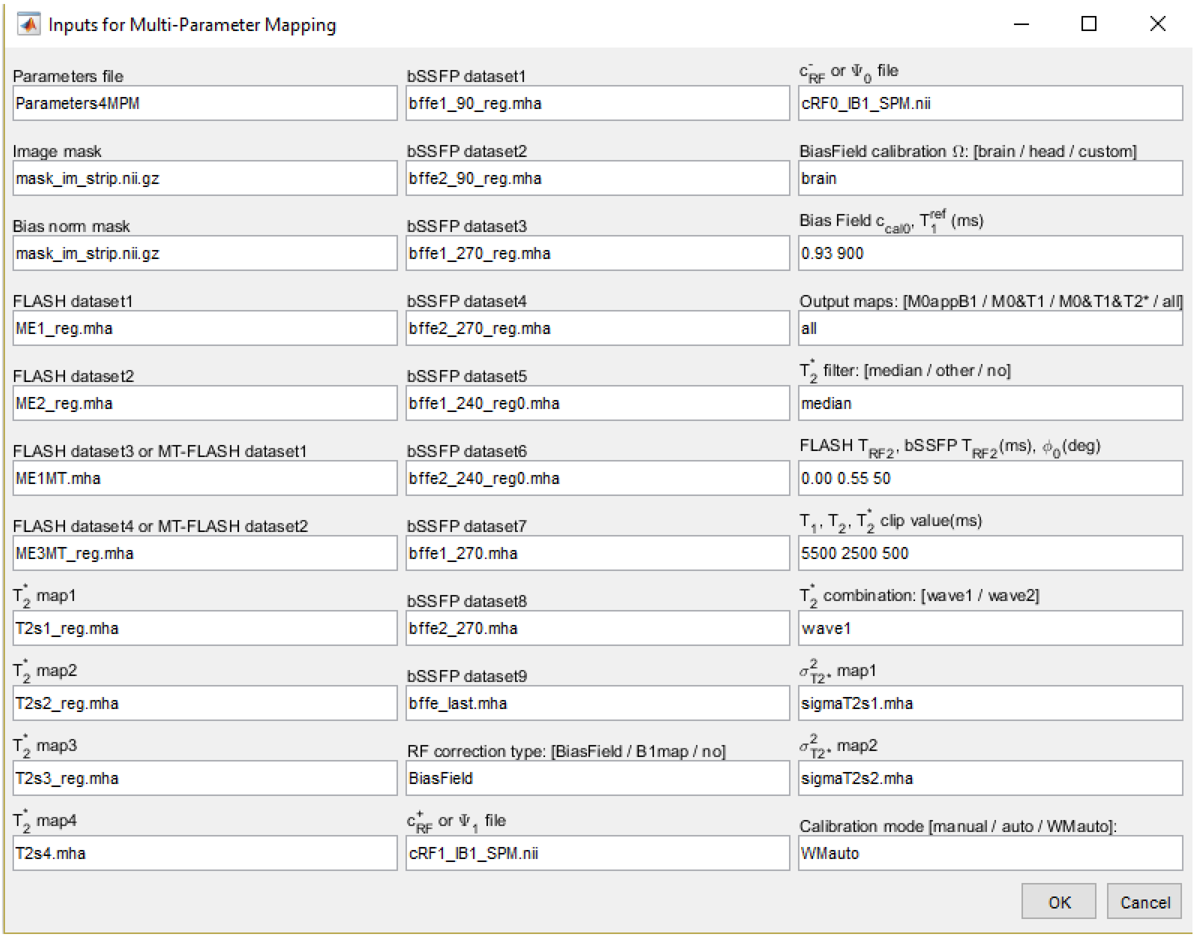
noise_matrix=[8x8 single] Noise correlation matrix, not used anywhere at present, but can be helpful to detect “dead coils”.

bffe_ang1=12 Lower flip angle (in deg) of the bSSFP dataset.

bffe_ang2=58 Higher flip angle (in deg) of the bSSFP dataset.

bffe_TR=4.8002 TR (in ms) of the bFFE datasets.

bffe_TE=2.4001 Echo time (in ms) of the bFFE datasets.



Inputs for Multi-Parameter Mapping

Parameters file
Parameters4MPM

Image mask
mask_im_strip.nii.gz

Bias norm mask
mask_im_strip.nii.gz

FLASH dataset1
ME1_reg.mha

FLASH dataset2
ME2_reg.mha

FLASH dataset3 or MT-FLASH dataset1
ME1MT.mha

FLASH dataset4 or MT-FLASH dataset2
ME3MT_reg.mha

T_2^* map1
T2s1_reg.mha

T_2^* map2
T2s2_reg.mha

T_2^* map3
T2s3_reg.mha

T_2^* map4
T2s4.mha

bSSFP dataset1
bffe1_90_reg.mha

bSSFP dataset2
bffe2_90_reg.mha

bSSFP dataset3
bffe1_270_reg.mha

bSSFP dataset4
bffe2_270_reg.mha

bSSFP dataset5
bffe1_240_reg0.mha

bSSFP dataset6
bffe2_240_reg0.mha

bSSFP dataset7
bffe1_270.mha

bSSFP dataset8
bffe2_270.mha

bSSFP dataset9
bffe_last.mha

RF correction type: [BiasField / B1map / no]
BiasField

c_{RF}^+ or Ψ_1 file
cRF1_IB1_SPM.nii

c_{RF}^- or Ψ_0 file
cRF0_IB1_SPM.nii

BiasField calibration Ω: [brain / head / custom]
brain

Bias Field c_{cal0}, T_1^{ref} (ms)
0.93 900

Output maps: [M0appB1 / M0&T1 / M0&T1&T2* / all]
all

T_2^* filter: [median / other / no]
median

FLASH T_{RF2}, bSSFP T_{RF2}(ms), ϕ_0(deg)
0.00 0.55 50

T_1, T_2, T_2^* clip value(ms)
5500 2500 500

T_2^* combination: [wave1 / wave2]
wave1

σ_{T2*}^2 map1
sigmaT2s1.mha

σ_{T2*}^2 map2
sigmaT2s2.mha

Calibration mode [manual / auto / WMauto]:
WMauto

Figure 1: MPM control panel containing up to 33 entries to specify.

The following file and parameter inputs of the control panel are explained below:

Image mask: Mask used to zero the background/air cavities after curve-fitting. This mask is usually created from the first echo of the PD-weighted FLASH dataset, after bias-field correction and filtering with Gradient Anisotropic Diffusion in 3D Slicer. If you want the final parametric maps to be skull-stripped, enter the *mask_im_strip.nii.gz* file name here, otherwise, if you want to visualize the whole head, use the *mask_im.mha* file.

Bias norm mask: Mask over which the N4ITK/SPM12 normalization will be performed. You can use either a skull-trip brain mask generated from the bet2 function in FSL (i.e. *mask_im_strip.nii.gz* of the sample data), or you can use the whole head mask (i.e. *mask_im.mha*).

FLASH dataset1: File name of the first FLASH dataset (PD-weighted because of lower flip angle α_1) combined in root-sum-of-squares. “ME” stands for MERGE or Multi-Echo.

FLASH dataset2: File name of the second FLASH dataset (T1-weighted because of higher flip angle α_2) combined in root-sum-of-squares.

FLASH dataset3 or MT-FLASH dataset1: File name of the third FLASH dataset (T1-weighted with flip angle α_3) combined in root-sum-of-squares. However, if within the parameter file **MT_scan='yes',** then the correct file to put in here is **ME1MT.mha**. This file is a MERGE combination of the first N_{MT} echoes of the FLASH1 dataset, where N_{MT} is the same total number of echoes used in the MT-FLASH sequences. I usually choose 8 echoes for MT-FLASH and 12 for FLASH1 and FLASH2.

FLASH dataset4 or MT-FLASH dataset2: File name of the fourth FLASH dataset (T1-weighted with higher flip angle α_4) combined in root-sum-of-squares. However, if in the parameter file, you have **MT_scan='yes'.** Then the correct file to put in here is **ME3MT.mha**. This file is a MERGE combination of the N_{MT} MT-FLASH echoes.

T_2* map1, etc…: File names of the T_2^* maps already curve-fitted from the multi-echo FLASH. Of course, you can always use a T_2^* map generated from a different pulse sequence if desired. Since the toolbox allows for up to 4 multi-echo FLASH datasets, up to 4 T_2^* maps can also be combined in a weighted average to obtain the final T_2^*. The program knows how many T_2^* maps to call by reading the number of original FLASH datasets from **N_T1FFE_ang.** However, if **MT_scan='yes'**, the 3rd T_2^* map from the MT scan will not be used, since is usually generated from fewer echoes and will have lower SNR than the other two maps. The final combined T_2^* map is also necessary to solve for the proton-density map M_0. The T_2^* maps provided as sample data have not been corrected for ΔB_0 inhomogeneity.

bSSFP datasets1, etc…: File names of the bSSFP datasets to be used in DESPOT2 (T_2-mapping). Note that bSSFP image files will only be opened if **Maps: 'all'** or **'T1&T2',** otherwise, no T_2 maps will be generated. Note that the datasets must be entered in the correct order as specified in the command window when you run the program and the control panel pops up: "Enter bSSFP datasets: ang1-phi1, ang2-phi1, etc. Also enter phase offsets in this order: 2 phases: phi1, phi2, 3 phases: phi1, phi2, phi3, and 4 phases: phi1, phi3, phi2, phi4."

RF Correction type: [BiasField/B1map/no]: This option specifies the choice of B_1 correction scheme. If you choose "B1map," you must enter the c_{RF}^+ map file (e.g. *cRF_AFI.mha*) within the box "**c_{RF}^+ or ψ_1 file**", as well as the c_{RF}^- map file (e.g. *cRFminus_AFI.mha*) in the box "**c_{RF}^- or ψ_0 file**". The c_{RF}^+ field is generated from a B_1-mapping sequence (such as AFI or SDAM, etc), while the cRF- can be optained by fitting the $M_{0,B1}^{app}$ image in N4ITK. If you choose "no," it will not perform any B_1 inhomogeneity correction. If you select "BiasField," you must include the names of the ψ_1 and ψ_0 bias fields (e.g. *cRF1_210_i400.mha* and *cRF0_185_i400b.mha*). You can also use SPM12 to generate the two bias fields.

BiasField calibration Ω: [brain/head/custom]: Determines the choice of calibration volume/area Ω for N4ITK (or SPM12). If you choose "brain," ensure that the skull-strip mask *mask_im_strip.nii.gz* file is present under the field "**Bias norm mask**." If you choose, "head," use the *mask_im.mha* file instead. If you choose "brain" or "custom," the program will also use the calibration factor c_{cal0} and T_1^{ref}, specified under **Bias Field c_{cal0}, T_1^{ref} (ms)** to normalize the bias fields and obtain the correct c_{net}^+. If you choose "head," the program will use the hard-coded factor (≈0.87, see line #159) calculated based on the total head size for a typical Philips TG setting. This field is irrelevant if you chose "B1map" as the RF Correction type.

Bias Field c_{cal0}, T_1^{ref} (ms): Calibration factor required to convert the Ψ_1 N4ITK bias field into a c_{net}^+ map. It is based on the assumption that the nominal flip angle should be linearly proportional to the mean flip angle over a specified region, such the skull-stripped brain or the whole head. This factor depends on the transmit gain (TG) setting of the scanner, and also the choice of the RF phase cycle increment ϕ_0, and the amount of MT effects (which depend on the RF pulse durations). The program will first calculate

T_1 using the initial c_{cal0}. Then using the T_1^{ref} (default: 900ms for WM peak), it will calculate the new c_{cal} value and the corrected T_1.

Output maps [M0appB1/M0&T1/T1&T2/M0&T1&T2*/all]: Specifies the choice of final parametric maps the program will output. If you want to output fewer maps than is specified by the *calculation* parameter in the *Parameters4MPM* file, you can do it here. Also see the discussion in the next section on how to create the $M_{0,B1}^{app}$, M_0^{app} and T_1^{app} images to estimate the bias fields with N4ITK.

T_2* filter: [median/other/no]: Determines the choice of T_2^* filter used. If "median," the 2D median filter function in Matlab will be applied slice-by-slice, if "other" is selected, the program will assume that the T_2^* is already filtered (in 3D Slicer), and if "no," no filtering will be applied.

FLASH T_{RF2}, bSSFP T_{RF2}, ϕ_0 (deg): Duration of the RF pulse (in ms) of the FLASH/bSSFP sequences at the higher flip angles, used to correct the final T_2 maps. The correction is not really needed for FLASH, unless you are doing ultra-short TE imaging in bone. If infinitesimal pulses are assumed in bSSFP (0 ms), the T_2 may be overestimated by ~7-8 %. The ϕ_0 is required to compute the A, B correction coefficients for non-ideal RF spoiling if using a B_1 map. The default is 50°, while a value of 0 will simply tell the program to omit the correction. The correction coefficients will also be saved within the parameter file.

T_1/T_2/T_2^*clip value (ms): Clipping values of the T_1/T_2^* maps. Note that the input T_2^* maps might have already been clipped, so clipping them to a higher value will not be useful.

T_2* combination: [wave1/wave2]: Choice of T_2^* combination technique, including weighted average based on T_1 yield "wave1," and combination based using the weighted uncertainty in each T_2^* least-squares fit "wave2". If motion artifacts are present, the second two options may yield better final T_2^* maps. Note that the first option does not require the input of variance maps "σ_{T2*}^2" in the next two boxes e.g. "*sigmaT2s1.mha*". Note that if more than 2 T_2^* maps are combined, only the *wave1* option is available.

wave1:

$$T_{2,final}^* = \frac{w_1 T_2^*(\alpha_1) + w_2 T_2^*(\alpha_2)}{w_1 + w_2}, \qquad w_i = \frac{\sin^2(\alpha_i)}{(1 - E_1\cos(\alpha_i))^2}.$$

wave2:

$$T_{2,final}^* = \frac{w_1 T_2^*(\alpha_1) + w_2 T_2^*(\alpha_2)}{w_1 + w_2}, \qquad w_i = 1 \big/ \sigma_{\mathrm{T2i}}^2 .$$

σ_{T2*}^2 maps: Image files containing the voxel-wise uncertainty in the fit of each T_2^* map, if using the *wave2* option above. The arrays must have been co-registered, just like the T_2^* maps. If *wave1* was selected, these two input files are irrelevant and will not be opened.

Calibration mode [manual/auto/WMauto]: This setting determines how the calibration of the c_{net}^+ and M_0 will be performed. If you pick "auto", make sure the T_1^{ref} above is the mid-point between the WM and GM T_1 histogram peaks. If you pick "WMauto" make sure the T_1^{ref} is the WM T_1 peak (~900 ms). Note that the automatic modes require the curve-fitting toolbox in Matlab while the manual mode does not, but will expect user inputs. If you pick the automatic mode, the program will also output the T_1 and

M_0 histograms before the calibration so that the user can make sure the histogram fit looks good. If the fit looks bad, you should re-run the program with a more appropriate choice of c_{cal0}. If testing phantom data, use the manual mode.

How to generate T_1^{app}/M_0^{app}/$M_{0,B1}^{app}$ to obtain Ψ_1/ Ψ_0 /c_{RF}^- bias fields from N4ITK

To generate the *T1app.mha*, *M0app.mha* and *T1_mask.mha* files, (or alternatively R0app.nii and R1app.nii for SPM12) run the *MPM_main_v1* program with the following two options: **Maps: M0&T1**, and **RF Correction type: no.** The code will generate the two maps as well as the mask and the following 4 figures will pop up:

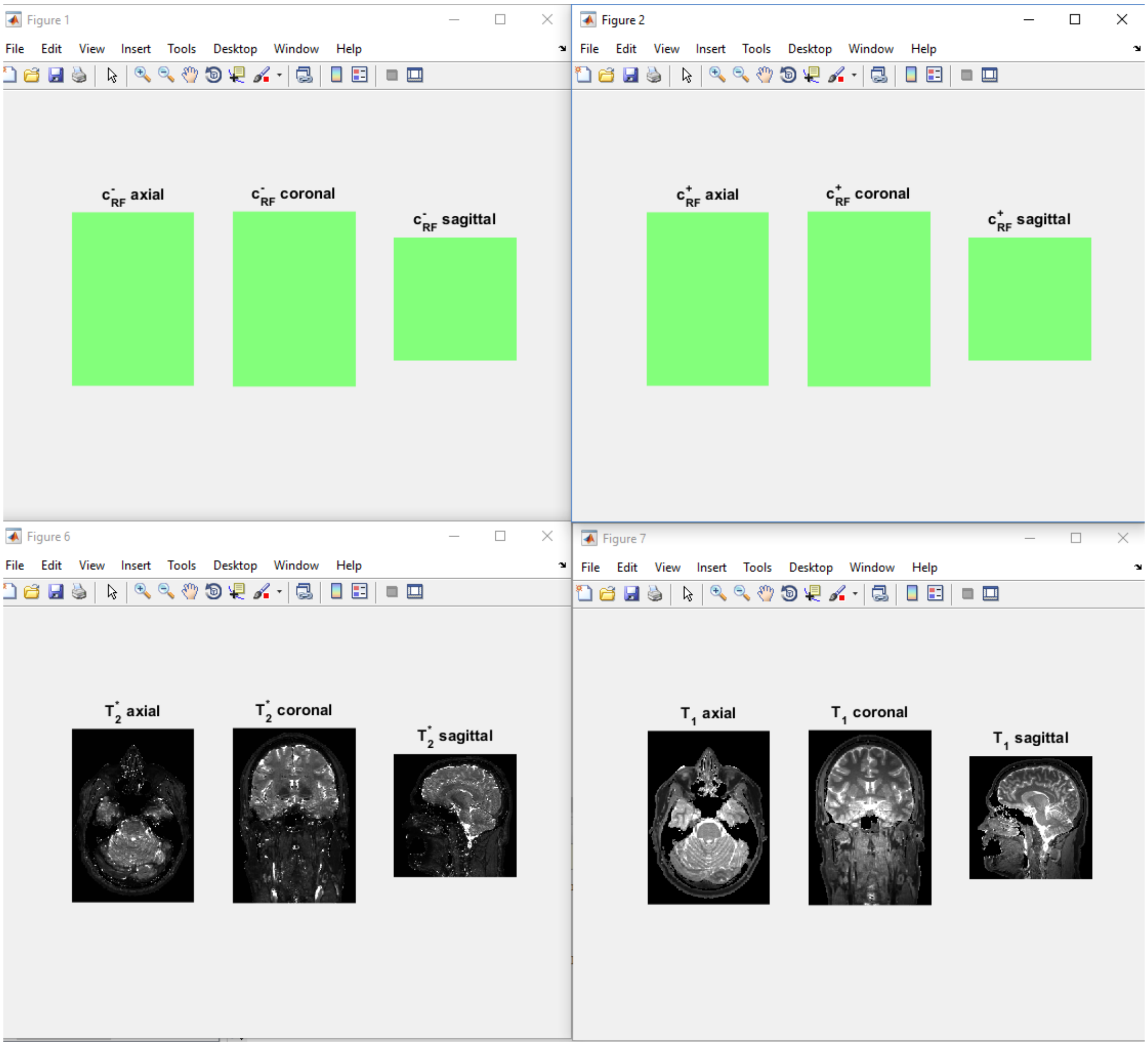


Figure 2: Output figures that are generated when creating the T1app and M0app images. The c_{RF}^+ and c_{RF}^- fields are unity because they have not been estimated yet. Note that in this case the T1 is actually the T1app.

Open the three files in 3D Slicer and input the following parameters or input/output files in the correct fields within the module *N4ITK MRI Bias correction*:

Input Image: **T1app (or M0app)**,

Mask Image: **T1_mask (or T1M0_mask)**

Output Volume: **T1corr (or none)**

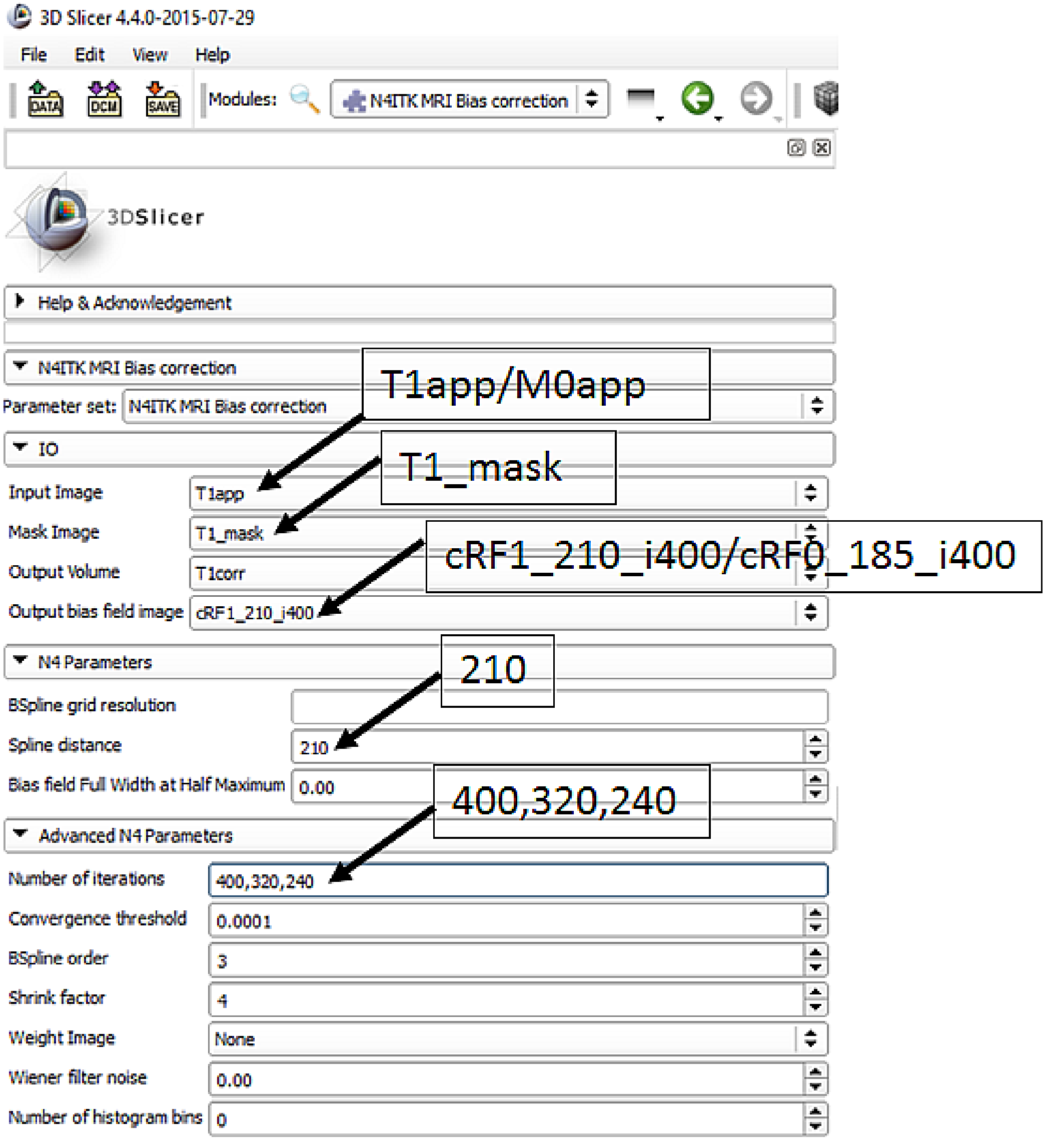
Output bias field image: **cRF1_210_i400** (or **cRF0_185_i400** for M0app)

BSpline Grid resolution: (delete the default value of 1,1,1)

Spline distance: **210** (or **185** for M0app)

Number of iterations: **400, 320, 240**

The other parameters can be left in their default values. The optimal spline distance of 210/185mm and number of iterations (400, 320, 240) for fitting the Ψ_{T1} and Ψ_{M0} bias fields were pre-determined experimentally from 3T MRI data of 8 volunteers (4 males/4 females). The execution time is usually about 2 min per bias field. Save the two output bias fields **cRF1_210_i400** and **cRF0_185_i400** as mha or nii/nii.gz files in your MATLAB folder (rather than the default of ".nrrd"). To obtain an even better final M_0 map, you may choose to run the code with **RF Correction type='BiasField,'** and the two computed bias fields: **cRF1_210_i400** and **cRF0_185_i400.** If you select also the hard-coded parameter in the code on line 15**:** `New_T1M0_mask='yes',` a better mask *T1M0_mask.mha* that also excludes adipose in addition to CSF and air will be generated. You can use this mask to obtain a better Ψ_0 bias field (labelled **cRF0_185_i400b** in the sample data provided). In general, this mask will not really improve the quality/accuracy of the Ψ_{T1} bias field.



Parameter set: N4ITK MRI Bias correction

Input Image: T1app

Mask Image: T1_mask

Output Volume: T1corr

Output bias field image: cRF1_210_i400

BSpline grid resolution:

Spline distance: 210

Bias field Full Width at Half Maximum: 0.00

Number of iterations: 400,320,240

Convergence threshold: 0.0001

BSpline order: 3

Shrink factor: 4

Weight Image: None

Wiener filter noise: 0.00

Number of histogram bins: 0

T1app/M0app

T1_mask

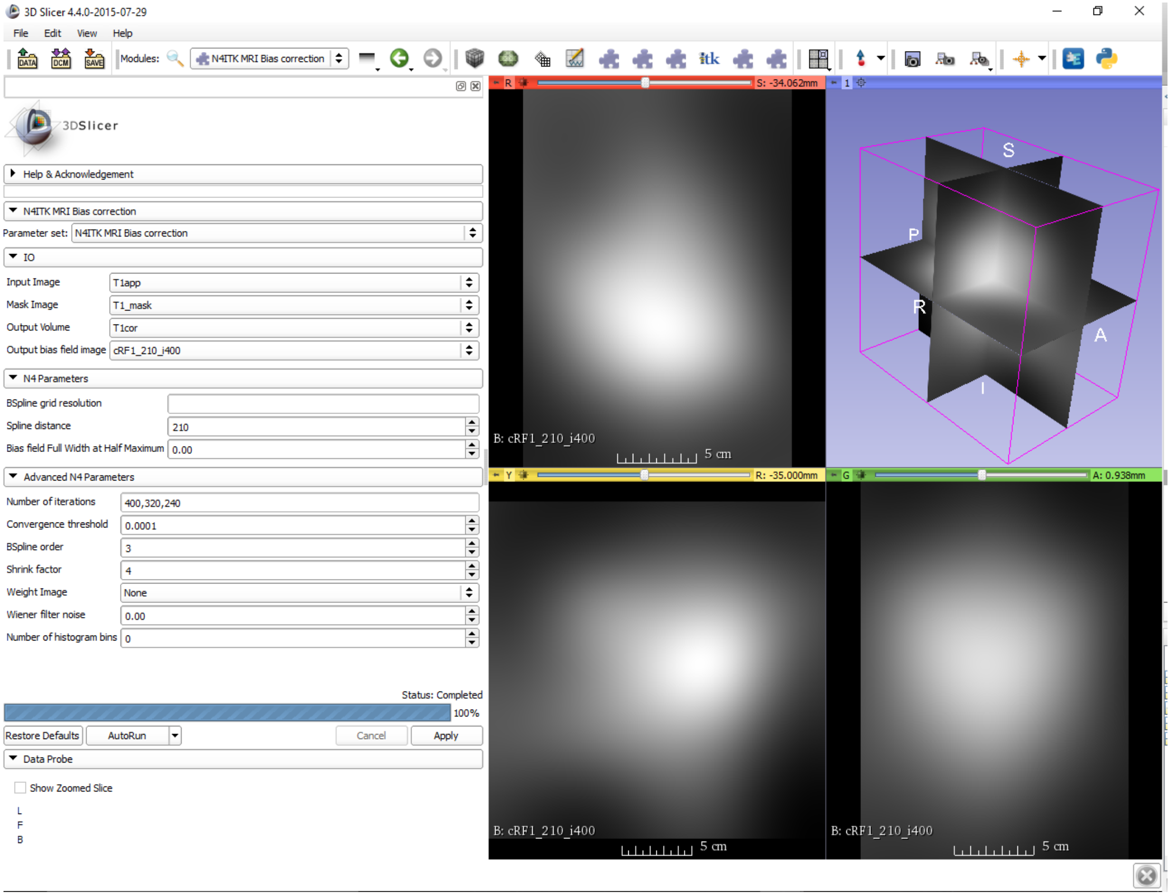
cRF1_210_i400/cRF0_185_i400

210

400,320,240

Figure 3: N4ITK MRI bias correction module in 3D Slicer, showing the correct input parameters needed to generate the two bias fields.

After N4ITK is finished fitting the bias field, it output a field that looks like the following:



N4ITK MRI Bias correction

Parameter set: N4ITK MRI Bias correction

IO

Input Image T1app

Mask Image T1_mask

Output Volume T1cor

Output bias field image cRF1_210_i400

N4 Parameters

BSpline grid resolution

Spline distance 210

Bias field Full Width at Half Maximum 0.00

Advanced N4 Parameters

Number of iterations 400,320,240

Convergence threshold 0.0001

BSpline order 3

Shrink factor 4

Weight Image None

Wiener filter noise 0.00

Number of histogram bins 0

Status: Completed

100%

Restore Defaults | AutoRun | Cancel | Apply

Data Probe

☐ Show Zoomed Slice

L
F
B

Figure 4: Output bias field (Ψ_1) in 3D slicer.

How to Normalize the Proton-Density M_0 Map

To obtain a normalized M0 map as final output, you must set the following hard-coded parameter on line 14: `M0_calibration='yes';` Now if you run the program in manual calibration mode, the following window will appear (shown in **Figure 5**):

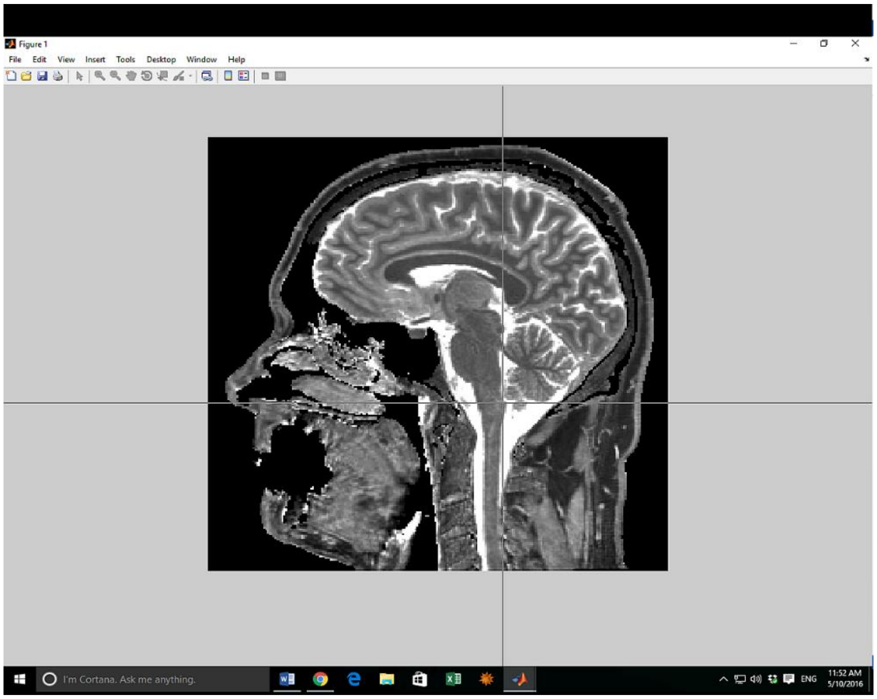


Figure 1
File Edit View Insert Tools Desktop Window Help
I'm Cortana. Ask me anything.
ENG
11:52 AM
5/10/2016

Figure 5: Sagittal brain image of the T1 map with cross-hairs to select a cut-off slice.

Select a cut-off location (only horizontal axis is relevant) to exclude the image slices of the neck. This will help to exclude muscle and fat tissues from the histogram calculation. Next, the following figure with crosshairs will pop up (**Figure 6**):

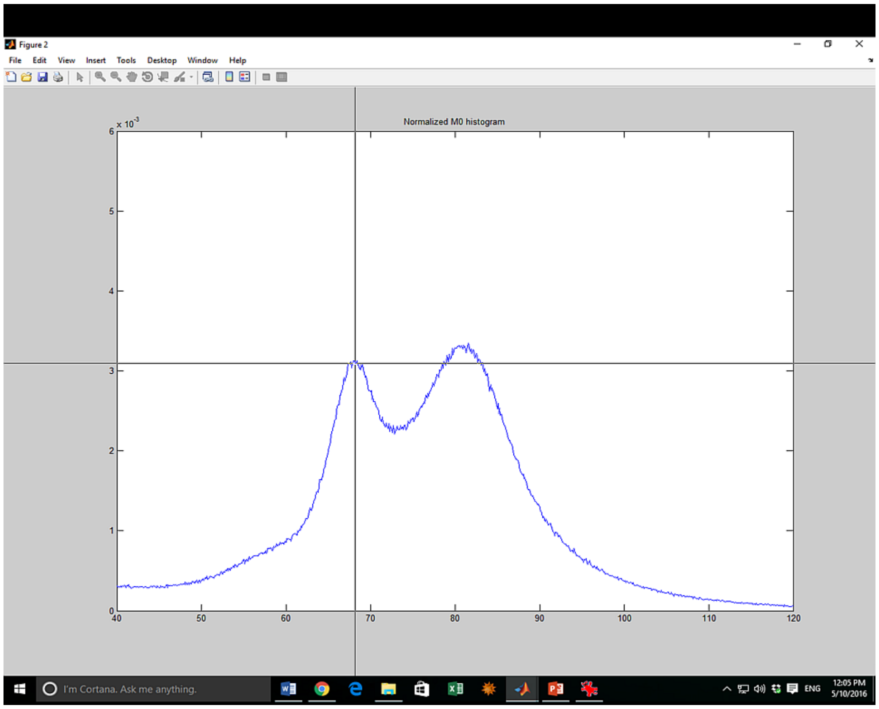


Figure 6: Proton-density histogram before normalization, showing WM (left) and GM (right) peaks.

Click on the white matter peak (to the left) and then on the grey matter peak (to the right). The program will then normalize the M_0 map to the midpoint between the two peaks and multiply by 76%, since the mid-point should lie at 76% proton-density. For an agar phantom, you should see only a single peak in the histogram and you would normalize it at 100% instead (see line ~1004 of the program).

When you run MPM_main_v1 again for the **volunteer dataset1** with **Output maps: M0&T1&T2*** and **RF Correction Type: BiasField**, the following 5 figures should appear.

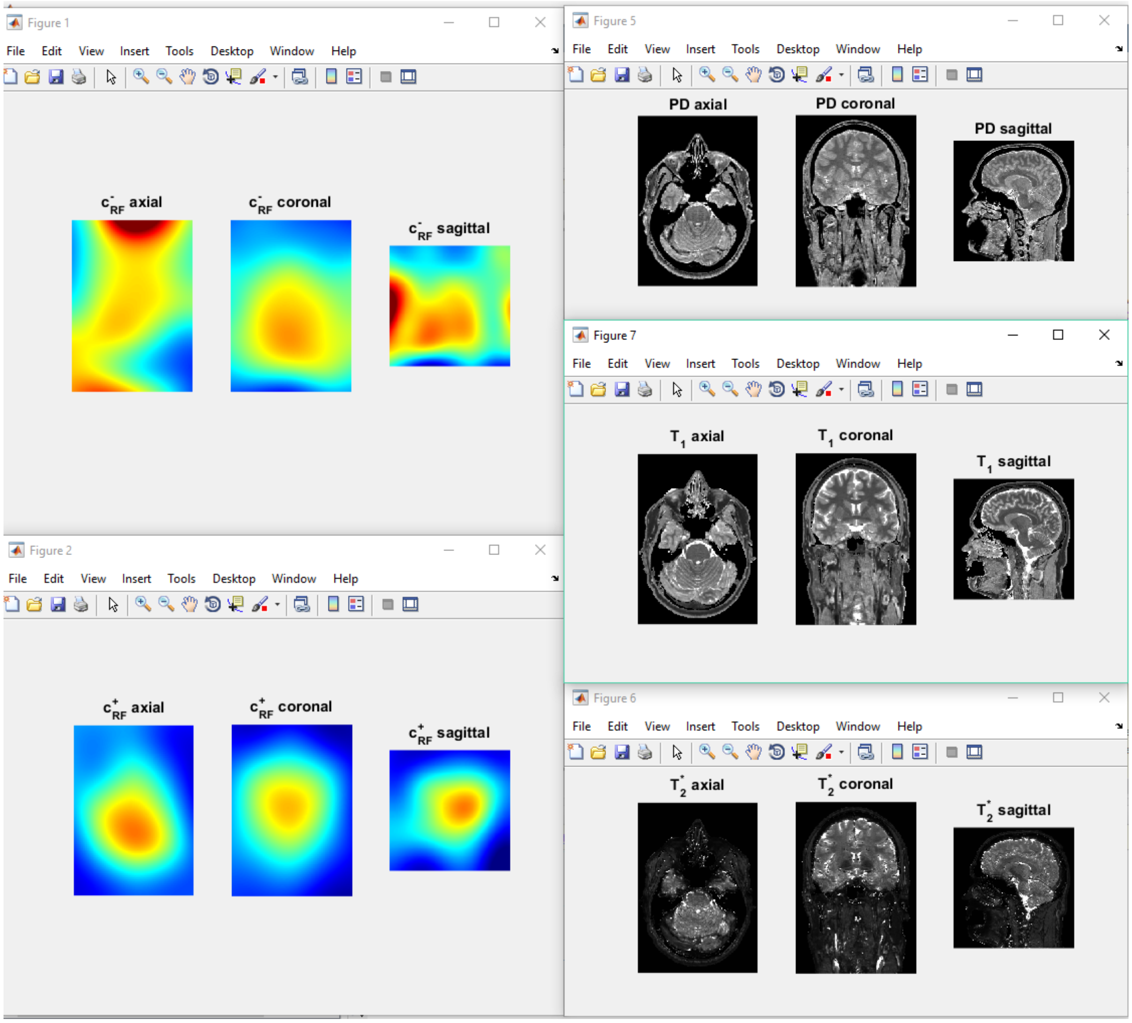


Figure 7: Example of parametric maps obtained for the **volunteer dataset1**.

How to register the datasets in 3D Slicer

Most patients or healthy volunteers will move their head by ~0.2 mm to ~2 mm over a 30 min MRI examination. Even if they stand very still, vibration of the bed from the acoustic noise can cause a gradual shift in the head position during the examination. Consequently, it is crucial to co-register all the images acquired across the different pulse sequences prior to curve-fitting. Even a small shift of <0.5mm may result in a noticeable deterioration of the final parametric maps. I recommend performing the image registration in 3D slicer, but you can always use a different software if you prefer, (such as FSL). The registration is the most time-consuming process of the MPM pipeline, especially if you want to fine-tune your results with deformable image registration (which is more time-consuming). Registration of the bSSFP images can also be especially challenging, since the band-artifacts tend to violate the assumption of mutual information. Nevertheless, this section will provide robust parameters to use to obtain good final results, even in the case of bSSFP images.

When performing image registration, the scientist must decide which image remains “fixed” and which ones are “moving.” The moving image(s) is/are thus registered to the fixed image. The sample data contains the following image files that you will have to open first in 3D Slicer:

M0_1.mha

ME1.mha

ME2.mha

ME1MT.mha

ME3MT.mha

T2s1.mha

T2s2.mha

T2s3.mha

sigmaT2s_1.mha

sigmaT2s_2.mha

sigmaT2s_3.mha

The **M0_1** image is simply the first echo of the PD-weighted FLASH, while **ME1** is a MERGE of this same dataset. First, use **M0_1** to create an image mask for the image registration. To do this, go to the module entitled *Foreground masking (BRAINS)*. Set **M0_1** as “Input image volume,” set “Output Mask->Create new Volume as” and name it ***mask0***. Unclick the box *Mask Output.* Now change the following parameters from their default values:

Otsu Correction Factor: **0.3**,

Closing Size: **5**

ROIAuto Dilate Size: **4**

Output Image Pixel Type: **int**

Click on “apply” to run the module and obtain *mask0*. Now repeat the previous step to create a second mask called mask1, but with the following 3 different settings: *Otsu Correction Factor: 0.6*, *Closing size: 0* and *ROIAuto Dilate Size: 0*. The first mask mask0 will be used in the image registration, while mask1 will be used to remove the bias field in **M0_1** to create **mask_im.mha**.

Change the module to *General Registration (BRAINS)*. Enter the following input/outputs and parameters within the following sections:

Input Images

Fixed Image Volume: **ME1**

Moving Image Volume: **ME2**

Percentage of Samples: **0.2**

Output Settings

Slicer Linear Transform: **ME1_ME2_trans**

Output Image Volume: **ME2_reg**

Registration Phases

Rigid (6DOF): **(check-marked)**

Image Mask and Pre-Processing

ROI **(check-marked)**

(ROI Masking input fixed): **mask0**

Advanced Output Settings

Interpolation mode: **Windowed Sinc, (or B-spline works well too).**

Expert-only Parameters

Histogram bin count: **128**

Click on "apply" to run the module. This will register the **ME2** image to the **ME1** image and also save the transform. You can then use the transform to resample **T2s2** and **sigmaT2s_2** to the same registration as **ME2_reg** using the *Resample Scalar/Vector/DWI Volume* module (to be shown later).

Do the same as above to register **ME3MT** (moving image) to **ME1MT** (fixed image). You can name the transform **ME1_ME3_trans**.

To register the bSSFP images, use the following settings:

Input Images

Fixed Image Volume: ***ME2_reg***

Moving Image Volume: ***bffe1_90, etc …***

Percentage of Samples: ***0.05 (5%)***

Output Settings

Slicer Linear Transform: ***bffe1_90_trans, etc …***

Output Image Volume: ***bffe1_90_reg, etc …***

Registration Phases

Rigid (6DOF): ***(check-marked)***

Image Mask and Pre-Processing

ROI ***(check-marked)***

(ROI Masking input fixed): ***mask0***

Advanced Output Settings

Interpolation mode: ***Windowed Sinc***

Expert-only Parameters

Histogram bin count: ***50***

The smaller percentage of samples used (5% rather than 20%), as well as the lower number of histogram bins (50 rather than 128) will help prevent the band artifacts from driving the registration into a local minimum, since band artifacts violate the assumption of mutual information. Always use ME2_reg as the fixed image, rather than ME1, since ME2_reg has more mutual information (especially CSF) with bSSFP.

To create the mask_im.mha image, go to the *N4ITK Bias correction* module and use the following input/output files and parameters:

IO

Input Image: **M0_1**

Mask Image: **mask1**

Output Volume: **mask_im**

N4 Parameters

Spline Distance: **185**

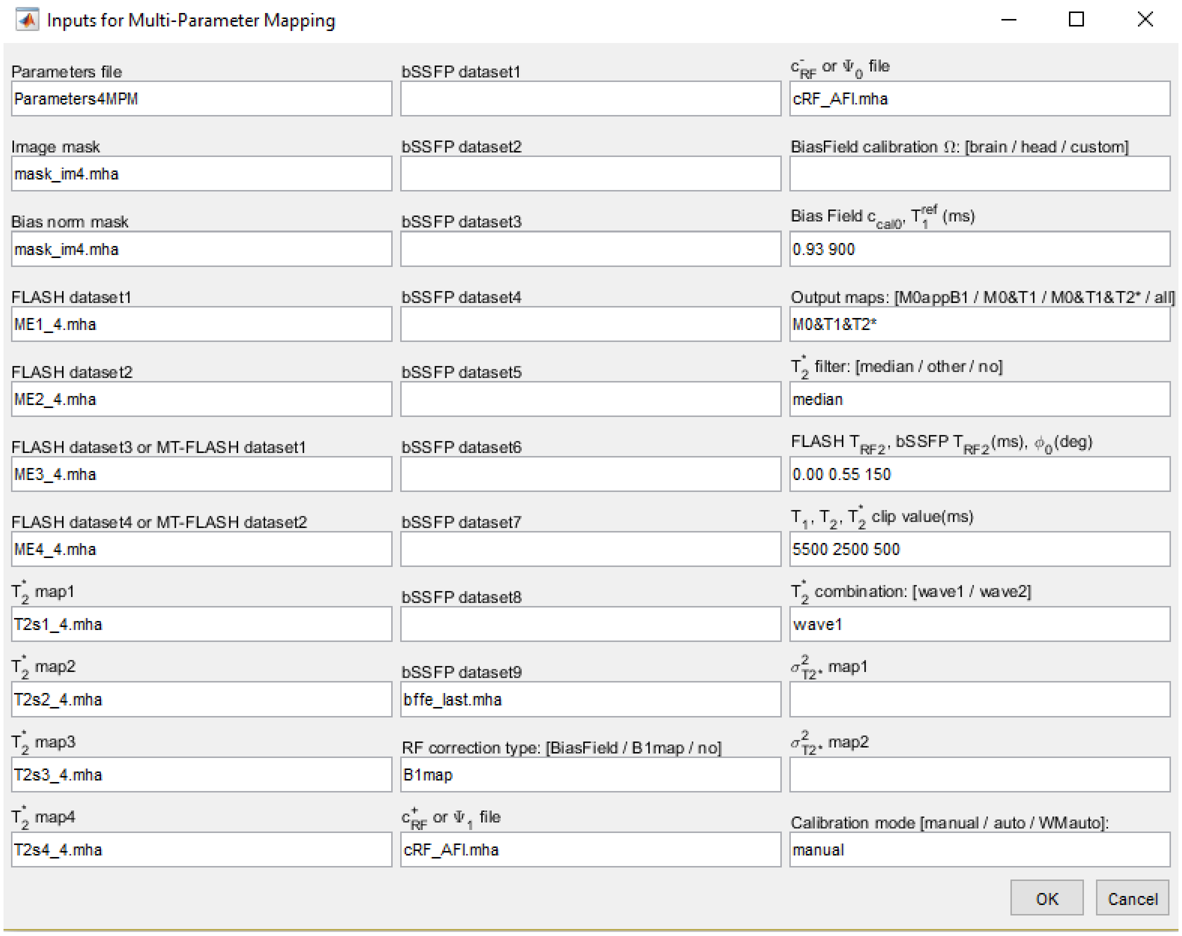
Advanced N4 Parameters

Number of iterations: **400, 320, 240**

The mask_im.mha image will have fewer disconnected pixels if you also filter it using the *Gradient Anisotropic Diffusion* module using a *conductance* of 1.00, 6 *iterations* and the default time step of 0.0625.

Sample Examples

The MPM toolbox can fit up to 4 SPGR flip angles when mapping M0, T1 and T2*. If you look at the sample data provided within the folder "Phantom dataset with 4 flip angles," you will see 4 ME datasets and 4 T2* maps. If you correctly use the following input parameters:



Inputs for Multi-Parameter Mapping

Parameters file	bSSFP dataset1	c^{-}_{RF} or Ψ_0 file
Parameters4MPM		cRF_AFI.mha
Image mask	bSSFP dataset2	BiasField calibration Ω: [brain / head / custom]
mask_im4.mha		
Bias norm mask	bSSFP dataset3	Bias Field c_{cal0}, T_1^{ref} (ms)
mask_im4.mha		0.93 900
FLASH dataset1	bSSFP dataset4	Output maps: [M0appB1 / M0&T1 / M0&T1&T2* / all]
ME1_4.mha		M0&T1&T2*
FLASH dataset2	bSSFP dataset5	T_2^* filter: [median / other / no]
ME2_4.mha		median
FLASH dataset3 or MT-FLASH dataset1	bSSFP dataset6	FLASH T_{RF2}, bSSFP T_{RF2}(ms), ϕ_0(deg)
ME3_4.mha		0.00 0.55 150
FLASH dataset4 or MT-FLASH dataset2	bSSFP dataset7	T_1, T_2, T_2^* clip value(ms)
ME4_4.mha		5500 2500 500
T_2^* map1	bSSFP dataset8	T_2^* combination: [wave1 / wave2]
T2s1_4.mha		wave1
T_2^* map2	bSSFP dataset9	σ^2_{T2*} map1
T2s2_4.mha	bffe_last.mha	
T_2^* map3	RF correction type: [BiasField / B1map / no]	σ^2_{T2*} map2
T2s3_4.mha	B1map	
T_2^* map4	c^{+}_{RF} or Ψ_1 file	Calibration mode [manual / auto / WMauto]:
T2s4_4.mha	cRF_AFI.mha	manual

Figure 8: Required input parameters for the example "Phantom dataset with 4 flip angles." Since the blank inputs are not needed here, unused file names left within those fields will not cause any error. Note that the RF phase cycle of the FLASH datasets was ϕ_0=150 deg in this case.

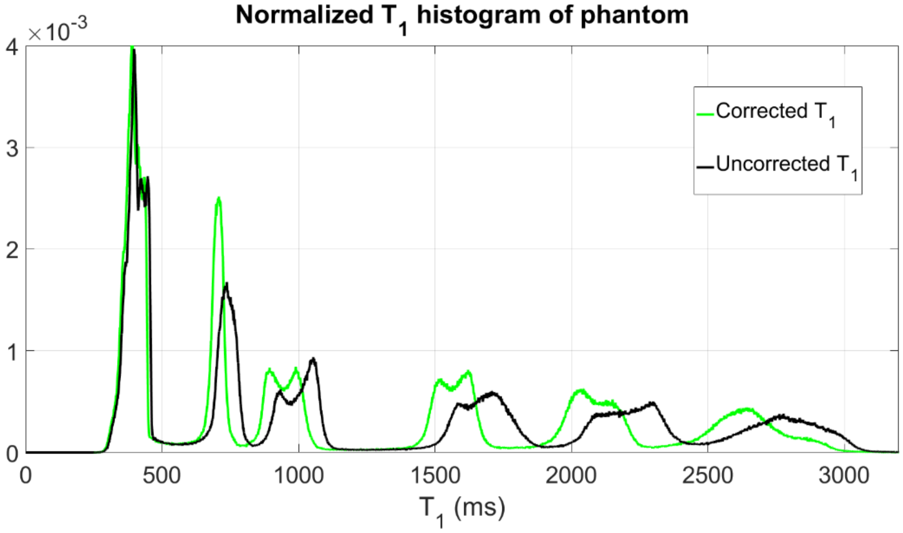


Figure 9: Phantom T_1 histograms with and without correction for non-ideal RF spoiling. The correction has little impact on very short T_1, but a significant impact at long T_1 and T_2 values.

Figure 9 compares the T_1 histograms of the phantom with and without the correction for imperfect spoiling. If you set ϕ_0=0, you should get the following parametric maps:

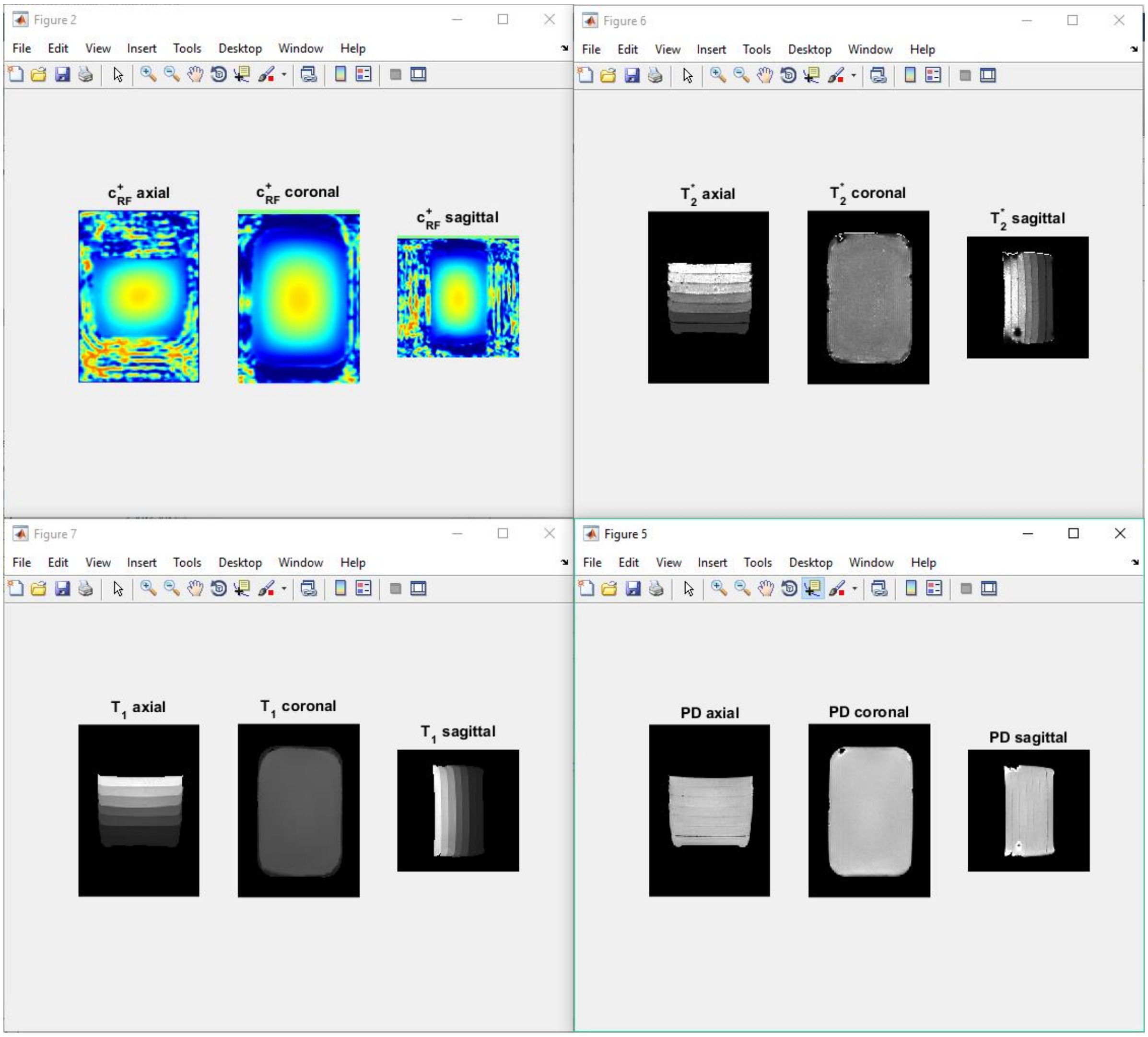
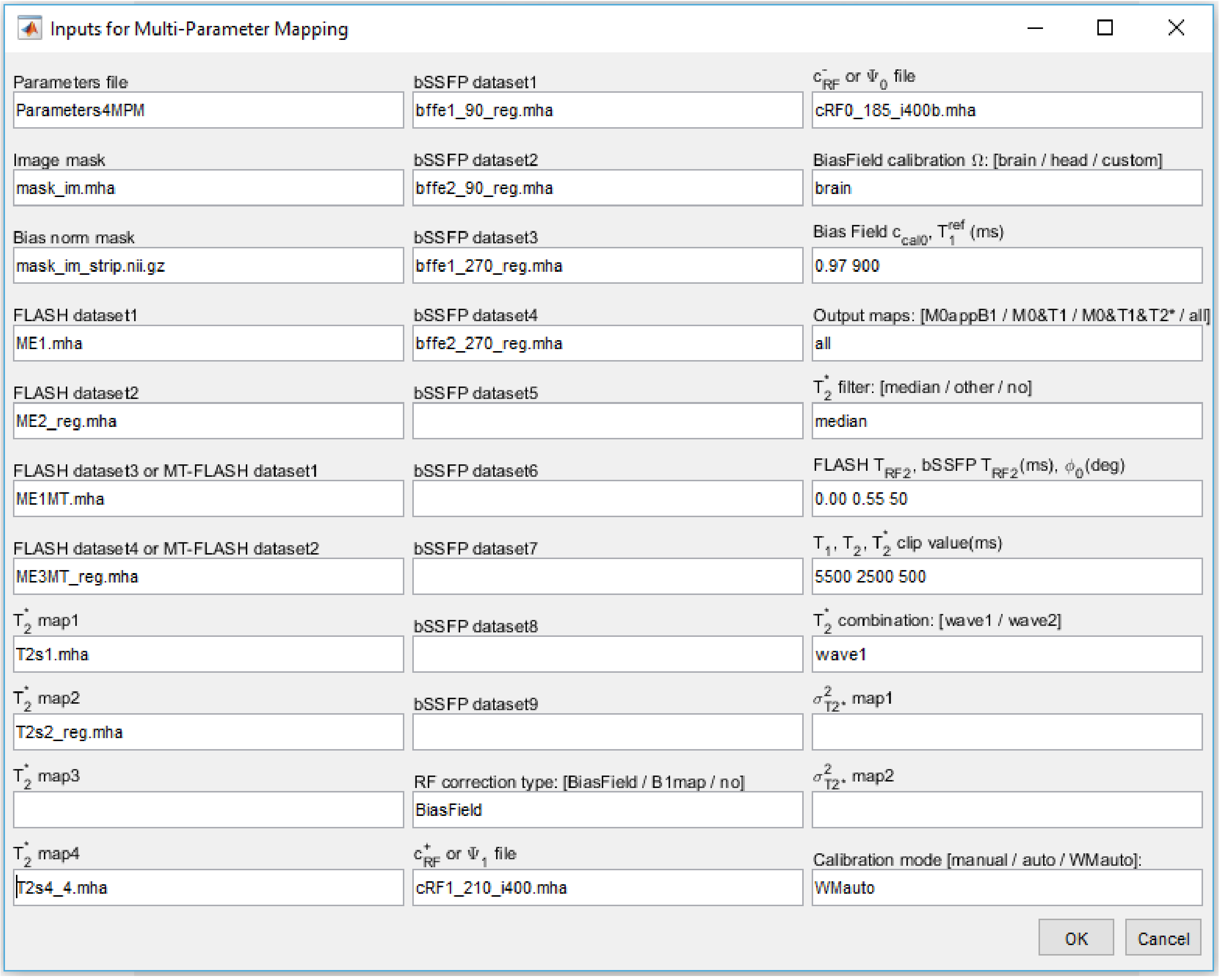


Figure 10: MPM example where 4 8-echo SPGR sequences were acquired at varying flip angles (3.5, 9, 20 and 40°) as well as an AFI sequence to map PD, T1 and T2*.

In this example, since a B_1 map (from AFI) is provided, it is assumed that $c_{RF}^{+} \approx c_{RF}^{-}$ when mapping the proton-density. It turns out that this assumption holds rather well in this example because the phantom has low conductivity (no salt added) and is fairly small (<2L) with respect to the body coil. However, when performing MPM on a real human brain, this assumption usually holds rather poorly and N4ITK is required to correctly estimate the c_{RF}^{-} field.

The toolbox can map up to 5 parameters (PD, T_1, T_2^{*}, MT_{sat} and T_2) and 2 bias fields (c_{RF}^{+} and c_{RF}^{-}). The volunteer dataset2 provides an example. If you correctly input the following file names and parameters:



Inputs for Multi-Parameter Mapping

Parameters file Parameters4MPM	bSSFP dataset1 bffe1_90_reg.mha	c^{-}_{RF} or Ψ_0 file cRF0_185_i400b.mha
Image mask mask_im.mha	bSSFP dataset2 bffe2_90_reg.mha	BiasField calibration Ω: [brain / head / custom] brain
Bias norm mask mask_im_strip.nii.gz	bSSFP dataset3 bffe1_270_reg.mha	Bias Field c_{cal0}, T_1^{ref} (ms) 0.97 900
FLASH dataset1 ME1.mha	bSSFP dataset4 bffe2_270_reg.mha	Output maps: [M0appB1 / M0&T1 / M0&T1&T2* / all] all
FLASH dataset2 ME2_reg.mha	bSSFP dataset5	T_2^* filter: [median / other / no] median
FLASH dataset3 or MT-FLASH dataset1 ME1MT.mha	bSSFP dataset6	FLASH T_{RF2}, bSSFP T_{RF2}(ms), ϕ_0(deg) 0.00 0.55 50
FLASH dataset4 or MT-FLASH dataset2 ME3MT_reg.mha	bSSFP dataset7	T_1, T_2, T_2^* clip value(ms) 5500 2500 500
T_2^* map1 T2s1.mha	bSSFP dataset8	T_2^* combination: [wave1 / wave2] wave1
T_2^* map2 T2s2_reg.mha	bSSFP dataset9	σ^2_{T2*} map1
T_2^* map3	RF correction type: [BiasField / B1map / no] BiasField	σ^2_{T2*} map2
T_2^* map4 T2s4_4.mha	c^{+}_{RF} or Ψ_1 file cRF1_210_i400.mha	Calibration mode [manual / auto / WMauto]: WMauto

Figure 11: Input file names and parameters for the example within the folder “volunteer dataset2”.

You should see the following parametric maps and fields appear:

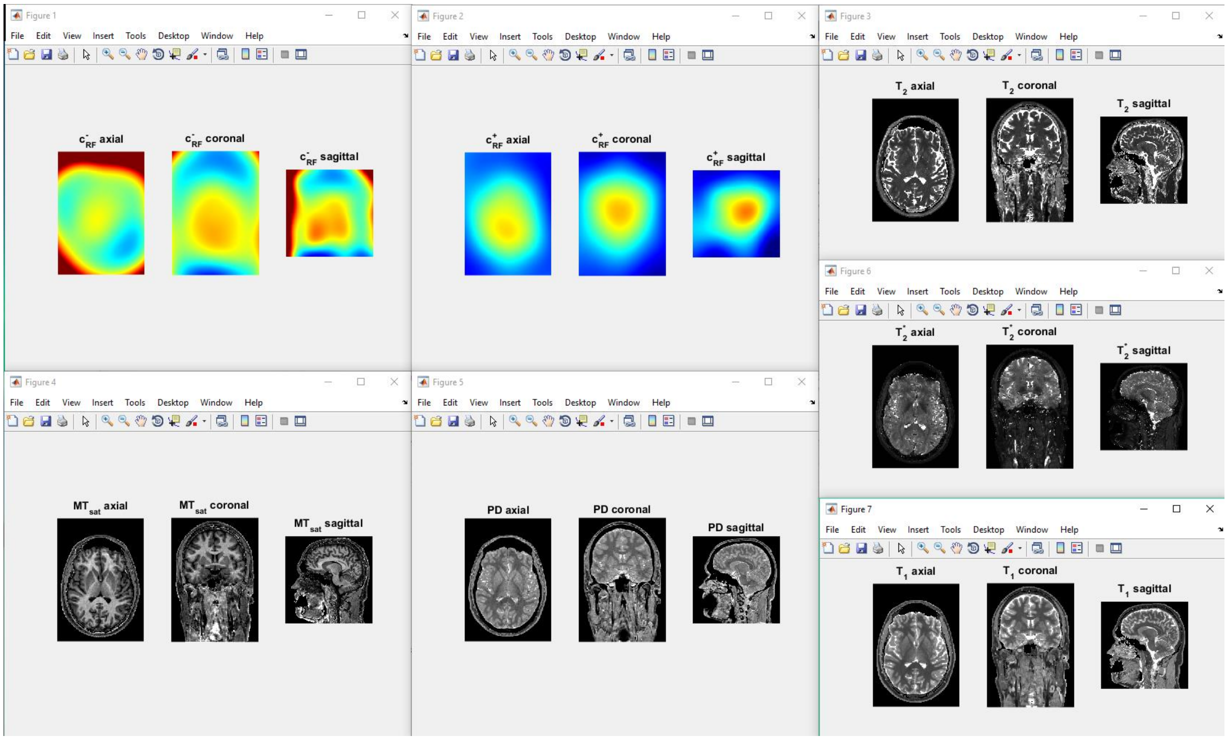


Figure 12: MPM results on an in vivo brain volunteer.

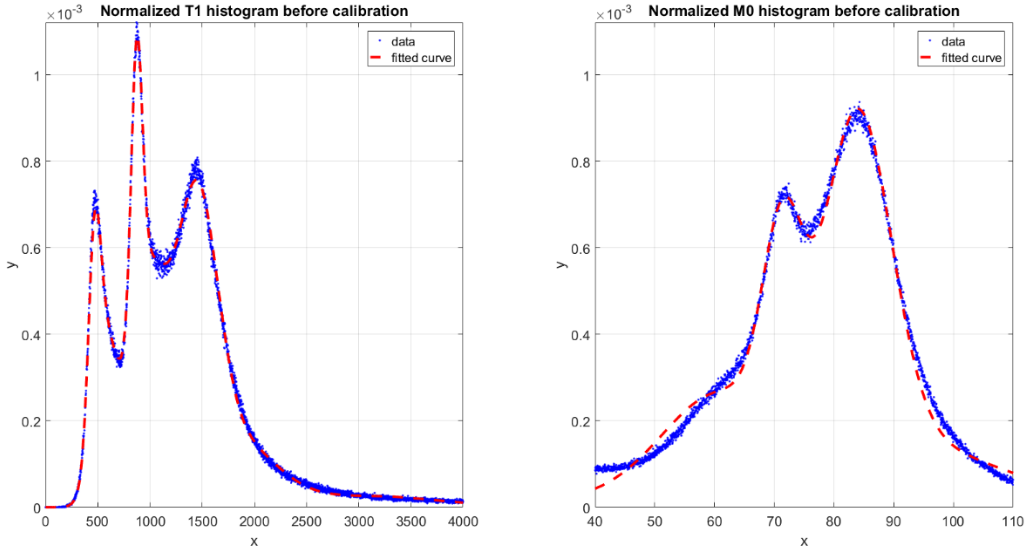
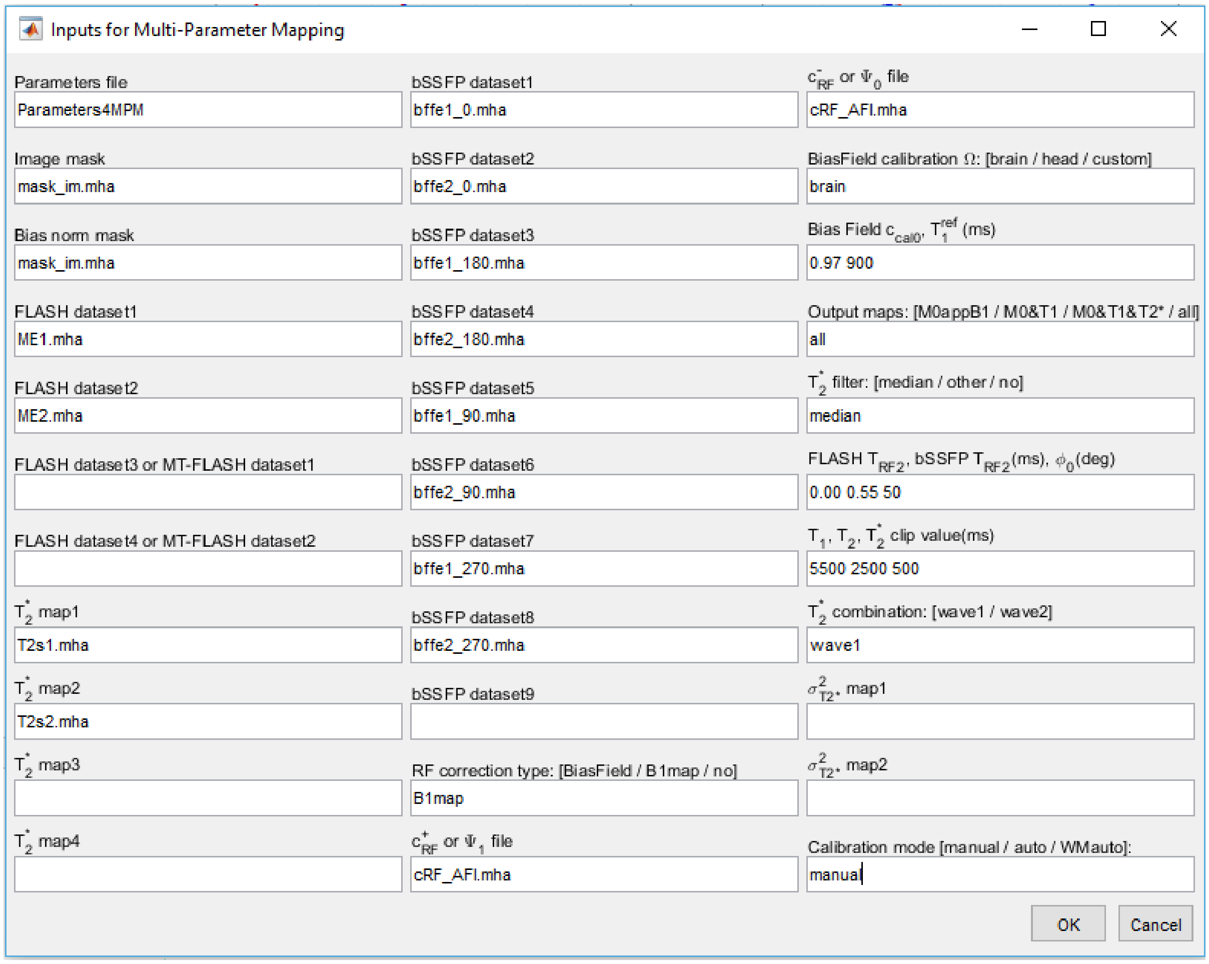


Figure 13: Example of the fitted T_1 and M_0 histograms of the volunteer.

Finally, the fourth example entitled "phantom dataset with 4 bSSFP offsets" shows the effect of solving for the T_2 using different combination of phase offsets. Since a minimum of two phase offsets by two flip angles are required, acquiring up to 4 phase offsets (by two flip angles, thus 8 bSSFP datasets) can help improve the final quality and SNR of the T_2 maps. The different resulting T_2 maps are labelled T2 (RSS combination), T2_anal, T2_anal1 and T2_anal2. To test this example, just input the images files and parameter names as shown below:



Inputs for Multi-Parameter Mapping

Parameters file: Parameters4MPM

Image mask: mask_im.mha

Bias norm mask: mask_im.mha

FLASH dataset1: ME1.mha

FLASH dataset2: ME2.mha

FLASH dataset3 or MT-FLASH dataset1:

FLASH dataset4 or MT-FLASH dataset2:

T_2^* map1: T2s1.mha

T_2^* map2: T2s2.mha

T_2^* map3:

T_2^* map4:

bSSFP dataset1: bffe1_0.mha

bSSFP dataset2: bffe2_0.mha

bSSFP dataset3: bffe1_180.mha

bSSFP dataset4: bffe2_180.mha

bSSFP dataset5: bffe1_90.mha

bSSFP dataset6: bffe2_90.mha

bSSFP dataset7: bffe1_270.mha

bSSFP dataset8: bffe2_270.mha

bSSFP dataset9:

RF correction type: [BiasField / B1map / no]: B1map

c_{RF}^{+} or Ψ_1 file: cRF_AFI.mha

c_{RF}^{-} or Ψ_0 file: cRF_AFI.mha

BiasField calibration Ω: [brain / head / custom]: brain

Bias Field c_{cal0}, T_1^{ref} (ms): 0.97 900

Output maps: [M0appB1 / M0&T1 / M0&T1&T2* / all]: all

T_2^* filter: [median / other / no]: median

FLASH T_{RF2}, bSSFP T_{RF2}(ms), ϕ_0(deg): 0.00 0.55 50

T_1, T_2, T_2^* clip value(ms): 5500 2500 500

T_2^* combination: [wave1 / wave2]: wave1

σ^2_{T2*} map1:

σ^2_{T2*} map2:

Calibration mode [manual / auto / WMauto]: manual

Figure 14: Input parameters for the "Phantom dataset with 4 bSSFP offsets" example. Notice the order of the phase offsets: 0, 180, 90, 270 deg.

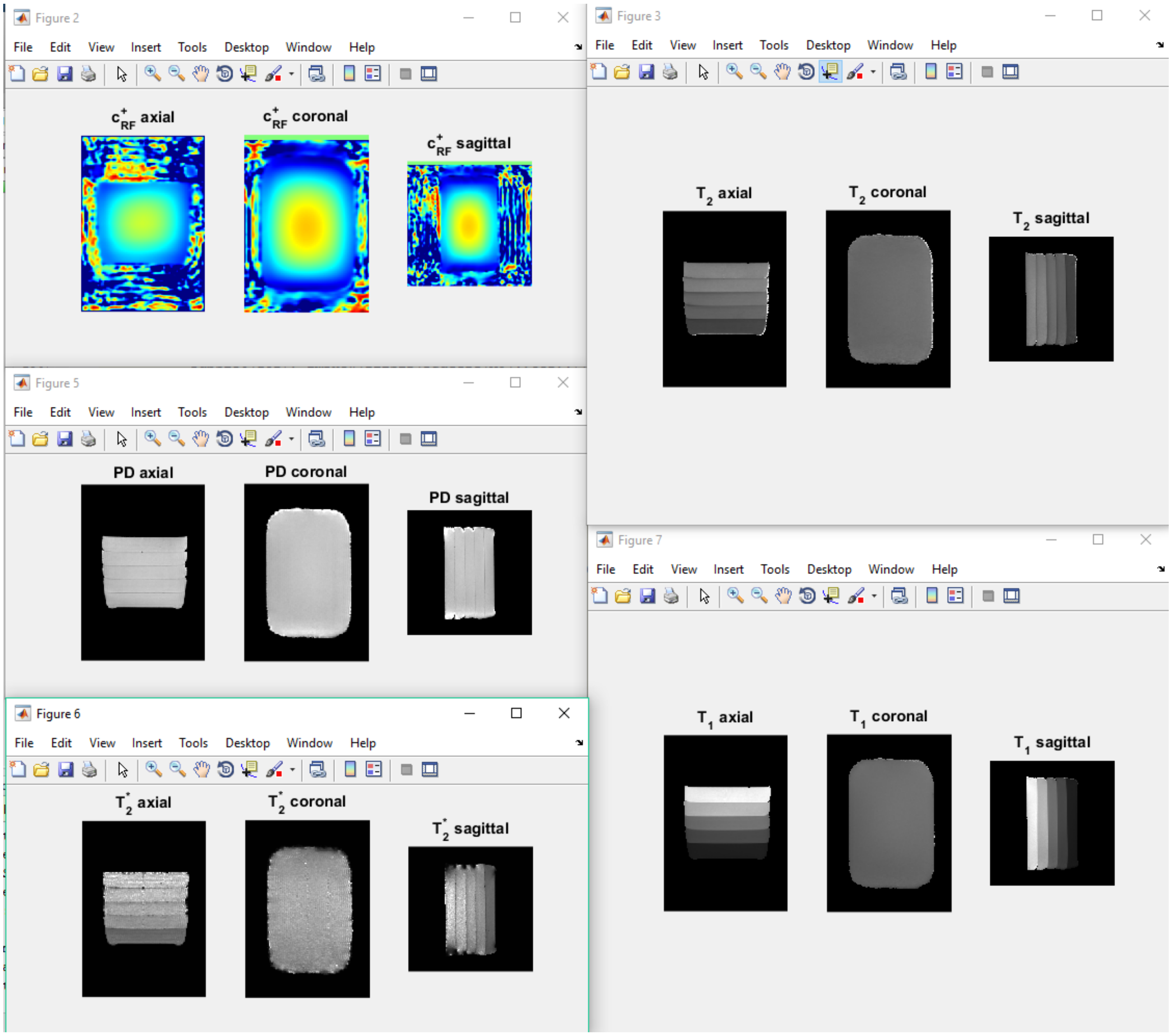


Figure 15: Resulting figures for the last example.

Contact Information

For any question(s), or if you would like me to send you additional sample datasets for testing, feel free to contact me via email: *jutras@ualberta.ca*.