SEPIA Documentation

Release 1.2.0

Kwok-Shing Chan

Oct 11, 2022

Getting started

Welcome to the SEPIA documentation! Here you can find all the documents related to SEPIA.

SEPIA provides a graphical user interface (GUI) in Matlab to build data processing pipelines related to quantitative susceptibility mapping (QSM).

SEPIA is designed to solve two issues we encountered in QSM:

1. Using algorithms from toolboxes developed by different research groups,
2. Having a platform such that we can quickly adjust (and remember) parameters of each algorithm.

The purpose of SEPIA is to provide a platform for easy access to different QSM processing methods in the field. For **SEPIA** v1.1.1, it provides interfaces to use the following toolboxes freely available online for academic purposes, i.e.

1. MEDI toolbox (updated Jan 15, 2020)
2. STI Suite (v3.0)
3. FANSI toolbox (v3.0, released on 2021.10.15, i.e., commit b6ac1c9e)
4. SEGUE (accessed 12 September 2019)
5. MRI susceptibility calculation methods (accessed 12 September 2019)
6. MRITOOLS (v3.5.5) (2022-Oct-11: v3.5.6 passed)

Through SEPIA we hope researchers who are not coming from the field could also be able to use QSM for their research. For those more experienced researchers, we also hope this tool can simplify your workflow.

Warning: We recommand to use only one specific SEPIA version throughtout a single study (i.e., re-run all the processing with the latest version or do not update SEPIA in the middle of your study) to ensure all data undergoing a consistient processing pipeline.

If you encounter a bug in SEPIA, please report to this GitHub issue page

If you have a more general question reagrding the usgae of SEPIA and/or other QSM questions, please make use of the GitHub Discussion page

CHAPTER 1

Table of Contents

1.1 Installation/Setp up

1.1.1 Prerequisite

To support the fully functionality of **SEPIA**, the following external libraries, which are freely available for academic purposes, are required. You can download these toolboxes/libraries using the following links:

- MEDI toolbox (updated Jan 15, 2020)
- STI Suite (v3.0)
- FANSI toolbox (v3.0, released on 2021.10.15, i.e., commit b6ac1c9e)
- SEGUE (accessed 12 September 2019)
- MRI susceptibility calculation methods (accessed 12 September 2019)
- mritools (v3.5.5)

If you encounter any difficulty to download these toolboxes please let us know by opening a new issue in the GitHub page.

1.1.2 Installation of SEPIA

Once you have all the toolboxes in place, you have to add the directory containing sepia.m, i.e. the SEPIA HOME directory, to your MATLAB PATH

This can be done by: 'Set Path' -> 'Add Folder' -> /your/sepia/directory/ -> 'Save'

Warning: To ensure only the selected algorithm is used for the QSM processing, please do not manually add the paths to the external toolboxes you want to run in SEPIA to the Matlab PATH, the *sepia_addpath* function will do the job for you:).)

or

with MATLAB's command: `addpath('/your/sepia/directory')`

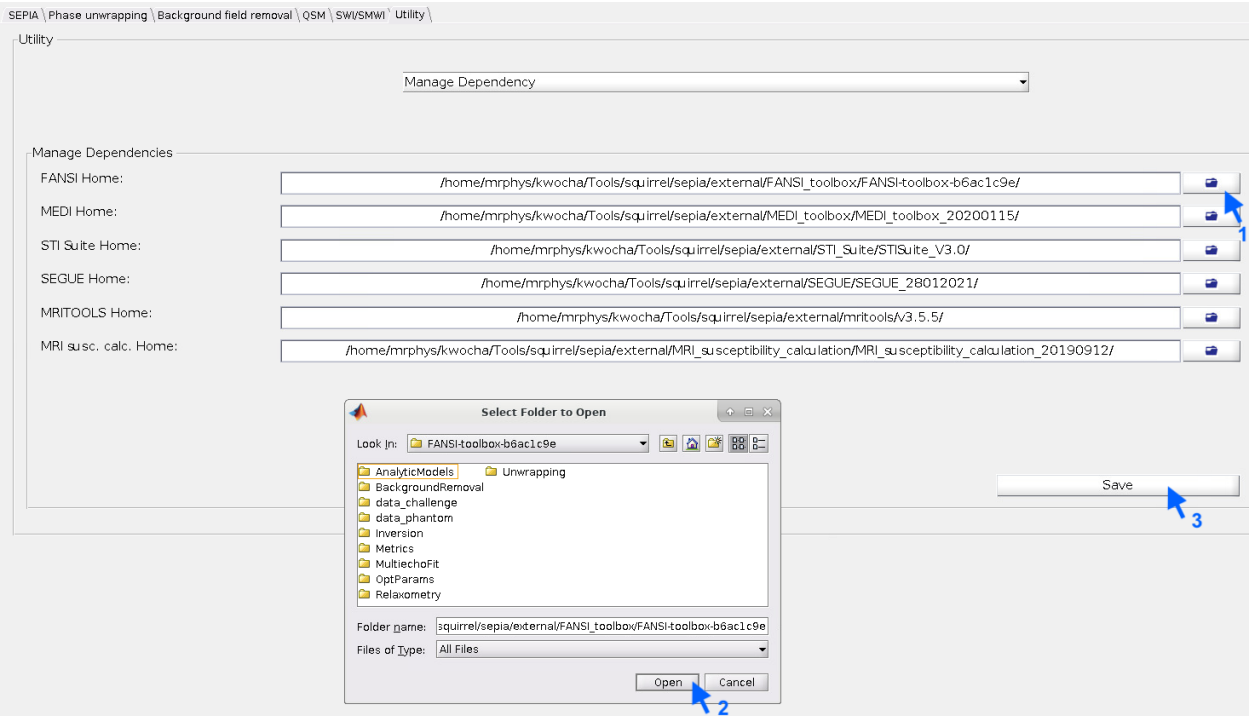
1.1.3 Managing external dependency

SEPIA is a pipeline processing tool focusing on integrating various tools in a single platform. Although SEPIA does provide QSM methods for some basic processing, the majority of the methods supported by SEPIA are from external tool. Users have to download these tools from the URLs provided aboved before they can be run in SEPIA. Once you have all the tools available in your computer, you can use the following options to manage these tools in the SEPIA environment.

Managing external dependency via GUI

You have to specify the directory of each toolbox. From SEPIA v1.0, this can be done on the SEPIA's GUI: simply initialises the GUI using command `sepia`, this should start the GUI. For the first time, you will see some warning messages regarding missing dependencies. Ignore those messages for now.

Navigates to the 'Utility' tab, and select 'Manage Dependency':



1. Use the icon to select the top folder of the tool
2. Click 'Open' to select the folder
3. Click 'Save' once you finish adding all dependencies

Now, restart the SEPIA's GUI by closing the window and re-open it using the `sepia` command in Matlab. You should recieve no more warning messages now if you provide all the required dependencies.

Managing external dependency directly on SpecifyToolboxesDirectory.m

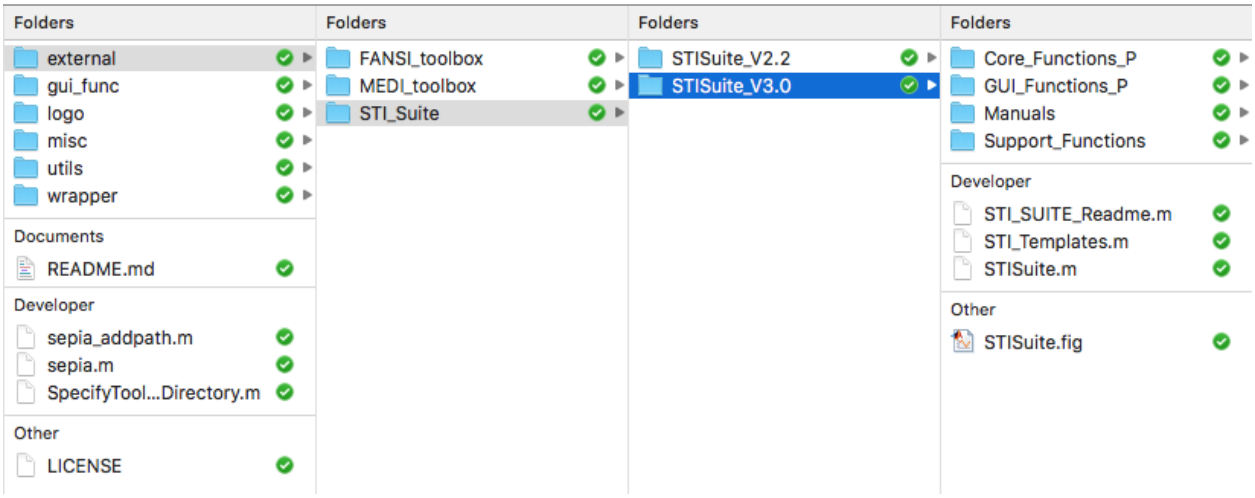
Alternatively, the traditional way of manging dependency in SpecifyToolboxesDirectory.m is still feasible:

```
MEDI_HOME = '/path/to/MEDI/toolbox/';
STISuite_HOME = '/path/to/STISuite/toolbox/';
FANSI_HOME = '/path/to/FANSI/toolbox/';
SEGUE_HOME = '/path/to/SEGUE/library/;'
MRISC_HOME = '/path/to/MRI_susceptibility_calculation/library/;'
MRITOOLS_HOME = '/path/to/MRITOOLS/library/;'
```

Warning: The variable names of the toolboxes' paths are changed from '_dir' to '_HOME' from v0.8. Please update your `SpecifyToolboxesDirectory.m` file accordingly to avoid error.

Warning: The variable 'ROMEO_HOME' is changed to 'MRITOOLS_HOME' in v1.1.1. Please update your `SpecifyToolboxesDirectory.m` file accordingly to avoid error. You can also specify the path to the previous ROMEO executable (but CLEAR-SWI might not work in this case).

For example, I have all my external toolboxes stored under the SEPIA home directory. Additionally, for each toolbox, I have different copies representing different versions when they were published



and here is an example of how is my SpecifyToolboxesDirectory.m defined:

```
% 1. sepcify the toolbox version you want to run
MEDI_version        = 'MEDI_toolbox_20200115';
FANSI_version       = 'FANSI-toolbox-77023b65';
STISuite_version    = 'STISuite_V3.0';
SEGUE_version       = 'SEGUE_28012021';
MRISC_version       = 'MRI_susceptibility_calculation_20190912';
MRITOOLS_version    = 'v3.5.5';

% 2. get the Sepia HOME directory from this script
fullName        = mfilename('fullpath');
SEPIA_HOME      = fileparts(fullName);

% 3. specify the top level of external dependency directory
external_dir    = [SEPIA_HOME filesep 'external' filesep];

% 4. specify the parent directory of each toolbox
MEDI_dir        = [external_dir 'MEDI_toolbox' filesep];
```

```
FANSI_dir       = [external_dir 'FANSI_toolbox' filesep];
STISuite_dir    = [external_dir 'STI_Suite' filesep];
SEGUE_dir       = [external_dir 'SEGUE' filesep];
MRISC_dir       = [external_dir 'MRI_susceptibility_calculation' filesep];
MRITOOLS_dir    = [external_dir 'MRITOOLS' filesep];

% 5. sepcify the final destination of each toolbox you want to run in Sepia
MEDI_HOME         = [MEDI_dir         MEDI_version         filesep];
FANSI_HOME        = [FANSI_dir        FANSI_version        filesep];
STISuite_HOME     = [STISuite_dir     STISuite_version     filesep];
SEGUE_HOME        = [SEGUE_dir        SEGUE_version        filesep];
MRISC_HOME        = [MRISC_dir        MRISC_version        filesep];
MRITOOLS_HOME     = [MRITOOLS_dir     MRITOOLS_version     filesep];
```

IMPORTANT: Please do not modify the original structure of these toolboxes, SEPIA searches the path of the related functions based on the original folder structure.

Now you can start the GUI by entering `sepia` in the MATLAB's command window.

1.1.4 Setup deep learning tools for SEPIA

Starting from v1.1.2, several deep learning methods for QSM processing are (experimentally) supported in SEPIA. Please refer to the individual algorithm pages on *QSM Algorithms* and *Background Magnetic Field Removal Algorithms* for more infomation on setting up these methods in SEPIA.

1.1.5 Compatibility

SEPIA is developed mainly in MATLAB R2016b on Linux and macOS. In general, all methods should compatible with earlier MATLAB versions up to R2014b. Most of the methods should also compatible with MATLAB R2017a or later, and other OS, except you might encounter issue with the following functions/algorithms

- **Laplacian Boundary Value (LBV) for background field removal**

  Note: If the LBV algorithm doesn't work on your operating system, you can go to the '_LBV' directory of the MEDI toolbox and try the following command in the Matlab command window to re-compile the library: `mex mexMGv6.cpp`

- Graphcut for phase unwrapping
- SEPIA v1.0 supports both FANSI v1.0 and v3.0. However, compartibility to FANSI v2.0 (commit 77023b65, released on 2020.07.27) is not yet tested!

1.2 Data Preparation

SEPIA supports the following type of data input:

- uncompressed or compressed NIfTI images (.nii and .nii.gz)

Warning: DICOM images input has been deprecated since v0.7.0.

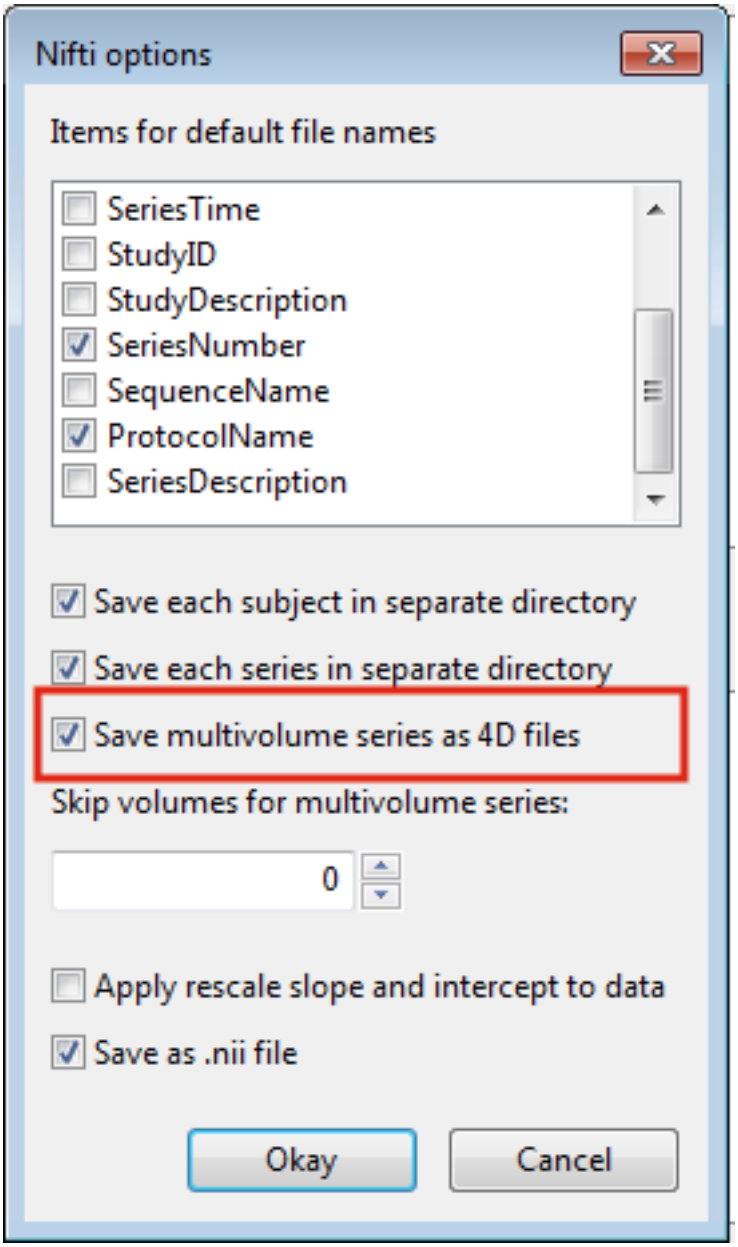
1.2.1 NIfTI images

NIfTI data is a handy way to work with **SEPIA**. **SEPIA** is mainly designed for 3D data. Therefore, the main standalone of **SEPIA** can only work with 3D/4D (row, column, slice, time) NIfTI data (3D for some standalone applications).

Data conversion

SEPIA is tested with (but not limited to) the following three data conversion tools:

1. dcm2niix: Please make sure the 'merge' option (-m) is set to 'no' for the conversion (i.e. `dcm2niix -m n`). In this way, multiple 3D volumes (the number of volumes depends on the number of echoes acquired) will be created together with the JSON files containing the TE of each echo (if you enable the merge option of dcm2niix you will only have one JSON file containing one TE). You can then merge the echo images into 4D using tools like fslmerge.
2. dicm2nii
3. MRIConvert: Please have the following setting checked: 'Option' -> 'Save multivolumes series as 4D files':



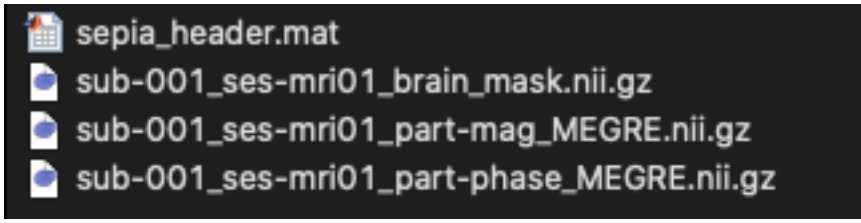
In this way, your mGRE data will be stored as 4D FSL NIfTI data that is a valid input of **SEPIA**.

Data input

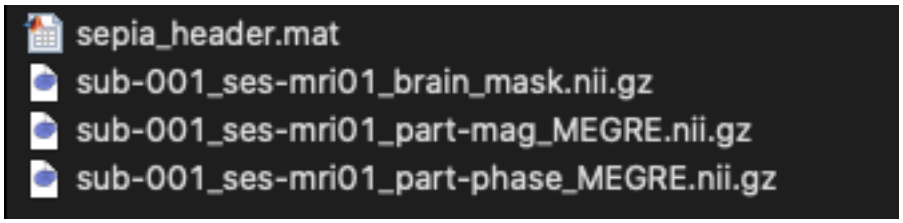
Once you have the NIfTI files ready, **SEPIA** provides two options to load the data:

- select the required files directly, or
- prepare the data with specific names and put all of them in a common directory, from which you can specify the input directory in **SEPIA**. The name requirement depends on the standalone you are working on. For more specific details please check the wiki pages of each standalone applications:

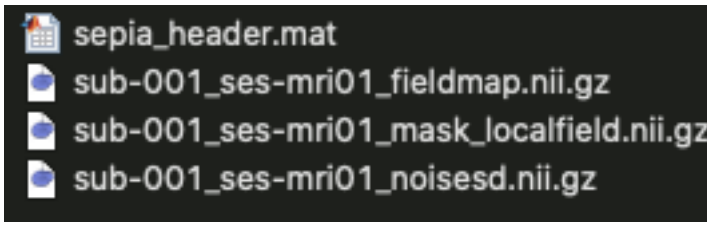
  - *SEPIA (One-stop QSM processing)*, **e.g.,**



  - *Phase Unwarpping Standalone*, **e.g.,**



  - *Background Magnetic Field Removal Standalone*, **e.g.,**

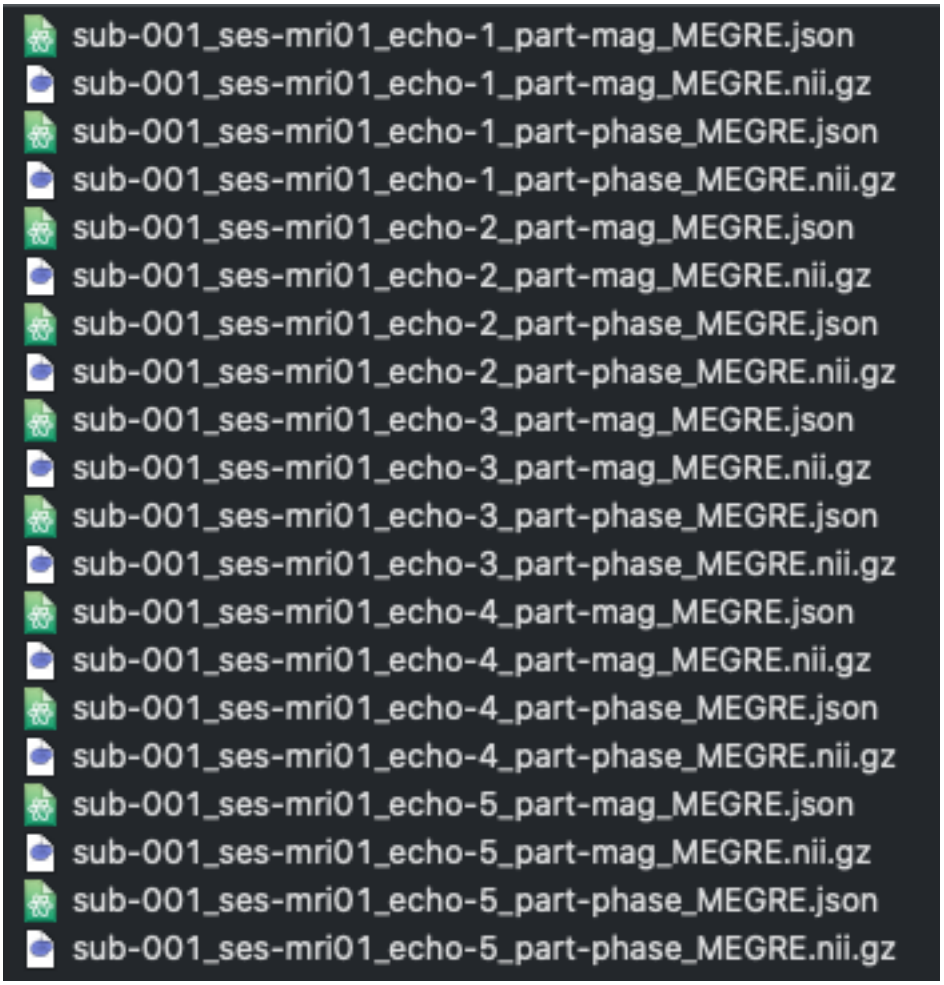


  - *QSM Standalone*, **e.g.,**

- **Brain Imaging Data Structure (BIDS) specification** Starting from v1.0, it is possible to specify a directory that follows the BIDS specification as an input directory for **SEPIA** one-stop standalone application. Specifically, the following rules must be fulfilled:

  1. Filename of the phase data must contain the following 'key|label' pair: 'part-phase';
  2. Filename of the magnitude data must contain the following 'key|label' pair: 'part-mag';
  3. Filename of the JSON file must contain the following 'key|label' pair: 'part-mag';
  4. No NIfTI and JSON files other than the data for QSM processing that have filenames containing the 'key|label' pairs 'part-mag' and 'part-phase'.
  5. For multi-echo data, the 'key' labelled 'echo-' must present in the filename.

  Here is an example of a valid BIDS directory content for **SEPIA**



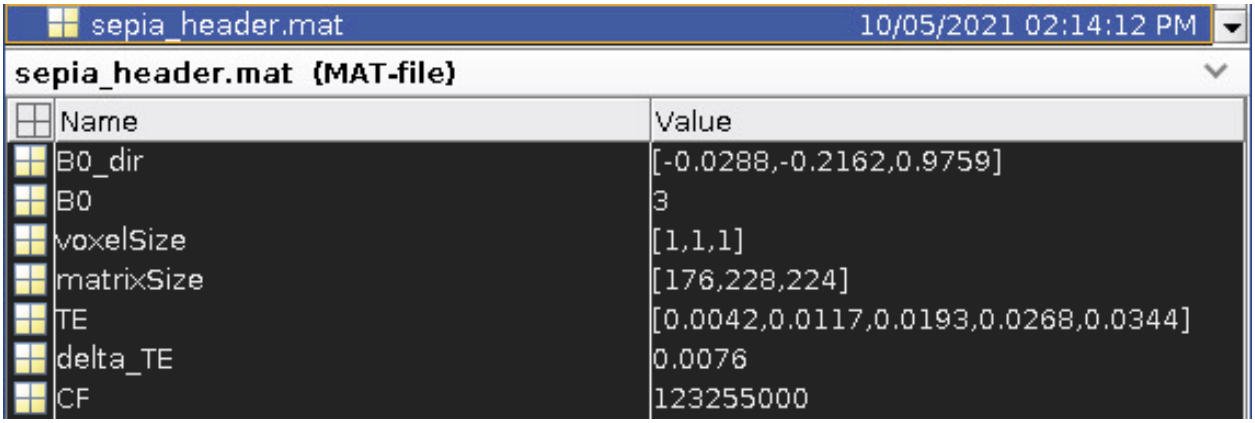
1.3 SEPIA header

To obtain a frequency shift map or magnetic susceptibility map with correct units, it is important to inform the corresponding algorithm(s) with information including the main magnetic field strength (B0) of the scanner, the main magnetic field direction with respect to the field of view (FOV) and the echo times (TEs) used in the acquisition. Unfortunately, such information may not be (directly) available in the NIfTI header after the DICOM conversion. Therefore, an additional file is needed to provide this information in **SEPIA** which is called SEPIA header.

SEPIA header is a MAT-file (.mat) that stores some important header information of the input data. The following example shows the variables (case-sensitive) that must be stored in the header file:

```
B0 = 3;                 % magnetic field strength, in Tesla
B0_dir = [0;0;1];       % main magnetic field direction, [x,y,z]
CF = 3*42.58*1e6;       % imaging frequency, in Hz (B0*gyromagnetic_ratio)
TE = [te1,te2,te3,te4]; % echo time for each GRE image, in second
delta_TE = te2-te1;     % echo spacing, in second
matrixSize = [64,64,64] % image matrix size
voxelSize = [1,1,1]     % spatial resolution of the data, in mm
```

Example of 5-echo 3D data:



You can create this file manually in Matlab. Alternatively, you can use the Utility standalone application to generate the file. Please visit the corresponding wiki page for more information about how to generate the header file.

Note: Starting from SEPIA v1.0, only variables 'B0' and 'TE' are essential in the SEPIA header file. The rest of the parameters can be read from the NIfTI file directly if dcm2niix (and other supported DICOM-to-NIfTI conversion tools) is used.

1.4 SEPIA Output Files

1.4.1 Output files of Total field reconvery and phase unwrapping

Data	Description
<Prefix>_fieldmap.nii.gz	Unwrapped total frequency shift map in Hz.
<Prefix>_weights.nii.gz	SNR-weighted image derived from standard deviation of noise in phase data.
<Prefix>_noisesd.nii.gz	Estimated standard deviation of noise in the phase data.
<Prefix>_part-phase_rad.nii.gz	Wrapped phase data in radian (only if the input data contains voxel exceeds the range of [-pi,pi]).
<Prefix>_part-phase_unwrapped.nii.gz	Unwrapped phase data in radian (if selected).
<Prefix>_part-phase_reverse.nii.gz	inverted phase data, = -(phase)
<Prefix>_part-phase_bipolarcorr.nii.gz	phase data corrected for bipolar readout gradient
<Prefix>_bipolar_phase.nii.gz	Estimated phase offset induced by bipolar gradient readout.
<Prefix>_mask_brain.nii.gz	Brain mask derived from brain extraction of FSL (if selected).
<Prefix>_mask_localfield.nii.gz	Signal mask for background field removal step (if voxel exclusion is selected with 'Brain mask' option).
<Prefix>_relativeresidual.nii.gz	Relative residual derived using mono-exponential model with a single frequency shift (if voxel exclusion is selected).
<Prefix>_relativeresidualweights.nii.gz	Weighting maps [0,1] derived from thresholding relative-residual map using user-defined value.
<Prefix>_mask_reliable.nii.gz	Derived from thresholding relative-residual map using user-defined value.

1.4.2 Output files of Background field removal

Data	Description
<Prefix>_localfield.nii.gz	Local (tissue) field map in Hz.
<Prefix>_mask_QSM.nii.gz	Signal mask for QSM step.

1.4.3 Output files of QSM dipole inversion

Data	Description
<Prefix>_Chimap.nii.gz	Magnetic susceptibility map in ppm.
<Prefix>_mask_referenceregion.nii.gz	Reference region used in QSM normalisation (if selected).

1.4.4 Other output files

Data	Description
sepia_config.m	Automatic generated script by the GUI of SEPIA containing all user specified parameters.
run_sepia.log	Event log file of the Matlab's command window output.
run_sepia.error	Error message of SEPIA (if any).

1.4.5 Output files of SWI/SMWI

Data	Description
<Prefix>_swi-phase.nii.gz	High-pass filtered phase data.
<Prefix>_swi-positive.nii.gz	Positive susceptibility-weighted images (if selected).
<Prefix>_swi-mIP-positive.nii.gz	Positive susceptibility-weighted images with minimum intensity projection over user-defined field of view (if selected).
<Prefix>_swi-negative.nii.gz	Negative susceptibility-weighted images (if selected).
<Prefix>_swi-mIP-negative.nii.gz	Negative susceptibility-weighted images with minimum intensity projection over user-defined field of view (if selected).
<Prefix>_smwi-paramagnetic.nii.gz	Paramagnetic susceptibility map-weighted images (if selected).
<Prefix>_smwi-mIP-paramagnetic.nii.gz	Paramagnetic susceptibility map-weighted images with minimum intensity projection over user-defined field of view (if selected).
<Prefix>_smwi-diamagnetic.nii.gz	iamagnetic susceptibility map-weighted images (if selected).
<Prefix>_smwi-mIP-diamagnetic.nii.gz	Diamagnetic susceptibility map-weighted images with minimum intensity projection over user-defined field of view (if selected).

1.5 Release note

1.5.1 1.2.0 (commit d2f54a3)

Release date: 11 October 2022

Toolbox related

- Support several deep learning based methods (BFRnet, xQSM, QSMnet+ and LP-CNN) on Linux
- Support atlas-based subcortical structure segmentation (CIT168 Reinforcement learning atlas, MuSus-100 and AHEAD) on Linux and Mac
- Integrate R2* mapping toolbox into SEPIA
- New function to further refine brain mask by thresholding high R2* voxels on brain edges
- When magnitude image is used for NDI, the image will be weighted by the intensity of the 99th percentile of the masked voxels instead of the maximum to improve robustness

1.5.2 1.1.1 (commit a7680bb) Patch update

Release date: 3 October 2022

Toolbox related

- New function to support refining brain mask before background field removal step
- Official support mritools v3.5.5 (https://github.com/korbinian90/CompileMRI.jl/releases/tag/v3.5.5), which included ROMEO and CLEAR-SWI
- 'ROMEO_HOME' is now renamed to 'MRITOOLS_HOME' in SpecifyToolboxesDirectory.m
- Add GPU compatibility for NDI

Bug fix

- Fixed bug for full pipeline application and phase unwrapping standalone application used different bipolar readout correction implementations
- Fixed bug for NDI when weight was used instead of weight^2

1.5.3 1.1.0 (commit 9ffe0e2)

Release date: 22 September 2022

GUI related

- Experimental support to export phase image from real|imaginary images (could be useful for GE data)

Toolbox related

- Better compatibility with ROMEO, including loading config file with ROMEO parameter back to GUI
- New implementation of bipolar readout phase offset correction (no phase unwrapping is required) and provide the estimated phase offset map as output
- New implementation of incorporating relative residual map into the weighting map when the "Exclude voxels using reisdual" box is checked with "weighting map" option

Bug fix

- Several minor bugs fixed

Backend related (advanced user)

- New backend architecture for SWI/SMWI. Now new SWI or SMWI methods can be added to SEPIA usinf the add-on feature similar to other methods

1.5.4 1.0.1 (commit 3a2b387)

Release date: 4 Aug 2022

Toolbox related

- Updated function performing phase conversion from arbitary DICOM values to radian (could result in minor numerical differences compared to previous versions if the input phase NIfTI not in radian)

Bug fix

- Fixed bug when phase NIfTI is in wrapped range with non-unity rescale slope (e.g. from Philips' scanners)
- Several other minor bugs fixed

1.5.5 1.0 (commit 8e35aee)

Release date: 25 Feb 2022

GUI related

- New utility tool for managing external dependencies

Toolbox related

- Support ROMEO as total field computation and phase unwrapping method
- Support MRI susceptibility calculation methods for QSM dipole field inversion
- Support FANSI v3.0 (note that the algorithm parameters are adapted for this version)
- Improve BIDS compartibility with SEPIA
- Update output filenames in accordance with BIDS format
- Improve the comparability of weighting maps across different datasets and methods
- GUI supports on managing toolbox dependencies

Bug fix

- Improve the robustness of measuring a reference phase point for B0 computation

Backend related (advanced user)

- New architecture for easier integration of new echo combination methods
- New data loading method to reduce memory usage during processing

1.5.6 0.8.1.1 (commit 52dd20b)

Release date: 6 May 2021

Bug fix

- Fixed bug when using single-echo dataset
- Fixed bug when input phase data in unit of radian with single datatype

1.5.7 0.8.1 (commit c78247d)

Release date: 4 Feb 2021

Toolbox related

- Log file and error message file are now paired (last 15 digits in the extension) instead of sorting in simple numerical order
- Log file and error message file are now generated in both GUI and command-based operations (when using `sepiaIO`)
- When running SEPIA, the current directory will temporaily move to the output directory to avoid overwriting temporary files if multiple processings happen simultaneously
- A SEPIA pipeline configuration file will be automatically generated using `sepiaIO` is the output directory does not have any existing configuration file. This would be useful to look up the pipeline used to produce the results when using command-based operation

Bug fix

- Bug fix when running FANSI (see here)
- Bug fix when getting B0 direction from Sagittal or Coronal acquisition (see here)
- Bug fix when running QSM standalone with magnitude image for regularisation (see here)
- Bug fix when running MEDI with zeropadding option is not equal to zero

Backend related (advanced user)

- Improved readiility of how the data are loaded in SEPIA, which could make better BIDS compartibility in the future

1.5.8 0.8.0 (commmit b4255d8)

Release date: 18 July 2020

GUI related

- New layout for input/output panel for data selection
- New pipeline configuration file (sepia_config.m), log file (run_sepia.log) and error message file (run_sepia.error)
- New feature to load parameters in a pipeline configuration file (sepia_config.m) to the GUI
- New option to save unwrapped echo phase
- New option to exlcude unreliable voxels
- New option to select reference tissue for QSM normalisation/referencing
- New option to remove residual B1 field in local field using spherical harmonic function with adjustable order of the fitting

Toolbox related

- Support the lastest version of MEDI toolbox (Jan 15, 2020)
- Support extra brain extraction (FSL's BET) parameters from MEDI toolbox
- New 'percentage' option for MEDI+0 algorithm
- Support the lastest version of FANSI toolbox (commit dc68c306)
- New option to use weak harmonic regularisation with FANSI

Backend related (advanced user)

- Support developers adding a third-party method as an addon
- Introduce tutorial scripts to guide developers on how to adding third-party method in SEPIA
- Introduce functions to simplify the workflow of creating new method panel
- The order of removal of residual B1 field and mask erosion is interchanged to produce better a fitting result

Bug fix

- Bug fix: running SEPIA without parrallel computing toolbox
- Bug fix: running MEDI toolbox nonlinear fit echo phase combination with 2 echoes
- Bug fix: running MEDI method in SEPIA
- Bug fix: running single echo data with exclusion of unreliable voxels option enabled

Please update the MEDI toolbox (Jan 15, 2020) and FANSI toolbox (commit dc68c306) to the lastest version for the best performance.

1.5.9 0.7.3 (commmit 68c53bc)

Release date: 9 Nov 2019

- Support nonlinear dipole inversion (NDI) as external library
- Support SEGUE as external library

1.5.10 0.7.2 (commmit bf020ce)

Release date: 4 Jun 2019

- Support single-echo dataset
- Bug fix with odd-number matrix dimension by zero-padding
- Offload unuse variables to reduce memory usage
- Bug fix for reading NIfTI when the rescale slope and intercept are not 1 and 0

1.5.11 0.7.1 (commmit dc51fbe)

Release date: 9 May 2019

- Support simple susceptibility weighted imaging (SWI) and susceptibility map weighted imaging (SMWI) as part of the GUI
- resolved loading/saving NIfTI issue related to 0.7.0 update
- DICOM input is deprecated: the only possible input is NIfTI data
- fixed bug when running MEDI with CSF regularisation
- fixed bug for single echo SWI
- now support automatic magnitude and phase images detection with name containing string "mag" for magnitude image and "ph" for phase image
- fixed global phase offset with graph-cut phase unwrapping

1.5.12 0.7.0 (commmit e66d8e4)

Release date: 12 Apr 2019

- redesigned log file format; the algorithms and parameters being used are much clearer and neat than before (previous log file cannot work in this version)
- resolved '.nii.nii' issue when using STI suite algorithms
- resolved no. of iterations with FANSI does not change issue
- resolved problematic QSM results with FANSI when an input matrix is an odd number
- resolved excluded unreliable voxels issue when 3D best path algorithm doesn't work
- improved build-in VSHARP results when there are masked voxels on the image edges
- added image erosion function for background field removal algorithms
- get header function is now compatible with the JSON files generated by dcm2niix and dicm2nii

1.5.13 0.6.0 (commmit 1c27dc4)

Release date: 1 Sep 2018

- updated diretcory structure
- added options to select individual files

1.6 SEPIA (One-stop QSM processing)

1.6.1 What is SEPIA?

SEPIA is a quantitative susceptibility mapping (QSM) pipeline analysis tool for (but not limited to) neuroimaging application. It provides all the essential functions to compute a susceptibility map from a 3D multi-echo GRE phase data, including phase unwrapping, background field contribution removal and dipole inversion. Incorporating with different toolboxes in SEPIA gives users the advantages of having a variety of options to build a pipeline that works the best for their data. When you use the SEPIA graphical user interface to process the data, a configuration (config) file will be generated that contains all the settings and commands that you've specified in the pipeline. This config file will be particularly useful for batch processing.

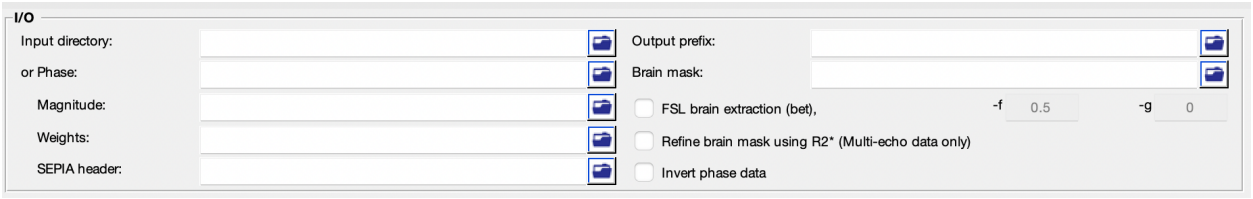
1.6.2 Structure of the application

This standalone consists of 4 panels:

- Input/Output(I/O) panel,
- Total field recovery and phase unwrapping panel,
- Background field removal panel, and
- QSM panel.

Description of each panel is given below:

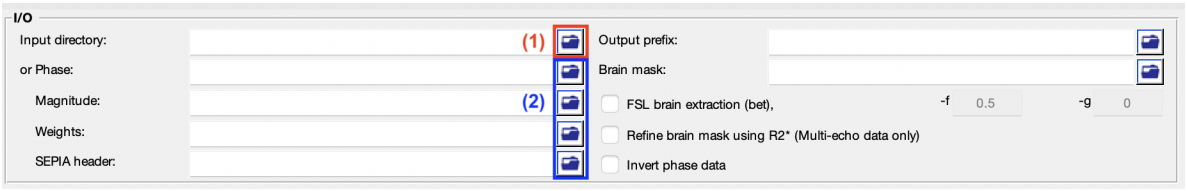
I/O panel



The I/O panel is responsible for data input/output and data processing that is not specific to QSM.

- Data input

  There are two pathways to specify input in this application:



  1. Specify a directory that contains all essential data.

     The essential data are:

Data	Description
Phase	3D/4D phase data ([x,y,slice,time]), must contain 'ph' in the filename, e.g. *phase.nii.gz* or *ph.nii.gz*,
Magnitude	3D/4D magnitude data ([x,y,slice,time]), must contain 'mag' in the filename, e.g. *magn.nii.gz* or *mag.nii.gz*;
Header	see *SEPIA header* for more information, must contain 'header' in the filename, e.g. *header.mat*
Mask	(optional) 3D signal mask, if provided, must contain string 'mask' in the filename, e.g. *mask.nii.gz*

Warning: Please make sure the filenames follow the above rules, or the BIDS specification (see **Data Input** in *Data Preparation*), and no other files in the directory sharing the same string labels (i.e. 'ph', 'mag', 'header' and 'mask').

2. Specify the required data separately using the GUI buttons.

Note: The 'Weights' input is an optional input. You can specify a 3D data which will be used as prior information in regularised optimisation in QSM dipole inversion. If the 'Weights' input is empty, the weighting map will be automatically computed in subsequent QSM processing.

- Output prefix

  By default, the output files generated by SEPIA will be stored in a directory named '*output*' under the directory of the input files (i.e. '_/your/input/directory/output/_'). The prefix of the output filename is '*Sepia*'. You can change the default output directory and prefix according to your preference. If the output directory does not exist, the application will create the directory.

  Note: Make sure the 'Output prefix' field contains a full path of the output directory and a filename prefix.

- Brain mask

  You can optionally specify a signal (brain) mask NIfTI file. If this input is empty and no mask is found in the input directory, SEPIA will automatically run the FSL's brain extraction tool (bet) provided with the MEDI toolbox to compute the brain mask.

- FSL brain extraction (bet)

  Brain mask can be computed using the Matlab implementation of FSL's BET provided with MEDI toolbox, with options including fractional intensity threshold (-f) and vertical in fractional intensity threshold (-g). More information regarding the options can be found in BET/UserGuide.

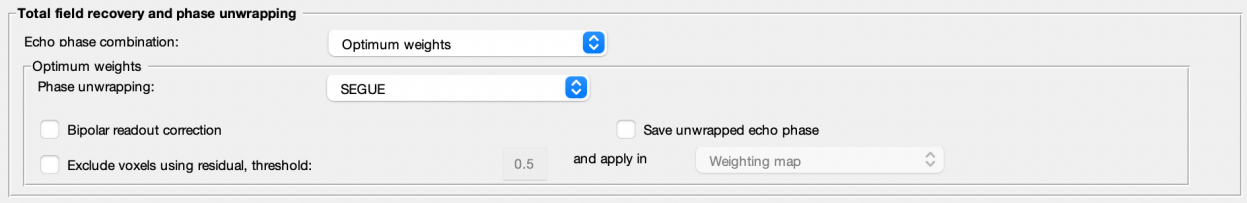
- Refine brain mask using R2* (Multi-echo data only)

  If enable, a R2* map will be computed and used to threshold out high R2* voxels on the edges of the brain mask.

- Invert phase data

  Checking this option will invert the contrast of the SEPIA output frequency and QSM maps. Mathematically it inverse the signal phase by computing the signal conjugate. It is useful if you want to have specific colour scheme for QSM (e.g. dark colour for paramagnetic susceptibility).

Total field recovery and phase unwrapping panel



- Echo phase combination

  Select a method for temporal phase unwrapping with multi-echo data.

  Note: If the number of echoes is less than 3 and 'MEDI nonlinear fit' is chosen, 'Optimum weights' method will be automatically used.

  Warning: The 'MEDI nonlinear fit (Bipolar, testing)' method is not fully supported yet.

- Phase unwrapping

  Select a method for spatial phase unwrapping.

  Warning: The '3D best path' method might not work in most operating systems.

- Bipolar readout correction

  Correct the phase inconsistency between odd and even echoes, and a gradient-like (should be only in the readout direction) magnetic field contributed from eddy current due to bipolar readout.

- Save unwrapped echo phase

  Export all unwrapped echo phase images as NIfTI.

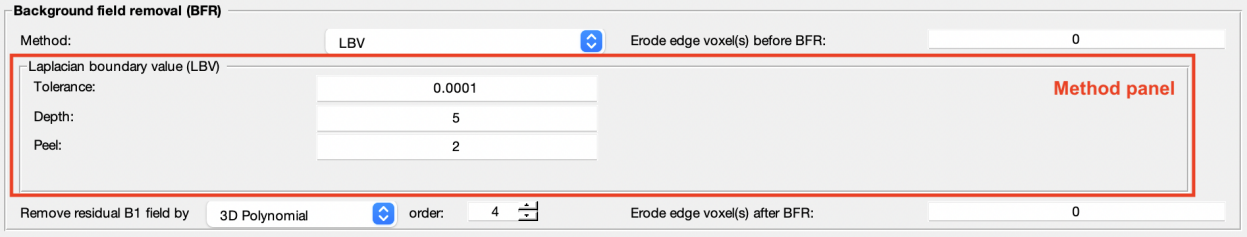
- Exclude voxels using residual, threshold:

  Exclude voxels that have high relative residual based on a single compartment model fitting. The output data with suffix '*relative-residual.nii.gz* will be used for thresholding. For voxels that have intensity **higher** than the threshold will be **excluded** from subsequent processing. Two methods are supported to exclude those voxels:

  1. 'Weighting map': Please see :ref: *weightings-in-sepia'* Section **Further modulation on the weighting maps**
  2. 'Brain mask': the excluded voxels will be excluded in the signal mask in the subsequent processing. This will affect both background field removal and QSM dipole inversion results.

  Only available for quantitative methods (i.e. '3D best path', 'Region growing (MEDI)', 'SEGUE' and 'ROMEO') and 'Graphcut' method.

Background field removal panel



- Method

  Select a background field removal method. The method parameters will be displayed on the method panel.

- Remove residual B1 field by

  Option to remove potential field contributions originated from B1 by polynomial fitting or spherical harmonic fit.

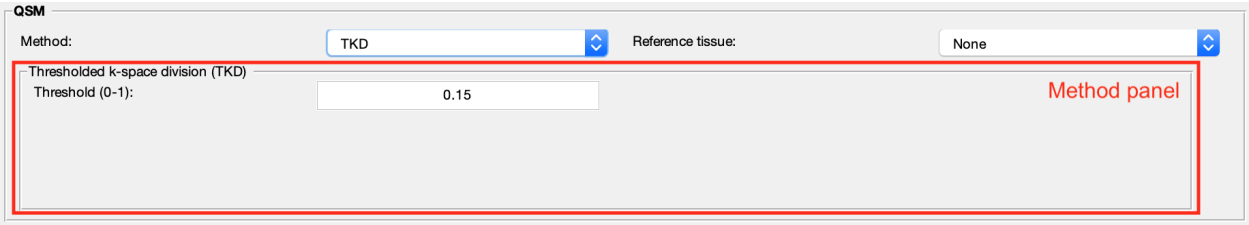
- Erode edge voxel(s) before BFR

  Remove n voxel(s) away from the edge of the brain mask **BEFORE** the background field removal step.

- Erode edge voxel(s) after BFR

  Remove n voxel(s) away from the edge of the brain mask **AFTER** the background field removal step. This operation is performed **prior** the 'Remove potenital B1 residual phase' operation (if selected).

QSM panel



- Method:

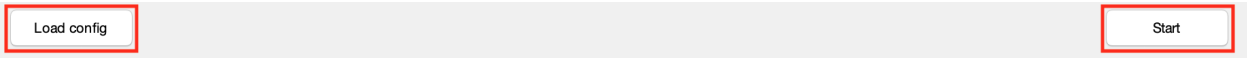
  Select a QSM dipole inversion method. The method parameters will be displayed on the method panel.

- Reference tissue

  Select a tissue for QSM value referencing.

  Warning: The 'CSF' tissue option works only when **multi-echo** magnitude data is provided.

Others



- Load config

  Import the method related settings specified in the SEPIA-generated config file to the SEPIA GUI. **NO** modification will be made in the I/O panel.

- Start

  Generate a SEPIA config file that contains all user-defined methods and parameters for QSM processing based on the setting in the GUI. SEPIA will run the config file immediately once it is generated.

1.7 Phase Unwarpping Standalone

1.7.1 Phase unwrapping in QSM

Phase wrapping occurs when continuous phase information is sampled in a discrete wrapped phase. The measured phase accumulation larger than one phase cycle is wrapped into the interval [-π, π), causing the discontinuity in the phase data. To recover the true phase values, one must solve this ambiguity problem by adding the correct integer number of phase cycles to the phase data in order to recover the true phase revolution.

The objective of this standalone application is to recover the actual, total phase shift of the acquired data.

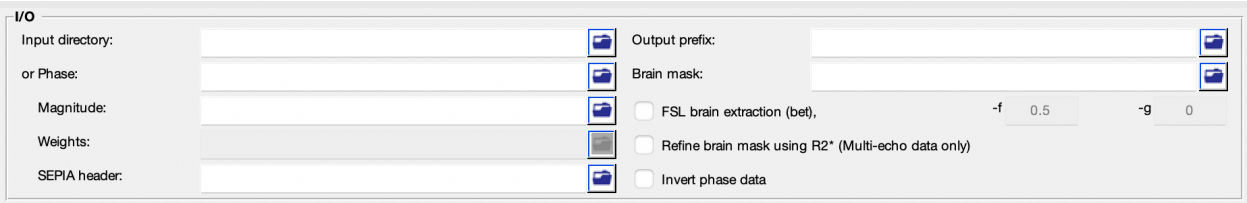
1.7.2 Structure of the application

This standalone consists of two panels:

- I/O panel, and
- Total field recovery and phase unwrapping panel.

Description of each panel is given below:

I/O panel



The I/O panel is responsible for data input/output and data processing that is not specific to QSM.

- Data input

  There are two pathways to specify input in this application:

  1. Specify a directory that contains all essential data.

     The essential data are:

Data	Description
Phase	3D/4D phase data ([x,y,slice,time]), must contain 'ph' in the filename, e.g. *phase.nii.gz* or *ph.nii.gz*,
Magnitude	3D/4D magnitude data ([x,y,slice,time]), must contain 'mag' in the filename, e.g. *magn.nii.gz* or *mag.nii.gz*;
Header	see *SEPIA header* for more information, must contain 'header' in the filename, e.g. *header.mat*
Mask	(optional) 3D signal mask, if provided, must contain string 'mask' in the filename, e.g. *mask.nii.gz*

> **Warning:** Please make sure the filenames follow the above rules, or the BIDS specification (see **Data Input** in *Data Preparation*), and no other files in the directory sharing the same string labels (i.e. 'ph', 'mag', 'header' and 'mask').

2. Specify the required data separately using the GUI buttons.

- Output prefix

  By default, the output files generated by SEPIA will be stored in a directory named '*output*' under the directory of the input files (i.e. '_/your/input/directory/output/_'). The prefix of the output filename is '*Sepia*'. You can change the default output directory and prefix according to your preference. If the output directory does not exist, the application will create the directory.

Note: Make sure the 'Output prefix' field contains a full path of the output directory and a filename prefix.

- Brain mask

  You can optionally specify a signal (brain) mask NIfTI file. If this input is empty and no mask is found in the input directory, SEPIA will automatically run the FSL's brain extraction tool (bet) provided with the MEDI toolbox to compute the brain mask.

- FSL brain extraction (bet)

  Brain mask can be computed using the Matlab implementation of FSL's BET provided with MEDI toolbox, with options including fractional intensity threshold (-f) and vertical in fractional intensity threshold (-g). More information regarding the options can be found in BET/UserGuide.

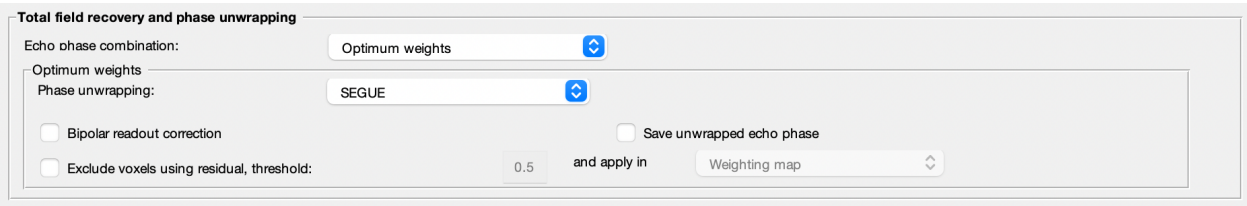
- Refine brain mask using R2* (Multi-echo data only)

  If enable, a R2* map will be computed and used to threshold out high R2* voxels on the edges of the brain mask.

- Invert phase data

  Checking this option will invert the contrast of the SEPIA output frequency and QSM maps. Mathematically it inverse the signal phase by computing the signal conjugate. It is useful if you want to have specific colour scheme for QSM (e.g. dark colour for paramagnetic susceptibility).

Total field recovery and phase unwrapping panel



- Echo phase combination

  Select a method for temporal phase unwrapping with multi-echo data.

Note: If the number of echoes is less than 3. 'Optimum weights' method will be automatically used.

> **Warning:** The 'MEDI nonlinear fit (Bipolar, testing)' method is not fully supported yet.

- Phase unwrapping

  Select a method for spatial phase unwrapping.

> **Warning:** The '3D best path' method might not work in most operating systems.

- Bipolar readout correction

  Correct the phase inconsistency between odd and even echoes, and a gradient-like (should be only in the readout direction) magnetic field contributed from eddy current due to bipolar readout.

- Save unwrapped echo phase

  Export all unwrapped echo phase images as NIfTI.

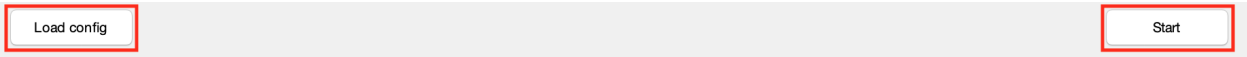
- Exclude voxels using residual, threshold:

  Exclude voxels that have high relative residual based on a single compartment model fitting. The output data with suffix '*relative-residual.nii.gz* will be used for thresholding. For voxels that have intensity **higher** than the threshold will be **excluded** from subsequent processing. Two methods are supported to exclude those voxels:

  1. 'Weighting map': Please see :ref: *weightings-in-sepia'* Section **Further modulation on the weighting maps**
  2. 'Brain mask': the excluded voxels will be excluded in the signal mask in the subsequent processing. This will affect both background field removal and QSM dipole inversion results.

  Only available for quantitative methods (i.e. '3D best path', 'Region growing (MEDI)', 'SEGUE' and 'ROMEO') and 'Graphcut' method.

Others



- Load config

  Import the method related settings specified in the SEPIA-generated config file to the SEPIA GUI. **NO** modification will be made in the I/O panel.

- Start

  Generate a SEPIA config file that contains all user-defined methods and parameters for QSM processing based on the setting in the GUI. SEPIA will run the config file immediately once it is generated.

1.8 Background Magnetic Field Removal Standalone

1.8.1 Background field removal in QSM

The phase we measured in a GRE acquisition is affected by not only the brain tissue but also sources like B0 inhomogeneity and air sinus. In order to compute the susceptibility sources only contributed by the brain tissue, it is important to remove all the non-local field effect. Fortunately, the characteristic of the local field is different from that of the

non-local field, it is possible to separate the two fields from the unwrapped total field. This standalone application is designed to solve the field separation problem.

Caution: It is crucial that the background field contribution is removed accurately in this stage. Otherwise, the remaining field due to background sources will be treated as part of the local field, degrading the quality of QSM result.

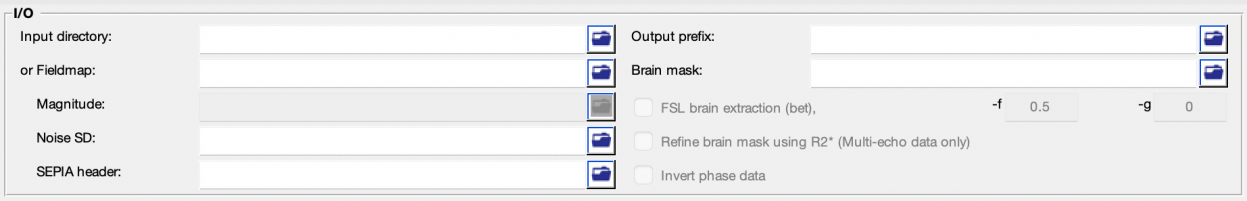
1.8.2 Structure of the application

This application consists of two panels:

- I/O panel, and
- Background field removal panel.

Description of each panel is given below:

I/O panel



The I/O panel is responsible for data input/output and data processing that is not specific to QSM.

- Data input

  There are two pathways to specify input in this application:

  1. Specify a directory that contains all essential data.

     The essential data are:

Data	Description
Total field	3D fieldmap (a.k.a. total field map) in Hz ([x,y,slice]), must contain 'fieldmap' in the filename, e.g. *fieldmap.nii.gz*
Header	see *SEPIA header* for more information, must contain 'header' in the filename, e.g. *header.mat*
Mask	3D signal mask, if provided, must contain string 'mask' in the filename, e.g. *mask.nii.gz*

     Warning: Please make sure the filenames follow the above rules (see also **Data Input** in *Data Preparation*) and no other files in the directory sharing the same string labels (i.e. 'fieldmap', 'header' and 'mask').

     Warning: The rule of filename for the fieldmap is changed from 'total-field' to 'fieldmap' in v1.0 in accordance to BIDS specification.

  2. Specify the required data separately using the GUI buttons.

> **Note:** The 'Noise SD' input is an optional input for 'PDF' algorithm. You can specify a 3D data that provides the noise standard deviation of the original phase data (e.g. *noisesd.nii.gz* derived from phase unwarpping step in SEPIA).

- Output prefix

  By default, the output files generated by SEPIA will be stored in a directory named '*output*' under the directory of the input files (i.e. '_/your/input/directory/output/_'). The prefix of the output filename is '*Sepia*'. You can change the default output directory and prefix according to your preference. If the output directory does not exist, the application will create the directory.

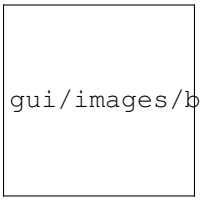
  > **Note:** Make sure the 'Output prefix' field contains a full path of the output directory and a filename prefix.

- Brain mask

  You can specify a signal (brain) mask NIfTI file.

  > **Note:** A mask **must** be provided (either in the input directory or specified) in this standalone.

Background field removal panel



- Method

  Select a background field removal method. The method parameters will be displayed on the method panel.

- Remove residual B1 field by

  Option to remove potential field contributions originated from B1 by polynomial fitting or spherical harmonic fit.

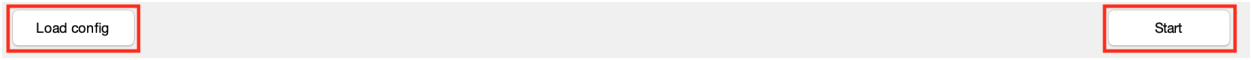
- Erode edge voxel(s) before BFR

  Remove the edge voxels from the brain mask before background field removal step. Useful when the input brain mask is not tightly fitted.

- Erode edge voxel(s) after BFR

  Further remove the edge voxels from the brain mask. Useful when the local field is not reliably estimated on the brain edges. This operation is performed **prior** the 'Remove potenital B1 residual phase' operation (if selected).

Others



- Load config

  Import the method related settings specified in the SEPIA-generated config file to the SEPIA GUI. **NO** modification will be made in the I/O panel.

- Start

  Generate a SEPIA config file that contains all user-defined methods and parameters for QSM processing based on the setting in the GUI. SEPIA will run the config file immediately once it is generated.

1.9 QSM Standalone

1.9.1 How can we map the magnetic susceptibility sources from local field map?

A local field (or tissue field) is the field generated by the magnetic susceptibility sources from the brain tissue. The assumption of the field generated by a magnetic susceptibility point source is a dipole field. We can then try to compute the susceptibility values by deconvolving the local field with a dipole field. However, due to the fact that the dipole field in the k-space (i.e. the dipole kernel) has zero values on the cone surface at 54.7° and values close to this surface will also be close to zeros. As a result, the number of unknown parameters is more than the measurements we have in the data, leading to the problem ill-conditioned and the resulting QSM map has the so-called streaking artefact. To solve the ill-conditioned problem, most of the QSM algorithms try to add a regularisation term in the QMS dipole inversion problem, imposing spatial smoothness in the QSM map and/or the QSM map has similar anatomical features as shown in the magnitude images in order to reduce the streaking artefact in the QSM result.

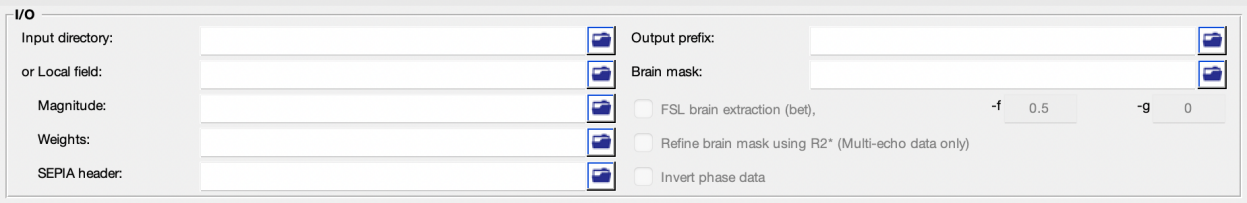
1.9.2 Structure of the application

This standalone consists of two panels:

- I/O panel, and
- QSM panel.

Description of each panel is given below:

I/O panel



The I/O panel is responsible for data input/output and data processing that is not specific to QSM.

- Data input

  There are two pathways to specify input in this application:

  1. Specify a directory that contains all essential data.

     The essential data are:

Data	Description
Local field	3D local (tissue) field map ([x,y,slice]) in Hz, must contain 'localfield' in the filename, e.g. *localfield.nii.gz*
Magnitude	(optional) 4D magnitude data ([x,y,slice,time]), must contain 'mag' in the filename, e.g. *mag.nii.gz*
Weights	(optional) 3D weighting map ([x,y,slice,time]), must contain 'weights' in the filename, e.g. *weights.nii.gz*
Header	see *SEPIA header* for more information, must contain 'header' in the filename, e.g. *header.mat*
Mask	3D signal mask, if provided, must contain string 'mask' in the filename, e.g. *mask.nii.gz*

Warning: Please make sure the filenames follow the above rules (see also **Data Input** in *Data Preparation*) and no other files in the directory sharing the same string labels (i.e. 'localfield', 'mag', 'weights', 'header' and 'mask').

2. Specify the required data separately using the GUI buttons.

Note: Depending on the QSM dipole inversion algorithm selected. The 'Magntude and 'Weights' input are optional input.

- Output prefix

  By default, the output files generated by SEPIA will be stored in a directory named '*output*' under the directory of the input files (i.e. '_/your/input/directory/output/_'). The prefix of the output filename is '*Sepia*'. You can change the default output directory and prefix according to your preference. If the output directory does not exist, the application will create the directory.

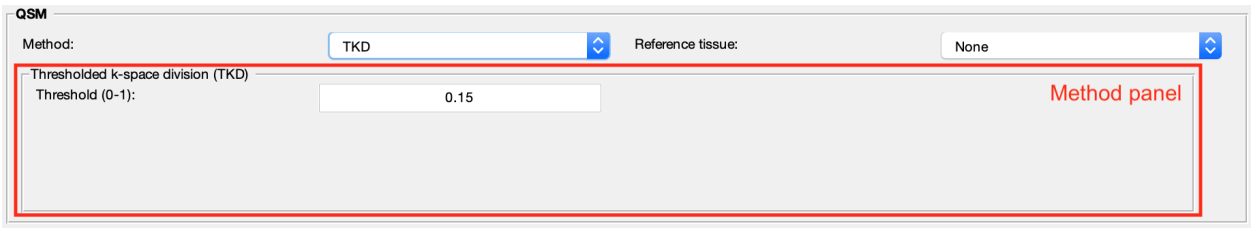
  Note: Make sure the 'Output prefix' field contains a full path of the output directory and a filename prefix.

- Brain mask

  You can specify a signal (brain) mask NIfTI file.

  Note: A mask **must** be provided (either in the input directory or specified) in this standalone.

QSM panel



- Method:

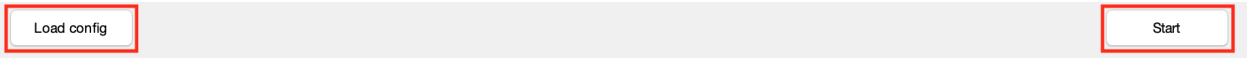
  Select a QSM dipole inversion method. The method parameters will be displayed on the method panel.

- Reference tissue

  Select a tissue for QSM value referencing.

  > **Warning:** The 'CSF' tissue option works only when **multi-echo** magnitude data is provided.

Others



- Load config

  Import the method related settings specified in the SEPIA-generated config file to the SEPIA GUI. **NO** modification will be made in the I/O panel.

- Start

  Generate a SEPIA config file that contains all user-defined methods and parameters for QSM processing based on the setting in the GUI. SEPIA will run the config file immediately once it is generated.

1.10 Phase Unwarpping Algorithms

1.10.1 Phase unwrapping in QSM

Phase wrapping occurs when continuous phase information is sampled in a discrete wrapped phase. The measured phase accumulation larger than one phase cycle is wrapped into the interval [-π, π), causing the discontinuity in the phase data. To recover the true phase values, one must solve this ambiguity problem by adding the correct integer number of phase cycles to the phase data in order to recover the true phase revolution.

The objective of this standalone application is to recover the actual, total phase shift of the acquired data.

1.10.2 Supported algorithms in SEPIA

Echo phase combination

Temporal phase unwrapping with multi-echo data

1. Optimum weights

   This is a weighted combination of the phase difference between successive echoes, in which the weights are inversely proportional to the variance of the noise of the fieldmap estimated from the magnitude echo images.

2. **MEDI nonlinear fit**

   This is a method in the MEDI toolbox

3. ROMEO total field calculation

   This is the method kindly provided from the ROMEO team. Please check the ROMEO GitHub page for the detailed arguments (settings).

Phase unwrapping

Spatial phase unwrapping

1. Laplacian (MEDI)

   Laplacian unwrapping implementation in MEDI toolbox

2. Laplacian (STI suite)

   Laplacian unwrapping implementation in STI Suite v3.0

3. 3D best path

   Robust region growing method yet only works in the DCCN cluster (recommended if you use this toolbox in the DCCN cluster)

4. **Region growing (MEDI)**

   Region growing method in the MEDI toolbox

5. Graphcut

   Graph-cut algorithm in the MEDI toolbox, sometimes uses with water-fat imaging.

6. SEGUE

7. ROMEO

   Using ROMEO for single phase unwrapping only.

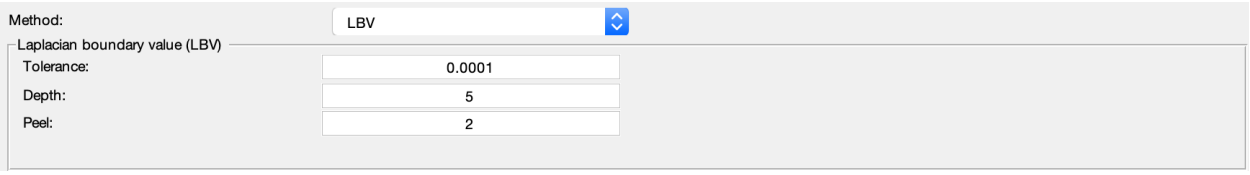
1.11 Background Magnetic Field Removal Algorithms

1.11.1 Background field removal in QSM

The phase we measured in a GRE acquisition is affected by not only the brain tissue but also sources like B0 inhomogeneity and air sinus. In order to compute the susceptibility sources only contributed by the brain tissue, it is important to remove all the non-local field effect. Fortunately, the characteristic of the local field is different from that of the non-local field, it is possible to separate the two fields from the unwrapped total field. This standalone application is designed to solve the field separation problem.

Caution: It is crucial that the background field contribution is removed accurately in this stage. Otherwise, the remaining field due to background sources will be treated as part of the local field, degrading the quality of QSM result.

Laplacian Boundary Value approach (LBV)



Reference

Zhou, D., Liu, T., Spincemaille, P., Wang, Y., 2014. Background field removal by solving the Laplacian boundary value problem. NMR in biomedicine 27, 312–319.

Algorithm parameters overview

algorParam.bfr.	Description
tol	Iteration stopping criteria on the coarest grid
depth	No. of length scales
peel	No. of boundary layers to be peeled off

Usage

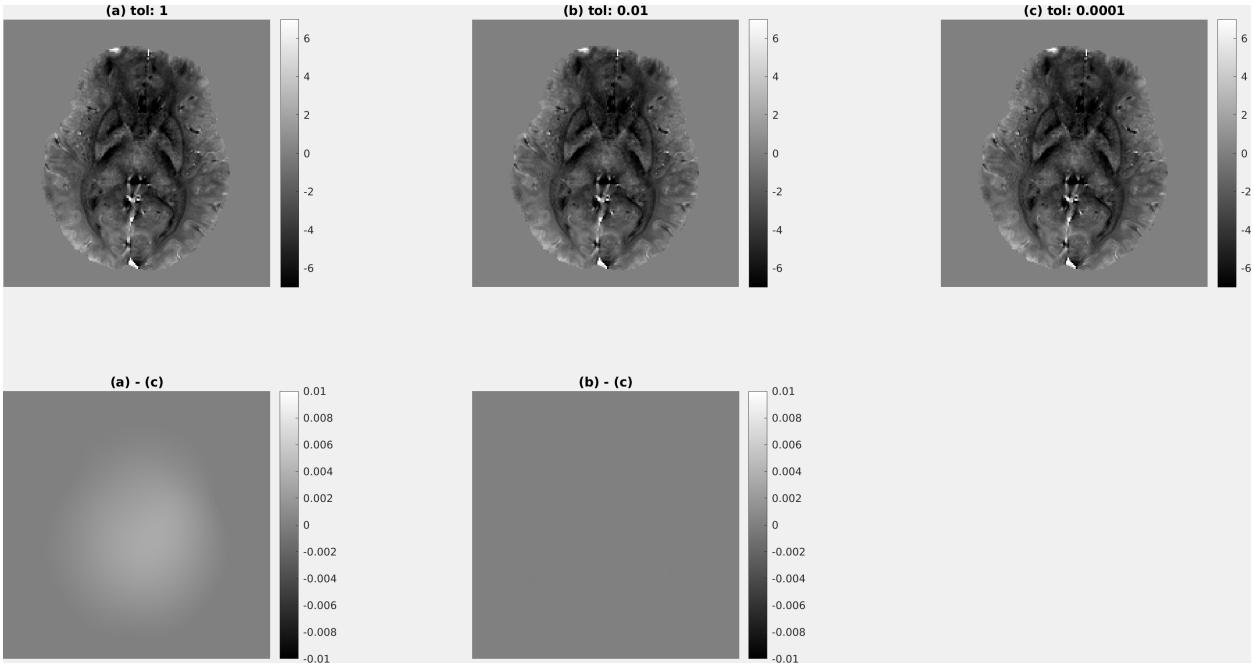
algorParam.bfr.tol

Iteration stopping criteria on the coarest grid

Default Value: 0.0001

Examples:

```
algorParam.bfr.tol = 1;
algorParam.bfr.tol = 0.01;
algorParam.bfr.tol = 0.0001;
```



algorParam.bfr.depth

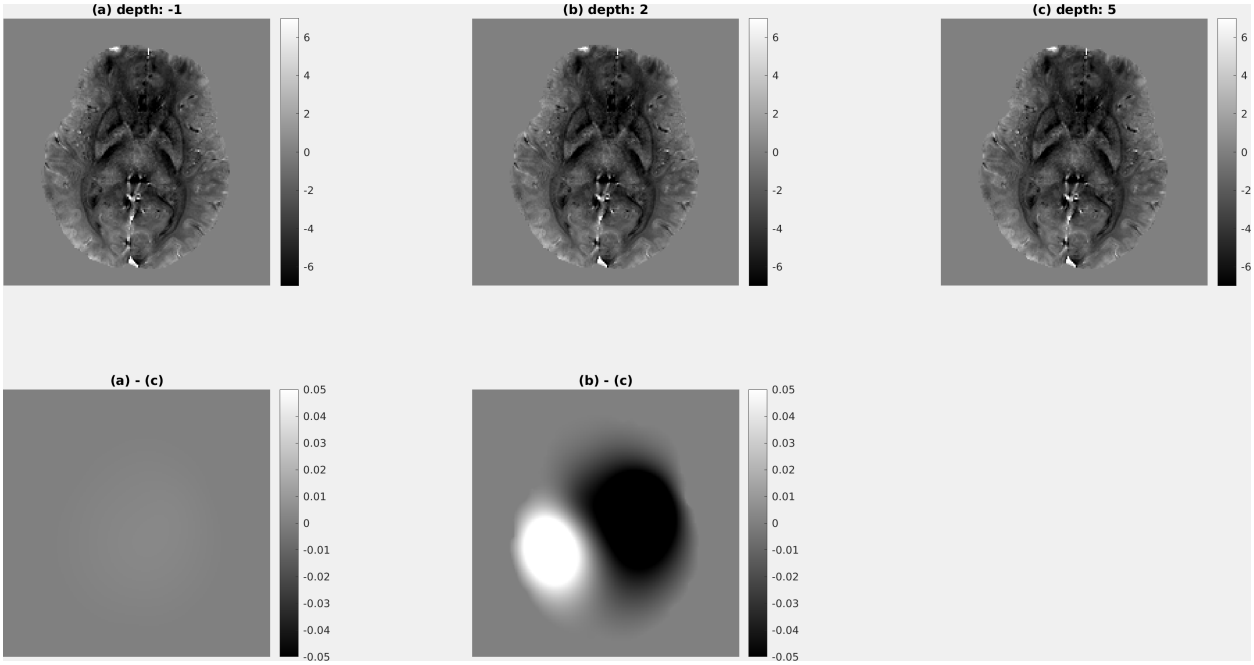
No. of length scales

Default Value: 5

Examples:

```
algorParam.bfr.depth = -1;
algorParam.bfr.depth = 2;
```

```
algorParam.bfr.depth = 5;
```



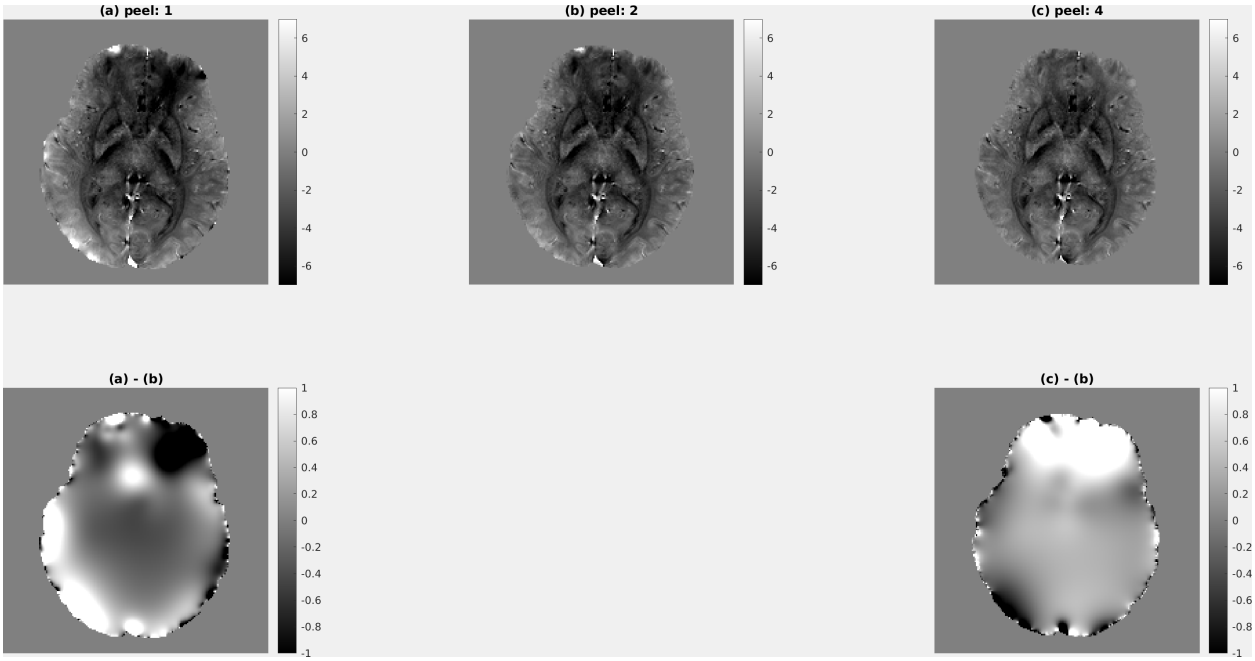
algorParam.bfr.peel

No. of boundary layers to be peeled off

Default Value: 2

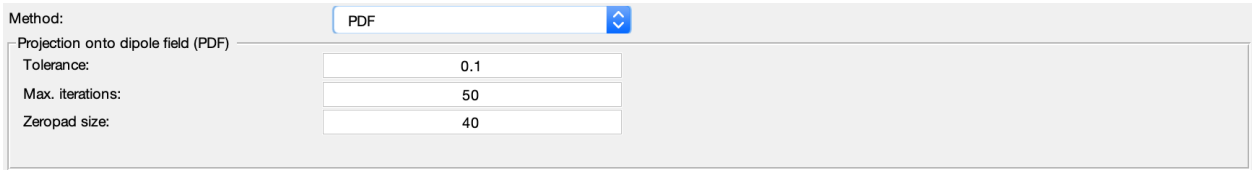
Examples:

```
algorParam.bfr.peel = 1;
algorParam.bfr.peel = 2;
algorParam.bfr.peel = 4;
```



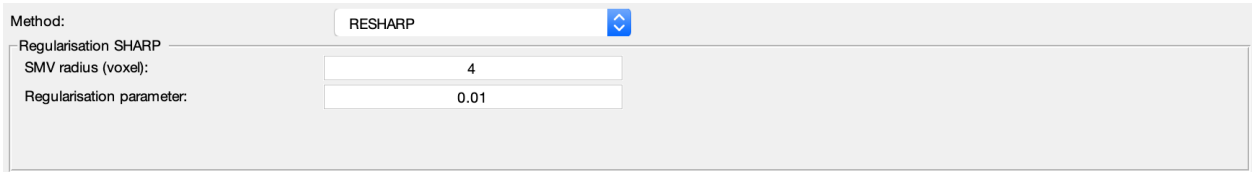
Projection onto Dipole Field (PDF)

Reference: Liu, T., Khalidov, I., Rochefort, L. de, Spincemaille, P., Liu, J., Tsiouris, A.J., Wang, Y., 2011. A novel background field removal method for MRI using projection onto dipole fields (PDF). NMR in biomedicine 24, 1129–1136.



Regularisation Enabled SHARP (RESHARP)

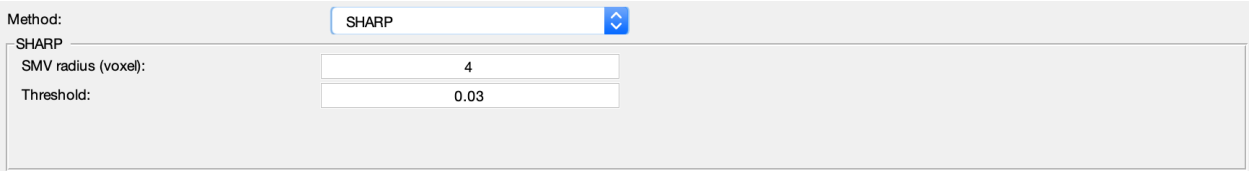
Reference: Sun, H., Wilman, A.H., 2014. Background field removal using spherical mean value filtering and Tikhonov regularization. Magnetic resonance in medicine 71, 1151–1157.



Sophisticated Harmonic Artefact Reduction for Phase data (SHARP)

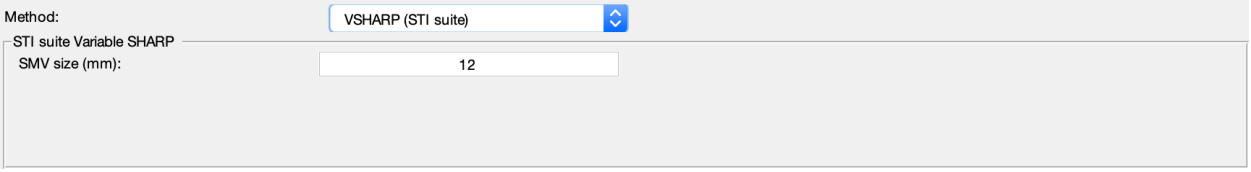
Reference: Schweser, F., Deistung, A., Lehr, B.W., Reichenbach, J.R., 2011. Quantitative imaging of intrinsic magnetic tissue properties using MRI signal phase: an approach to in vivo brain iron metabolism? Neuroimage 54,

2789–2807.



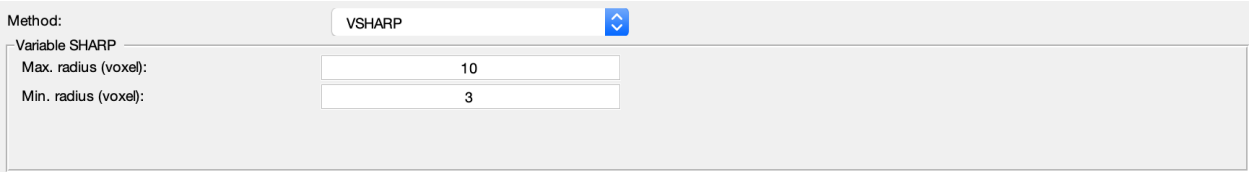
Variable-kernel SHARP (VSHARP STI Suite)

Reference: Li, W., Wu, B., Liu, C., 2011. Quantitative susceptibility mapping of human brain reflects spatial variation in tissue composition. Neuroimage 55, 1645–1656.



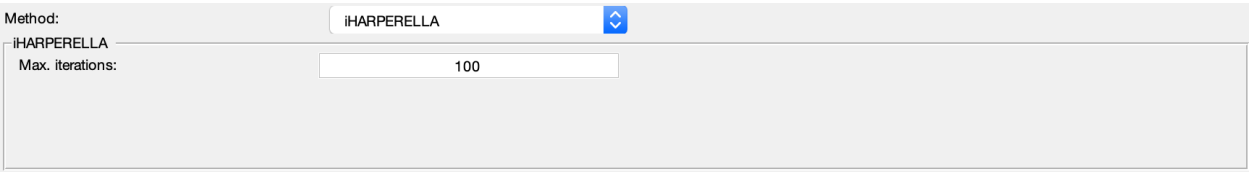
Variable-kernel SHARP (VSHARP)

Reference: Li, W., Wu, B., Liu, C., 2011. Quantitative susceptibility mapping of human brain reflects spatial variation in tissue composition. Neuroimage 55, 1645–1656.



Improved HARmonic (background) PhasE REmovaL using the LAplacian operator (iHARPERELLA)

Reference: Li, W., Avram, A.V., Wu, B., Xiao, X., Liu, C., 2014. Integrated Laplacian-based phase unwrapping and background phase removal for quantitative susceptibility mapping. NMR in biomedicine 27, 219–227.



BFRnet

Xuanyu Zhu, Yang Gao, Feng Liu, Stuart Crozier, Hongfu Sun, 2022. BFRnet: A deep learning-based MR background field removal method for QSM of the brain containing significant pathological susceptibility sources

Setup BFRnet for SEPIA

1. Download deepMRI from GitHub
2. Download the pre-trained BFRnet here as mentioned in the instruction on GitHub
3. Specify the full path to deepMRI code as 'deepMRI_HOME' in setup_BFRnet_environment.m in SEPIA_HOME/addons/bfr/BFRnet/
4. Specify the full path to the pre-trained network (should be checkpoints/BFRnet_L2_64PS_24BS_45Epo_NewHCmix.mat) from (2) as 'checkpoints' in setup_BFRnet_environment.m

Your setup_BFRnet_environment.m should look something like this:

```
%% This file specifies the Checkpoint file of BFRnet

% Specify the directory of the xQSM code from Github
deepMRI_HOME    = '/project/3015069.05/bids/code/deepMRI/';
% If you have your own trained network, specify the file that contains the trained parameters
% otherwise, use the one provided with the tool
checkpoints     = '/project/3015069.05/bids/code/deepMRI/BFRNet_data/checkpoints/BFRnet_L2_64PS_24BS_45Epo_NewHCmix.mat';
```

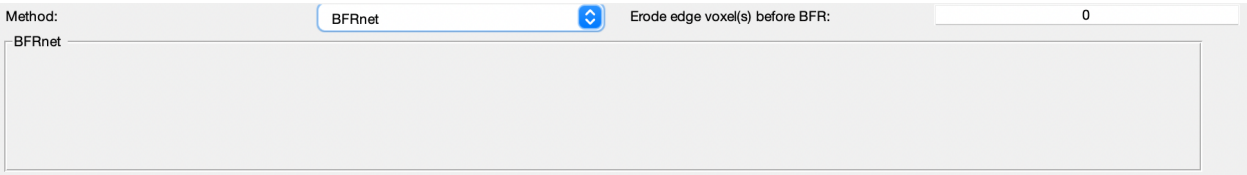
1. from GitHub

2. from Dropbox

Warning: The support this method is still in an early stage and only tested on a Linux machine.

BFRnet panel

There is no algorithm parameter needed to be adjusted with this tool at the moment.



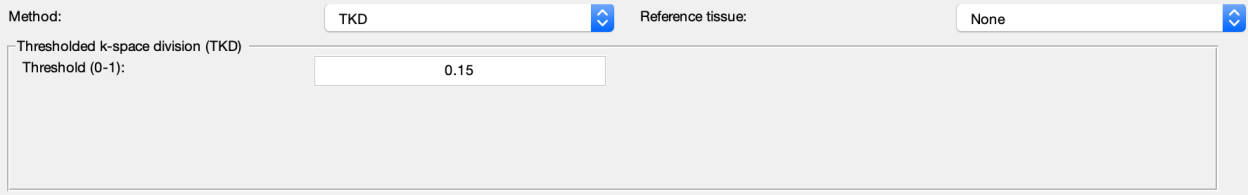
1.12 QSM Algorithms

1.12.1 How can we map the magnetic susceptibility sources from local field map?

A local field (or tissue field) is the field generated by the magnetic susceptibility sources from the brain tissue. The assumption of the field generated by a magnetic susceptibility point source is a dipole field. We can then try to compute the susceptibility values by deconvolving the local field with a dipole field. However, due to the fact that the dipole field in the k-space (i.e. the dipole kernel) has zero values on the cone surface at 54.7° and values close to this surface will also be close to zeros. As a result, the number of unknown parameters is more than the measurements we have in the data, leading to the problem ill-conditioned and the resulting QSM map has the so-called streaking artefact. To solve the ill-conditioned problem, most of the QSM algorithms try to add a regularisation term in the QMS dipole inversion problem, imposing spatial smoothness in the QSM map and/or the QSM map has similar anatomical features as shown in the magnitude images in order to reduce the streaking artefact in the QSM result.

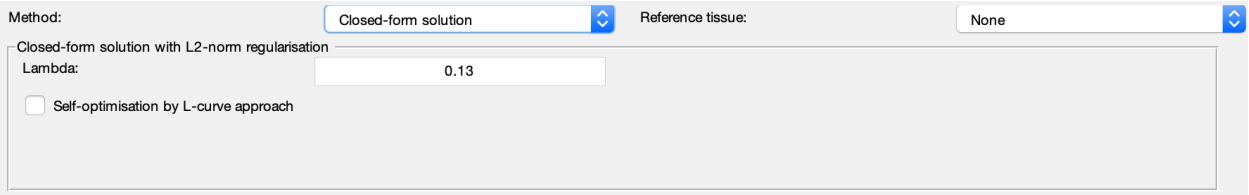
Thresholded k-space Division (TKD)

Reference: Wharton, S., Schäfer, A., Bowtell, R., 2010. Susceptibility mapping in the human brain using threshold-based k-space division. Magnetic resonance in medicine 63, 1292–1304.



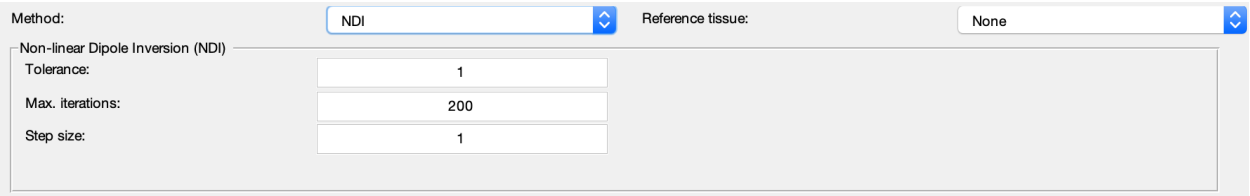
Closed-form solution with L2-norm regularisation

Reference: Bilgic, B., Chatnuntawech, I., Fan, A.P., Setsompop, K., Cauley, S.F., Wald, L.L., Adalsteinsson, E., 2014. Fast image reconstruction with L2-regularization. Journal of magnetic resonance imaging: JMRI 40, 181–191.



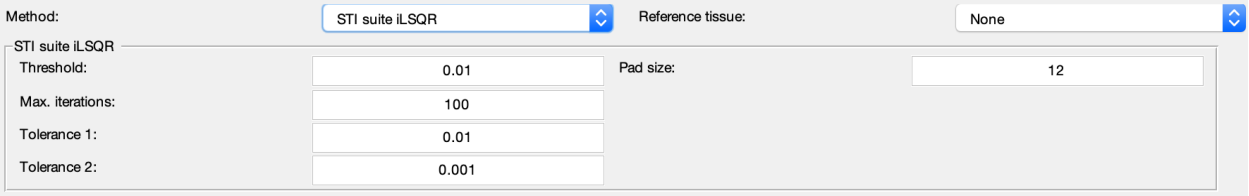
Nonlinear Dipole Inversion (NDI)

Reference: Polak, D., Chatnuntawech, I., Yoon, J., Iyer, S.S., Milovic, C., Lee, J., Bachert, P., Adalsteinsson, E., Setsompop, K., Bilgic, B., 2020. Nonlinear dipole inversion (NDI) enables robust quantitative susceptibility mapping (QSM). Nmr Biomed e4271.



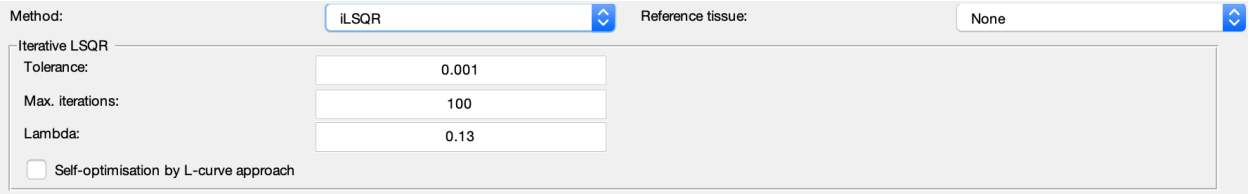
Iterative LSQR (iLSQR STI Suite)

Reference: Li, W., Wu, B., Liu, C., 2011. Quantitative susceptibility mapping of human brain reflects spatial variation in tissue composition. Neuroimage 55, 1645–1656.



Iterative LSQR (iLSQR)

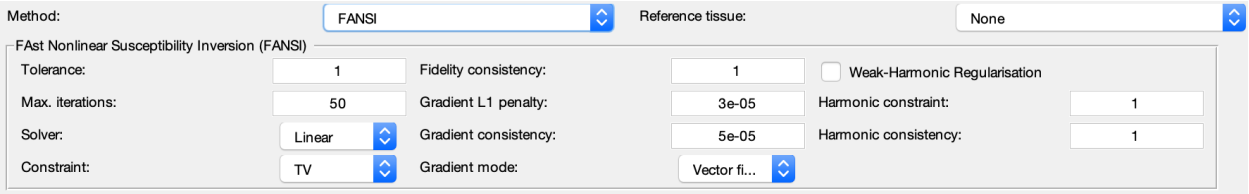
Reference: Li, W., Wu, B., Liu, C., 2011. Quantitative susceptibility mapping of human brain reflects spatial variation in tissue composition. Neuroimage 55, 1645–1656.



FAst Nonlinear Susceptibility Inversion (FANSI)

References: Milovic, C., Bilgic, B., Zhao, B., Acosta-Cabronero, J., Tejos, C., 2018. Fast nonlinear susceptibility inversion with variational regularization. Magnetic resonance in medicine 80, 814–821.

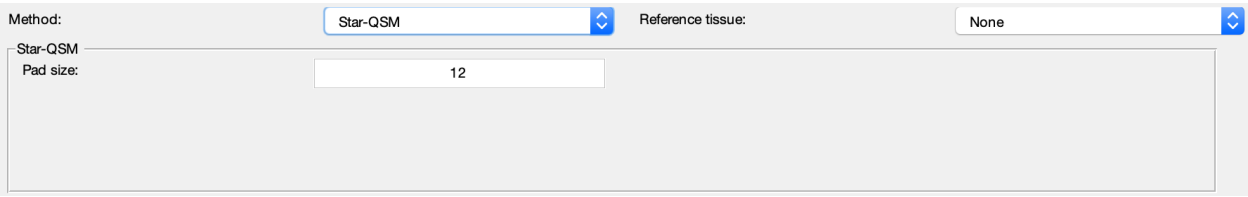
Milovic, C., Bilgic, B., Zhao, B., Langkammer, C., Tejos, C., Cabronero, J.A., 2019. Weak-harmonic regularization for quantitative susceptibility mapping. Magnetic resonance in medicine 81, 1399–1411.



Note: Current algorithm parameters are adapted for FANSI v3. Please refer to FANSI v1 for recommended parameter values if you used FANSI v1.

STreaking Artifact Reduction for QSM (Star-QSM)

Reference: Wei, H., Dibb, R., Zhou, Y., Sun, Y., Xu, J., Wang, N., Liu, C., 2015. Streaking artifact reduction for quantitative susceptibility mapping of sources with large dynamic range. NMR in biomedicine 28, 1294–1303.

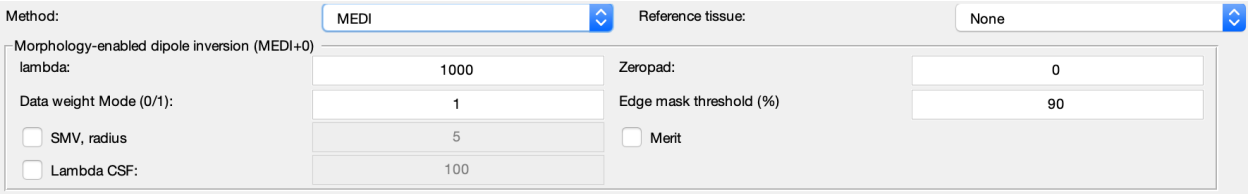


Morphology enabled dipole inversion (MEDI)

References: Liu, T., Liu, J., Rochefort, L. de, Spincemaille, P., Khalidov, I., Ledoux, J.R., Wang, Y., 2011. Morphology enabled dipole inversion (MEDI) from a single-angle acquisition: Comparison with COSMOS in human brain imaging. Magnetic resonance in medicine 66, 777–783.

LLiu, J., Liu, T., Rochefort, L. de, Ledoux, J., Khalidov, I., Chen, W., Tsiouris, A.J., Wisnieff, C., Spincemaille, P., Prince, M.R., Wang, Y., 2012. Morphology enabled dipole inversion for quantitative susceptibility mapping using structural consistency between the magnitude image and the susceptibility map. Neuroimage 59, 2560–2568.

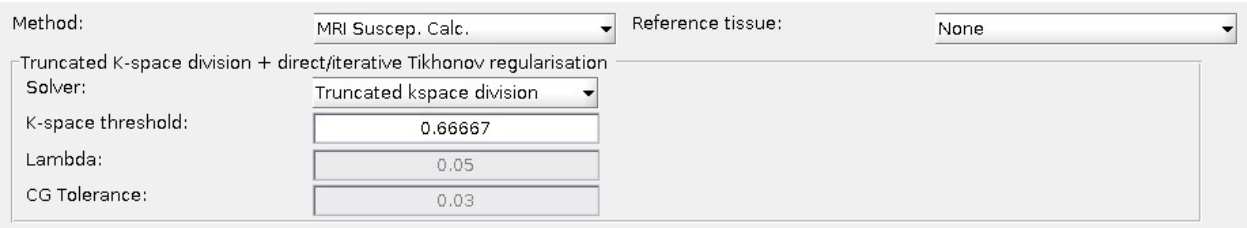
Liu, Z., Spincemaille, P., Yao, Y., Zhang, Y., Wang, Y., 2018. MEDI+0: Morphology enabled dipole inversion with automatic uniform cerebrospinal fluid zero reference for quantitative susceptibility mapping. Magnetic resonance in medicine 79, 2795–2803.



MRI Susceptibility Calculation Methods

Reference: For the TKD software implementation, the following citation shall be included in the acknowledgements: Shmueli, K et al. (2009). Magnetic susceptibility mapping of brain tissue in vivo using MRI phase data, Magnetic Resonance in Medicine vol 62 issue 6, 1510-1522 and Schweser, F et al. (2013). Toward online reconstruction of quantitative susceptibility maps: superfast dipole inversion, Magnetic Resonance in Medicine vol 69 issue 6, 1581-1593.

For the dirTik and iterTik software implementations in the package, the following citation shall be included in the acknowledgements: Karsa, A et al. (2019). High Repeatability of Quantitative Susceptibility Mapping (QSM) in the Head and Neck With a View to Detecting Hypoxic Cancer Sites, In Proceedings of the 27th Annual Meeting of the ISMRM, Montreal, p. 4939 and Schweser, F et al. (2013). Toward online reconstruction of quantitative susceptibility maps: superfast dipole inversion, Magnetic Resonance in Medicine vol 69 issue 6, 1581-1593."



QSMnet+

J. Yoon, E. Gong, I. Chatnuntawech, B. Bilgic, J. Lee, W. Jung, J. Ko, H. Jung, K. Setsompop, G. Zaharchuk, E.Y. Kim, J. Pauly, J. Lee. Quantitative susceptibility mapping using deep neural network: QSMnet. Neuroimage. 2018 Oct;179:199-206.

W. Jung, J. Yoon, S. Ji, J. Choi, J. Kim, Y. Nam, E. Kim, J. Lee. Exploring linearity of deep neural network trained QSM: QSMnet+. Neuroimage. 2020 May; 116619.

W. Jung, S. Bollmann, J. Lee. Overview of quantitative susceptibility mapping using deep learning: Current status, challenges and opportunities. NMR in Biomedicine. 2020 Mar; e4292.

Setup QSMnet+ for SEPIA

1. Download QSMnet+ from GitHub

2. If you haven't setup QSMnet+ in python, following the instruction in https://github.com/SNU-LIST/QSMnet, including downloading the pre-trained network and creating conda environment (see Section Manual in their GitHub page)
3. Specify the full path to QSMnet+ code as 'QSMnet_HOME' in setup_qsmnet_environment.m in SEPIA_HOME/addons/qsm/QSMnet/
4. Specify the full path of the Python interpreter that has QSMnet installed as 'python_interpreter' in setup_qsmnet_environment.m

Your setup_qsmnet_environment.m should look something like this:

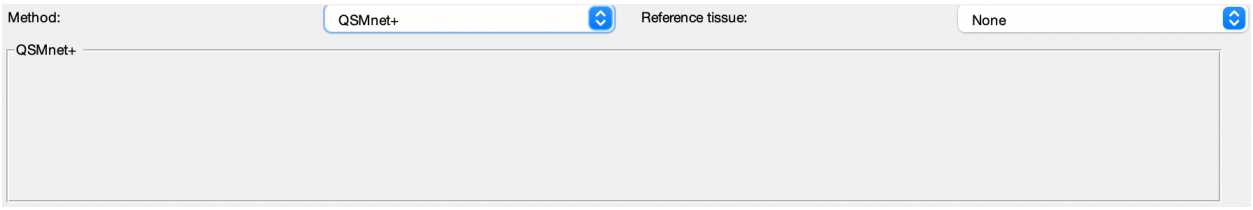
```
%% This file specifies the Python environment and Checkpoint directory of QSMnet+

% Specify the directory of the QSMnet+ code from Github
QSMnet_HOME         = '/project/3015069.05/bids/code/QSMnet/';
% Specify the Python environment that has QSMnet+ installed
python_interpreter  = '/project/3015069.05/bids/code/QSMnet/qsmnet/bin/python';
% Specify the directory that contains the trained parameters
dir_net             = fullfile(QSMnet_HOME,'Checkpoints/');
```

1. from GitHub
2. Interpreter from conda env
This should come with 1.

Warning: The support this method is still in an early stage and only tested on a Linux machine.

QSMnet+ panel

There is no algorithm parameter needed to be adjusted with this tool at the moment.



LP-CNN

Kuo-Wei Lai, Manisha Aggarwal, Peter van Zijl, Xu Li & Jeremias Sulam, 2020. Learned Proximal Networks for Quantitative Susceptibility Mapping

Setup LP-CNN for SEPIA

1. Download LP-CNN from GitHub
2. If you haven't setup LP-CNN in python, following the instruction in https://github.com/Sulam-Group/LPCNN, to create conda environment (see Section Environment Settings in their GitHub page)
3. Specify the full path to LP-CNN code as 'LPCNN_HOME' in setup_LPCNN_environment.m in SEPIA_HOME/addons/qsm/LPCNN/

4. Specify the full path of the Python interpreter that has LP-CNN installed as 'python_interpreter' in setup_LPCNN_environment.m

Your setup_LPCNN_environment.m should look something like this:

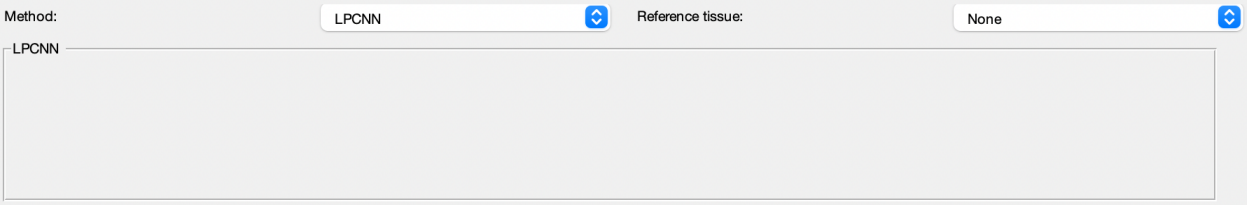
```
%% This file specifies the Python environment and Checkpoint of LPCNN

% Specify LPCNN_HOME corresponding to the code from Github
LPCNN_HOME          = '/project/3015069.05/bids/code/LPCNN/';
% Specify the Python environment that has LPCNN installed
python_interpreter  = '/project/3015069.05/bids/code/LPCNN/lpcnn-env/bin/python';
% Specify the directory that contains the training parameters
checkpoint_fn       = fullfile(LPCNN_HOME,'checkpoints','lpcnn_test_Bmodel.pkl');
% checkpoint_fn       = '/project/3015069.05/bids/code/LPCNN/checkpoints/lpcnn_test_Bmodel.pkl';
```

1. from GitHub
2. Interpreter of conda env.
This should come with 1.

Warning: The support this method is still in an early stage and only tested on a Linux machine.

LP-CNN panel

There is no algorithm parameter needed to be adjusted with this tool at the moment.



xQSM

Yang Gao, Xuanyu Zhu, Bradford A. Moffat, Rebecca Glarin, Alan H. Wilman, G. Bruce Pike, Stuart Crozier, Feng Liu, Hongfu Sun, 2020. xQSM: quantitative susceptibility mapping with octave convolutional and noise-regularized neural networks.

Setup xQSM for SEPIA

1. Download deepMRI from GitHub
2. Download the pre-trained xQSM here as mentioned in the instruction on GitHub
3. Specify the full path to deepMRI code as 'deepMRI_HOME' in setup_xQSM_environment.m in SEPIA_HOME/addons/qsm/xQSM/
4. Specify the full path to the folder containing the pre-trained networks (should be checkpoints/) from (2) as 'checkpoints_dir' in setup_xQSM_environment.m

Your setup_xQSM_environment.m should look something like this:

```
%% This file specifies the Checkpoint directory of xQSM

% Specify the directory of the xQSM code from Github
deepMRI_HOME    = '/project/3015069.05/bids/code/deepMRI/';
% If you have your own trained network, specify the file that contains the trained parameters
% otherwise, uses the one provided with the tool
checkpoint_dir  = '/project/3015069.05/bids/code/deepMRI/xQSM_data/checkpoints';
```

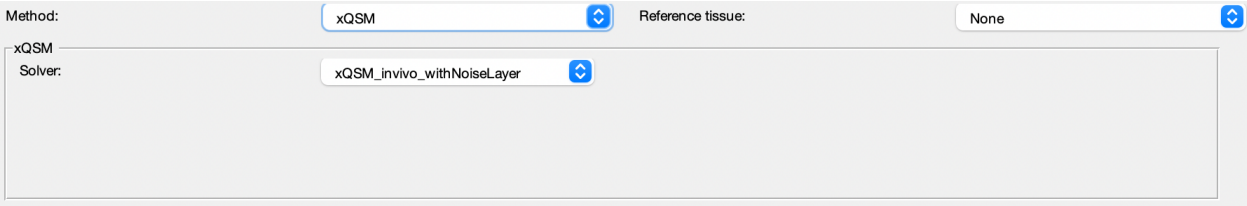
1. from GitHub

2. from Dropbox

Warning: The support this method is still in an early stage and only tested on a Linux machine.

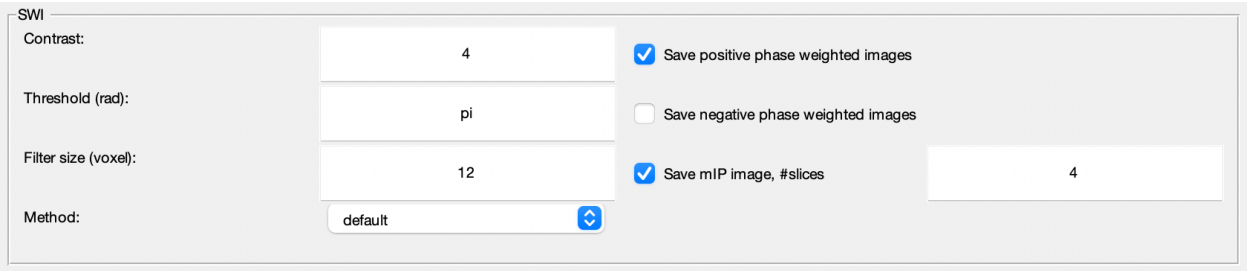
xQSM panel

In the xQSM panel you can select one of the five provided pre-trained networks.



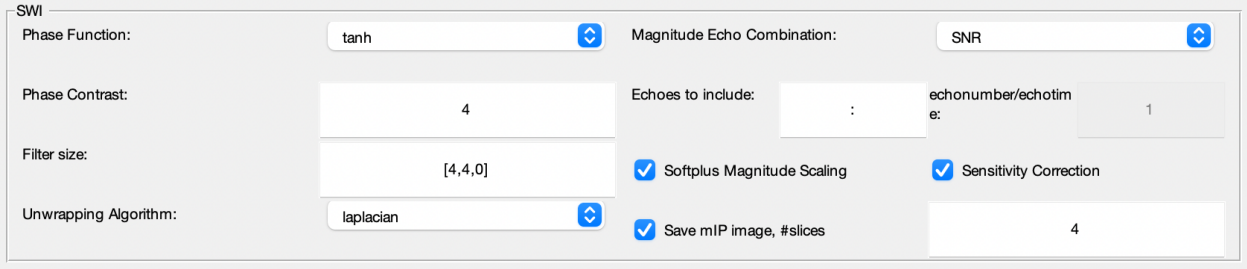
1.13 SWI/SMWI Algorithms

1.13.1 SWI (2D Hamming)



1.13.2 CLEAR-SWI

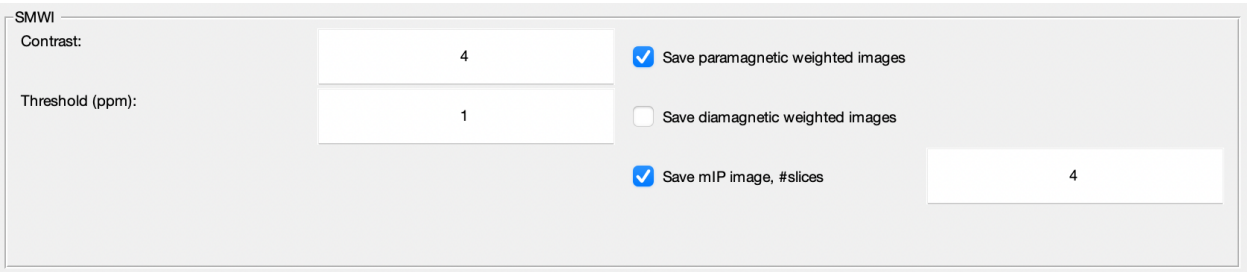
Reference: Korbinian Eckstein, Beata Bachrata, Gilbert Hangel, Georg Widhalm, Christian Enzinger, Markus Barth, Siegfried Trattnig, Simon Daniel Robinson, 2021. Improved susceptibility weighted imaging at ultra-high field using bipolar multi-echo acquisition and optimized image processing: CLEAR-SWI



Pleas refer to https://github.com/korbinian90/CLEARSWI.jl to see more information regarding the options/parameters of this method.

1.13.3 SMWI

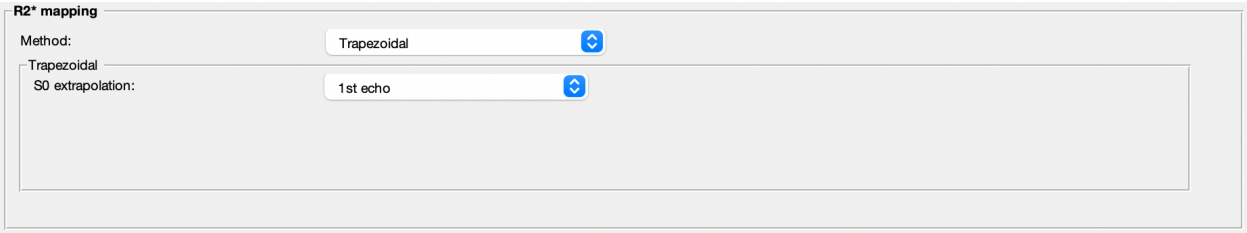
Reference: Gho, S.-M., Liu, C., Li, W., Jang, U., Kim, E.Y., Hwang, D., Kim, D.-H., 2014. Susceptibility map-weighted imaging (SMWI) for neuroimaging. Magnetic resonance in medicine 72, 337–346.



1.14 R2* Algorithms

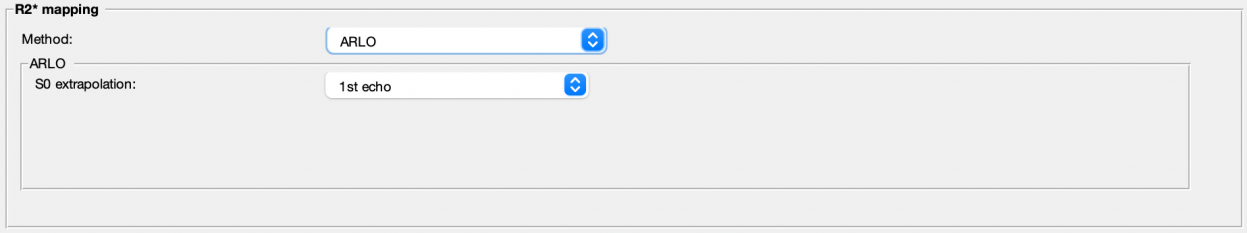
1.14.1 Closed-form solutino using trapezoidal approximation

Pei, M., Nguyen, T.D., Thimmappa, N.D., Salustri, C., Dong, F., Cooper, M.A., Li, J., Prince, M.R., Wang, Y., 2014. Algorithm for fast monoexponential fitting based on Auto-Regression on Linear Operations (ARLO) of data. Magnetic resonance in medicine 73, 843–850.



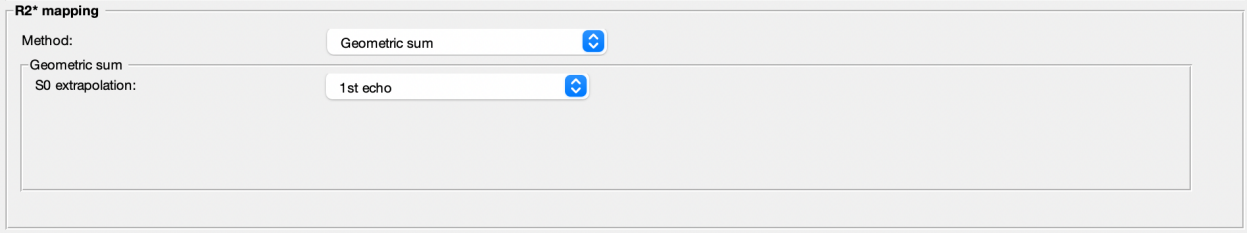
1.14.2 Algorithm for fast monoexponential fitting based on Auto-Regression on Linear Operations (ARLO) of data

Pei, M., Nguyen, T.D., Thimmappa, N.D., Salustri, C., Dong, F., Cooper, M.A., Li, J., Prince, M.R., Wang, Y., 2014. Algorithm for fast monoexponential fitting based on Auto-Regression on Linear Operations (ARLO) of data. Magnetic resonance in medicine 73, 843–850.



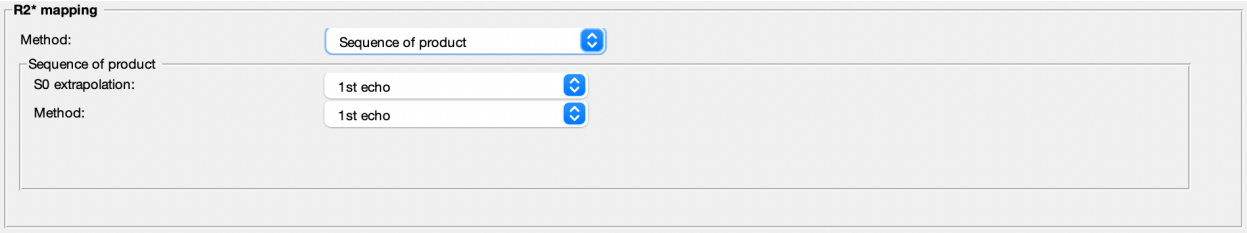
1.14.3 Closed-form solution using Geometric Sum

There is no reference for this method. I created this to test different closed-form solution. Should only be applicable to data with evenly distributed TE.



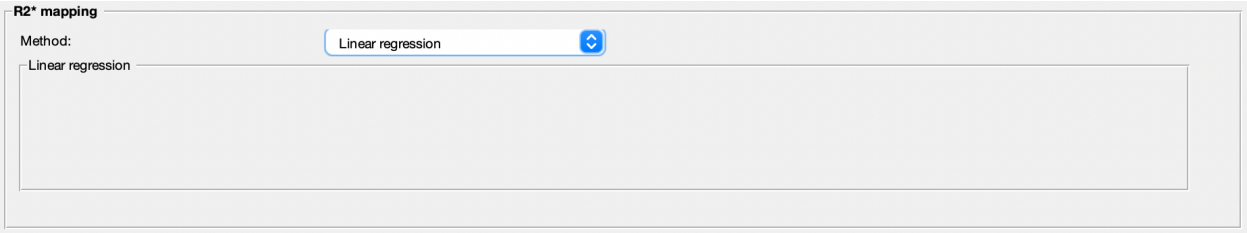
1.14.4 Closed-form solution using Sequence of Product

There is no reference for this method. I created this to test different closed-form solution.



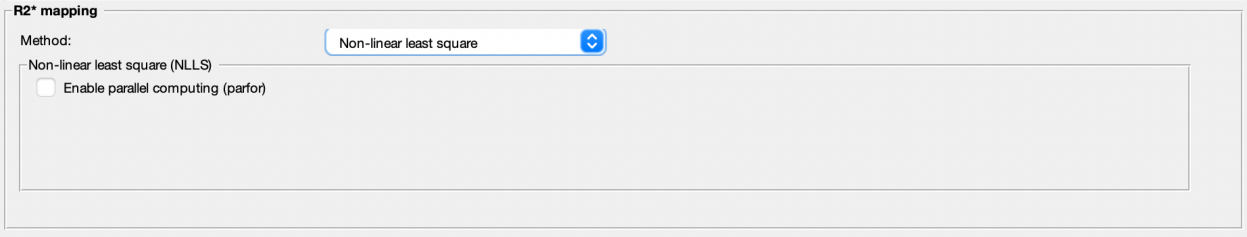
1.14.5 Linear regression

Mono-exponential R2* fitting with a linear model (in Logarithm space)



1.14.6 Non-linear least square

Mono expoential R2* mapping with a non-linear least square solver.



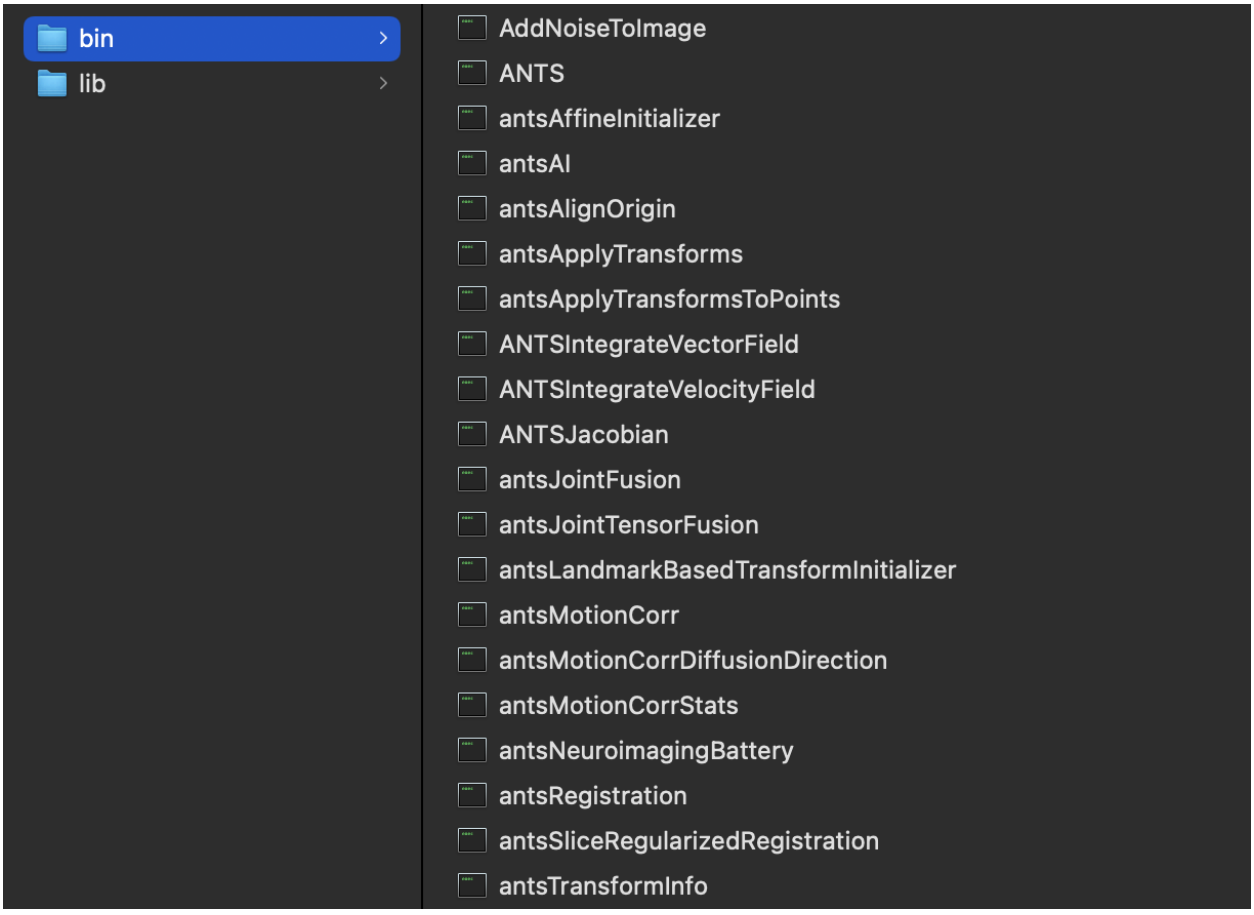
1.15 Subcortical structure segmentation in SEPIA

Starting from v1.2, SEPIA supports several atlas-based subcortical structure segmentation methods, based on non-linear registration using ANTs.

1.15.1 Set-up SEPIA for subcortical structure segmentation

Before using any of these methods, 'ANTS_HOME' has to be specified in SpecifyToolboxesDirectory.m or using the Manage Dependency tool in the Utility tab.

The path required for 'ANTS_HOME' should be the bin folder of ANTs that contains all the libraries, e.g.,



and in the Manage Dependency tool it should be like

Note: If you don't have ANTs, you can still use SEPIA for QSM reconstruction but you can't use the segmentation methods provided in the Analysis tab.

Then you need to download the atlases from their corresponding online sources. For Mac and Linux users, this can be done by running the shell script download_atlas.sh in the SEPIA_HOME folder. Start a command window and enter the following:

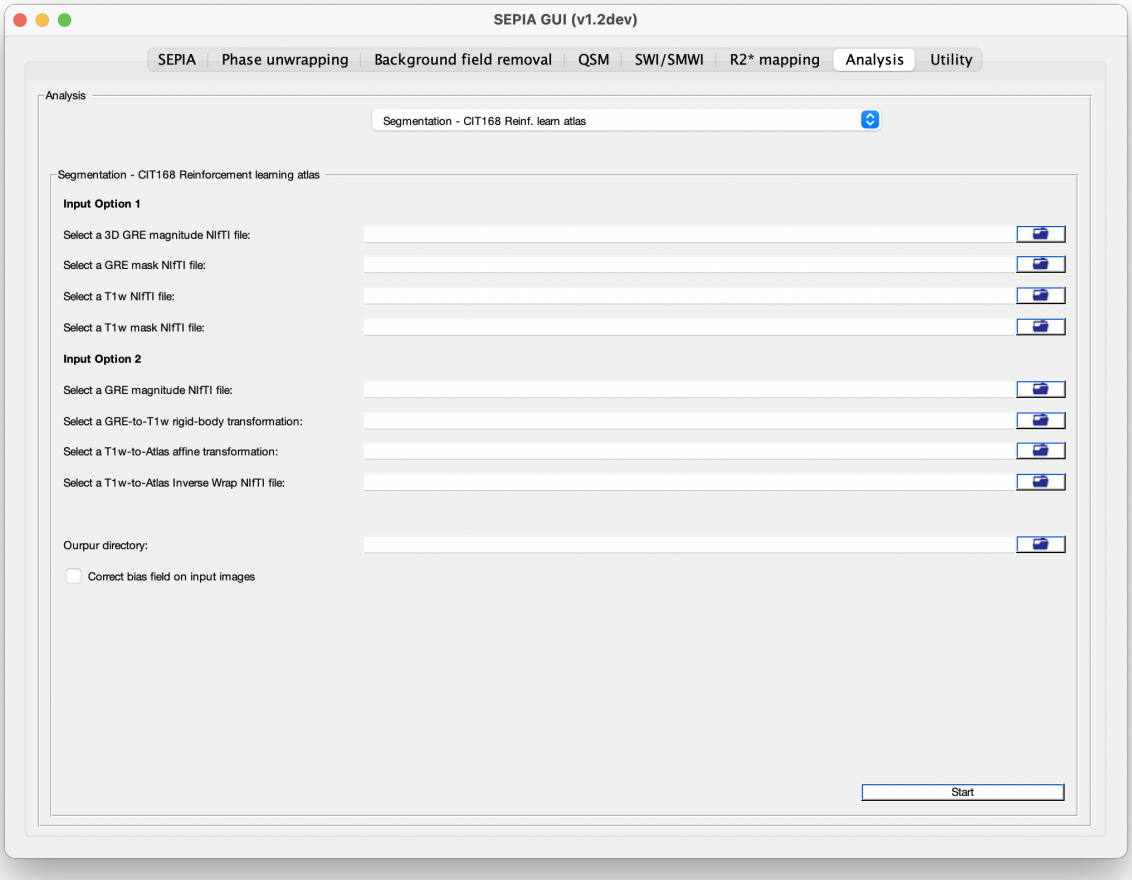
```
sh download_atlas.sh
```

By default, the atlases will be downloaded in SEPIA_HOME/atlas/. Make sure you have enough disk space in your computer and do not alter the location where the atlases stored.

There are currently three atlases supported:

CIT168 Reinforcement learning atlas

Pauli, W.M., Nili, A.N., Tyszka, J.M., 2018. A high-resolution probabilistic in vivo atlas of human subcortical brain nuclei. Sci Data 5, 180063.



For details regarding the atlas and labels, please refer to the reference above.

There are two possible approaches to bring the atlas labels to the GRE space, which approach will be used depending on the input of the user.

Approach 1: Proving NIFTI data input

Data required:

- a 3D GRE magnitude NIFTI image (e.g. 1st echo)
- a 3D GRE NIFTI brain mask
- a T1w NIFTI images
- a T1w NIFTI brain mask

Procedures:

1. (optional) Bias field correction on both T1w and GRE image using N4BiasFieldCorrection.
2. Coregistration between GRE image and T1w image using rigid-body transformation.
3. Coregistration between T1w image and the T1w image provided with the atlas using nonlinear transformation (SyN).
4. Applying the derived transformation matrices to bring the atlas labels to the GRE space.

Approach 2: Proving the transformation information derived from ANTs

Data required:

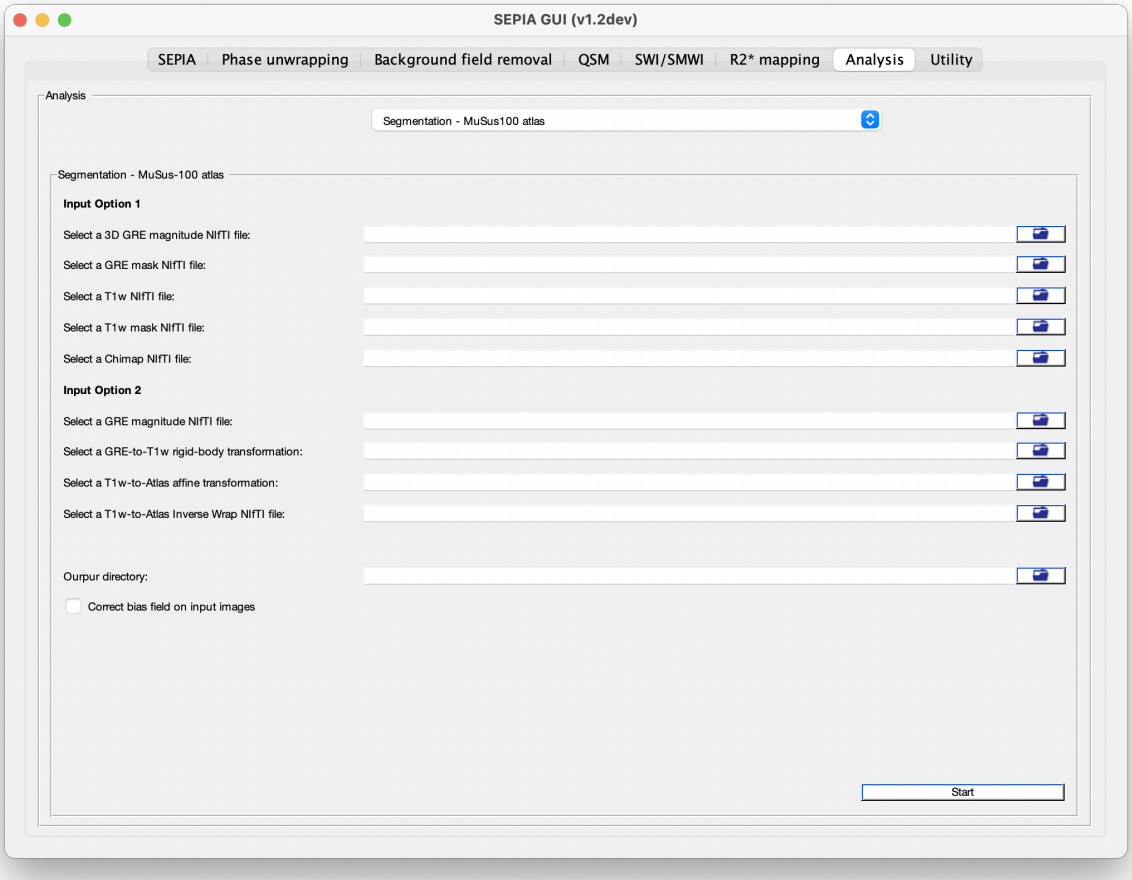
- a 3D GRE magnitude NIFTI image (to define the final space only)
- a GRE to T1w rigid-body transformation matrix file (usually with suffix *_0GenericAffine.mat*)
- a T1w to atlas T1w template affine transformation matrix file (usually with suffix *_0GenericAffine.mat*)
- a T1w to atlas T1w template inverse wrap field (usually with suffix *_1InverseWarp.nii.gz*)

Procedure:

1. The provided transformation information is used to bring the atlas labels to the GRE space.

Multi-modal-fused magnetic Susceptibility (MuSus-100) atlas

He, C., Guan, X., Zhang, W., Li, J., Liu, C., Wei, H., Xu, X., Zhang, Y., 2022. Quantitative susceptibility atlas construction in Montreal Neurological Institute space: towards histological-consistent iron-rich deep brain nucleus subregion identification. Brain Struct Funct 1–23.



For details regarding the atlas and labels, please refer to the reference above.

There are two possible approaches to bring the atlas labels to the GRE space, which approach will be used depending on the input of the user.

Approach 1: Proving NIFTI data input

Data required:

- a 3D GRE magnitude NIFTI image (e.g. 1st echo)
- a 3D GRE NIFTI brain mask
- a T1w NIFTI images
- a T1w NIFTI brain mask
- a Chimap NIFTI image

Procedures:

1. (optional) Bias field correction on both T1w and GRE image using N4BiasFieldCorrection.
2. Coregistration between GRE image and T1w image using rigid-body transformation.
3. Bringing Chimap to T1w space and create a hybrid image using the following equation:

$$hybrid = T1w - \mu Chi \tag{1.1}$$

   where T1w is the T1w image normalised to range [0,255] (clipped at 0.5 and 99.5 percentile of masked voxels), mu = 400 and Chi is the magnetic susceptibility map in ppm.

4. Coregistration between the hybrid image and the template hybrid image provided with the atlas using nonlinear transformation (SyN).
5. Applying the derived transformation matrices to bring the atlas labels to the GRE space.

Approach 2: Proving the transformation information derived from ANTs

Data required:

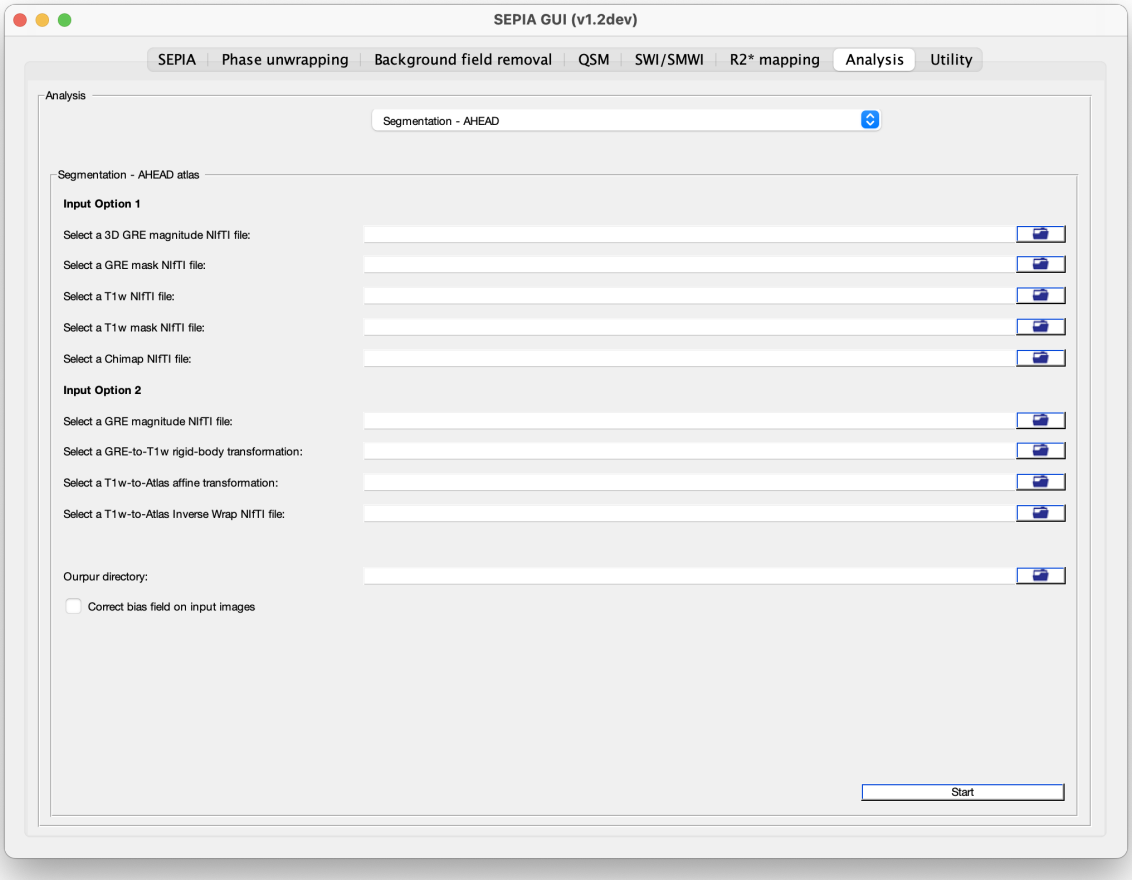
- a 3D GRE NIFTI image (to define the final space only)
- a GRE to T1w rigid-body transformation matrix file (usually with suffix *_0GenericAffine.mat*)
- a T1w to atlas T1w template affine transformation matrix file (usually with suffix *_0GenericAffine.mat*)
- a T1w to atlas T1w template inverse wrap field (usually with suffix *_1InverseWarp.nii.gz*)

Procedure:

1. The provided transformation information is used to bring the atlas labels to the GRE space.

Amsterdam Ultra-high field adult lifespan database (AHEAD) atlas

Alkemade, A., Mulder, M.J., Groot, J.M., Isaacs, B.R., Berendonk, N. van, Lute, N., Isherwood, S.J., Bazin, P.-L., Forstmann, B.U., 2020. The Amsterdam Ultra-high field adult lifespan database (AHEAD): A freely available multimodal 7 Tesla submillimeter magnetic resonance imaging database. Neuroimage 221, 117200.



For details regarding the atlas and labels, please refer to the reference above.

Note: The AHEAD atlas has very high resolution (0.5 mm isotropic). It will take longer compared to other atlas to obtain the labels.

There are two possible approaches to bring the atlas labels to the GRE space, which approach will be used depending on the input of the user.

Approach 1: Proving NIFTI data input

Data required:

- a 3D GRE magnitude NIFTI image (e.g. 1st echo)
- a 3D GRE NIFTI brain mask
- a T1w NIFTI images
- a T1w NIFTI brain mask
- a Chimap NIFTI image

Procedures:

1. (optional) Bias field correction on both T1w and GRE image using N4BiasFieldCorrection.
2. Coregistration between GRE image and T1w image using rigid-body transformation.
3. Bringing Chimap to T1w space and create a hybrid image using the following equation:

$$hybrid = T1w - \mu Chi \tag{1.2}$$

where T1w is the T1w image normalised to range [0,255] (clipped at 0.5 and 99.5 percentile of masked voxels), mu = 400 and Chi is the magnetic susceptibility map in ppm.

4. Create a hybrid image in the template space using the median R1 and Chi maps using the same equation as above.
5. Coregistration between the hybrid image and the template hybrid image provided with the atlas using nonlinear transformation (SyN).
6. Applying the derived transformation matrices to bring the atlas labels to the GRE space.

Approach 2: Proving the transformation information derived from ANTs

Data required:

- a 3D GRE NIFTI image (to define the final space only)
- a GRE to T1w rigid-body transformation matrix file (usually with suffix *_0GenericAffine.mat*)
- a T1w to atlas T1w template affine transformation matrix file (usually with suffix *_0GenericAffine.mat*)
- a T1w to atlas T1w template inverse wrap field (usually with suffix *_1InverseWarp.nii.gz*)

Procedure:

1. The provided transformation information is used to bring the atlas labels to the GRE space.

1.16 Weightings in SEPIA

1.16.1 How does SEPIA compute the weights for dipole field inversion algorithms?

SEPIA utilises the inverse of the fieldmap standard deviation map (noisesd.nii.gz) as the weights for potential dipole field inversion algorithms usage. Specifically, The following steps are applied:

Step 1

Invert the fieldmap standard deviation and remove non-value entries (i.e., NaN & inf become 0)

$$weights = \frac{1}{fieldmapSD} \tag{1.3}$$

Step 2

Normalisation of the weights. To establish more comparable weights between subjects and between protocols, the weights are first normalised by the value defined as (median + 3IQR). Because of the fast R2* tissues (e.g., globus

pallidus), the histogram of the weights is usually negatively skewed. The threshold of (median +3IQR) should capture most of the brain issue <= 1.

$$weights = \frac{weights}{median(weights(mask)) + 3 \times IQR(weights(mask))} \tag{1.4}$$

Step 3

To avoid the weights estimated from vaious echo combination methods and dataset having significant differences in magnitude overall, the median of the histogram of the weights is re-centred to 1.

$$weights = weights - median(weights(mask)) + 1 \tag{1.5}$$

Step 4

The last step is to reduce the extreme values on the right hand side of the histogram, which can introduce an overall weight offset between different echo combination methods if the dipole field inversion methods performs weight normalisation using the maximum value of the input data (e.g., FANSI). Since the weights (to be more precisely, the fieldmap standard deviation, noisesd.nii.gz) are commonly derived from the magnitude data, these extreme values often correspond to the fresh spins in the arteries and spatially sparse. To reduce the extreme values while preserving the smoothness of the weighting map, the weighting map is thresholded (threshold defined as median + 3IQR) and the extreme values are replaced by a 3x3x3 (voxel) box filtered copy:

$$weights(weights > threshold) = weights_{smooth}(weights > threshold) \tag{1.6}$$

where

$$weights_{smooth} = smooth3(weight. * mask) \tag{1.7}$$

And this is the final output of the weights. Since the median after normalisation will be less than 1. Therefore, the minimum value of the weights will not be equal to zero, roughly speaking, most gray matter and white matter could have weight ~1; globus pallidus, red nucleus and substantia nigra ~0.7-0.9; veneous structures ~0.3-0.6.

1.16.2 Further modulation on the weighting maps

It is possible to further adjust the weighting map in SEPIA if a "quantitative unwrapping method" is chosen (e.g., SEGUE, ROMEO & region growing). This can be done by checking the "Exclude voxels using residual" box and select "Weighting map" as the data to be applied. Here, the residual is the relative residual of fitting the multi-echo data with a mono-exponential model:

$$relativeresidual = \frac{\sum^t |\hat{S}(TE) - S(TE)|^2}{\sum^t |S(TE)|^2} \tag{1.8}$$

where S(TE) is the acquired data with the phase subtracted from the 1st echo

$$S(TE) = S(TE)e^{-i\theta_{S(TE_1)}} \tag{1.9}$$

and S hat is the simulated mono-exponential model signal with the phase subtracted from the 1st echo

$$\hat{S}(TE) = S_0 e^{-R_2^* TE + i\omega TE} e^{-i\omega TE_1} \tag{1.10}$$

where omega is the angular frequency derived from the total field map, R2* is estimated using closed-from solution and S0 is the extrapolated signal amplitude at TE=0.

The relative residual is a representation of the goodness of fit to the monoexponential model and this information can be brought to the weighting map using the following operations:

Step 1: Clipping

$$relativeresidualweights(relativeresidualweights > thres) = thres \tag{1.11}$$

Step 2: Weighting component from the relative residual

$$relativeresidualweights = (thres - relativeresidualweights)/thres \tag{1.12}$$

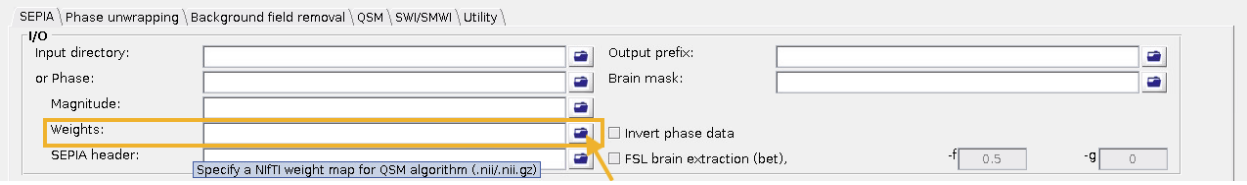
which has values between 0 (bad fit to monoexponential model) and 1 (good fit).

Step 3: Applying the weights from relative residual on previusly derived weighting map

$$weights = weights. * relativeresidualweights \tag{1.13}$$

1.16.3 Override SEPIA weighting method

If you prefer to derive your own weighting map and use it in SEPIA instead of the default weighting method of SEPIA in the One-stop processing application, you can sepcify your own NIfTI file in the I/O panel, or put the weighting map with a string 'weights' in the filename (e.g., 'data001_weights.nii.gz') along with your phase and magnitude data if you select a directory that contains SEPIA default naming structure files as the input. In this case, no weighting map will be degenerated by the software.



Warning: User-defined weighting map is not supported if you use BIDS directory as the input.

1.16.4 How does SEPIA compute the weights before v1?

Before v1, SEPIA utilises also the inverse of the field map standard deviation map as the weights, but the normalisation is different and more primitive.

Step 1

Invert the fieldmap standard deviation and remove non-value entries (i.e., NaN & inf become 0)

$$weights = \frac{1}{fieldmapSD} \tag{1.14}$$

Step 2

Normalisation of the weights. Normalisation is performed by simply using the maximum value in the data so that the range of the weights is between 0 and 1

$$weights(mask) = \frac{weights(mask)}{max(weights(mask))} \tag{1.15}$$

The potential issue with this approach is the maximum value relying on a single voxel so it could be subject to outliers and variations between dataset (e.g., different subjects or acquisition protocol can produce different maximum). As a results, there could be a global differences in terms of the magnitude of the weights between different datasets. If a dipole field inversion algorithm takes the weights for the processing, without further normalisation by the algorithms, the differences of the overall weights magnitude could impose additional regularisation differences between datasets (e.g., among subjects of the same study) even the same regularisation parameter is used across the entire study.

Warning: The medians of the weights of these two versions are in different range (before v1: less than 1 and around 0.3-0.4; v1: close to 1), meaning it may require adjusting the regularisation parameter to match regularisation effect between the two versions. Therefore, it is not recommended to mix software versions in a single study.

1.17 Lookup table of algorithm parameters

1.17.1 General algorithm parameters

algorParam.general.	Description
isBET	Logical value of performing brain extraction in SEPIA
fractional_threshold	Corresponding to '-f' option in FSL's BET
gradient_threshold	Corresponding to '-g' option in FSL's BET
isInvert	Use conjugate of the complex-valued data for processing
isRefineBrainMask	Exclude high R2* voxels from the edge of the brain mask

1.17.2 Total field recovery and phase unwrapping algorithm parameters

algorParam.unwrap.	Description
echoCombMethod	Temporal phase unwrapping method
unwrapMethod	Spatial phase unwrapping method
isEddyCorrect	Logical value of performing phase correction on data acquired with bipolar readout gradient
excludeMaskThreshold	Threshold on relative residual map to exclude unreliable voxels
excludeMethod	Method of how the unreliable voxels to be excluded
isSaveUnwrappedEcho	Logical value of saving unwrapped phase for all echoes

ROMEO (see also here)

algorParam.unwrap.	Description
offsetCorrect	corresponding to –phase-offset-correction
mask	single mask
qualitymaskThreshold	Threshold for ROMEO quality mask
useRomeoMask	Use ROMEO mask in SEPIA

1.17.3 Background field removal algorithm parameters

algorParam.bfr.	Description
refine_method	Method of residual B1 field removal
refine_order	Order of polynomial/spherical harmonic fitting
erode_radius	Voxel to be removed from mask edges after background field removal
erode_before_radius	Voxel to be removed from mask edges before background field removal
method	Method for background field removal

LBV

algorParam.bfr.	Description
tol	Iteration stopping criteria on the coarest grid
depth	No. of length scales
peel	No. of boundary layers to be peeled off

PDF

algorParam.bfr.	Description
tol	Iteration stopping criteria
iteration	Maximum no. of iterations allowed
padSize	Array size for zero padding

RESHARP

algorParam.bfr.	Description
radius	Radius of the spherical mean value operation
alpha	Regularisation parameter used in Tikhonov regularisation

SHARP

algorParam.bfr.	Description
radius	Radius of the spherical mean value operation
threshold	Threshold used in truncated SVD

VSHARP

algorParam.bfr.	Description
radius	Vector of radii of the spherical mean value operation in descending order

VSHARP (STI Suite)

algorParam.bfr.	Description
radius	Radius of the spherical mean value operation

iHARPERELLA

algorParam.bfr.	Description
iteration	Maximum no. of iterations allowed

1.17.4 QSM dipole field invevrsion algorithm parameters

algorParam.qsm.	Description
reference_tissue	Magnetic susceptibility map in ppm
method	Method for dipole field inversion

TKD

algorParam.qsm.	Description
threshold	Threshold on the dipole kernel

Closed-form solution

algor-Param.qsm.	Description
lambda	Regularisation parameter used in L2-norm regularisation
optimise	Logical value of self-determination of regularisation parameter based on an L-curve approach

NDI

algorParam.qsm.	Description
tol	Iteration stopping criteria
maxiter	Maximum no. of iterations allowed
stepSize	Gradient step size of the optimizer for each iteration

STI suite iLSQR

algorParam.qsm.	Description
threshold	Threshold on the dipole kernel
maxiter	Maximum no. of iterations allowed
tol1	Iteration stopping criteria at first level
tol2	Iteration stopping criteria at second level

iLSQR

algorParam.qsm.	Description
tol	Iteration stopping criteria
maxiter	Maximum no. of iterations allowed
lambda	Regularisation parameter used in L2-norm regularisation
optimise	Logical value of self-determination of regularisation parameter based on an L-curve approach

FANSI

algorParam.qsm.	Description
tol	Iteration stopping criteria
maxiter	Maximum no. of iterations allowed
lambda	Gradient L1 penalty, regularisation weight
mu1	Gradient consistency weight
mu2	Fidelity consistency weight
solver	Linear or non-linear algorithm for dipole inversion
constraint	TV or TGV regularisation
gradient_mode	Method for regularisation spatially variable weight
isWeakHarmonic	Logical value of using weak harmonic regularisation
beta	Harmonic constrain weight
muh	Harmonic consistency weight

Star-QSM

algorParam.qsm.	Description
padsize	Array size for zero padding

MEDI

algorParam.qsm.	Description
lambda	Regularisation parameter
wData	Method of data weighting
zeropad	Array size for zero padding
percentage	Percentage of voxels considered to be edges
isSMV	Logical value of performing spherical mean value operator
radius	Radius of the spherical mean value operation
merit	Logical value of performing modal error reduction through iterative tuning
isLambdaCSF	Logical value of performing automatic zero reference (MEDI+0)
lambdaCSF	Regularisation parameter used on CSF mask

MRI Suscep. Calc.

algorParam.qsm.	Description
solver	Methods to be used for dipole field inversion
threshold	Threshold for TKD
lambda	Regularisation parameter for Tikhonov algorithms
toleance	tiolerance level for CG solver

xQSM

algorParam.qsm.	Description
solver	Pre-trained networks for dipole field inversion

1.18 Choosing a right phase unwrapping method for your data

SEPIA provides a range of phase unwrapping methods from the supported toolboxes. In this demonstration, I will try to show the differences between two main phase unwrapping methods: (1) region growing and (2) Laplacian-based - an important processing step to reveal the (true) phase of MRI gradient echo data for QSM processing.

1.18.1 Data preparation

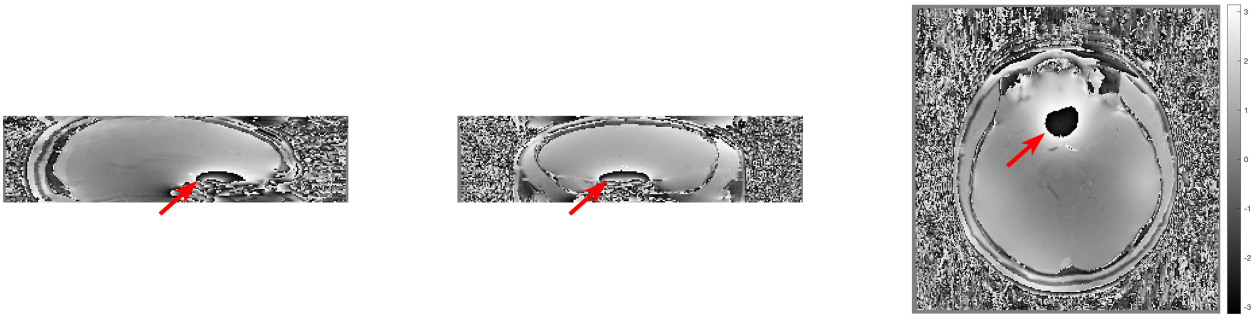
The *in vivo* test data was downloaded from Cornell MRI lab and converted to NIfTI format by dicm2nii. FSL's BET was used to extract the brain mask to improve the processing result.

1.18.2 Processing

Test 1: Unwrapping the phase with a small amount of wrapping effect

To begin with, let have a look of the phase image from the 1st echo GRE data (TE=3.6ms)

Wrapped phase 1st echo



Under a linear model assumption the phase of each voxel over time can be expressed by the following equation:

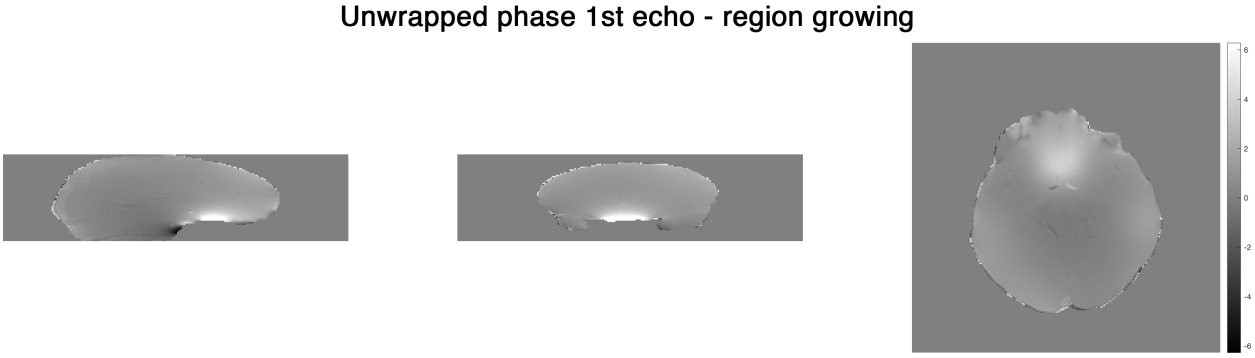
$$phase = angular frequency \times time[1]$$

However, this phase value is constrained in the interval of [-pi,pi) in the MRI phase data. Any value that is outside of this interval will be wrapped back to this interval (see red arrows). For QSM processing, it is important to unwrap the phase such that the true frequency shift inside the voxel can be computed using the above equation.

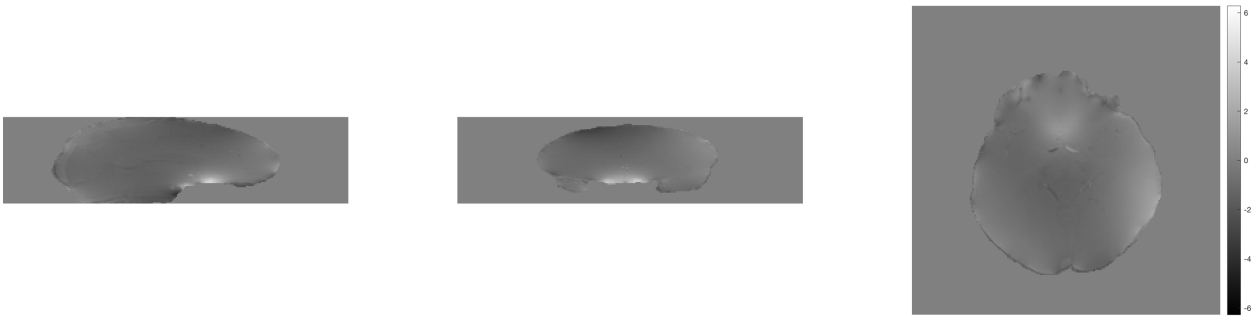
To unwrap the phase images, there are two main methods being used: (1) **region growing** and (2) **Laplacian operator**. In SEPIA, you can use the following commands to access the methods:

```
% region growing
unwrappedPhase = UnwrapPhaseMacro(wrappedPhase,matrixSize,voxelSize,'method','rg',
↪'magn',magn,'mask',mask);
% Laplacian
unwrappedPhase = UnwrapPhaseMacro(wrappedPhase,matrixSize,voxelSize,'method',
↪'laplacian');
unwrappedPhase = UnwrapPhaseMacro(wrappedPhase,matrixSize,voxelSize,'method',
↪'laplacian_stisuite');
```

Region growing



Laplacian

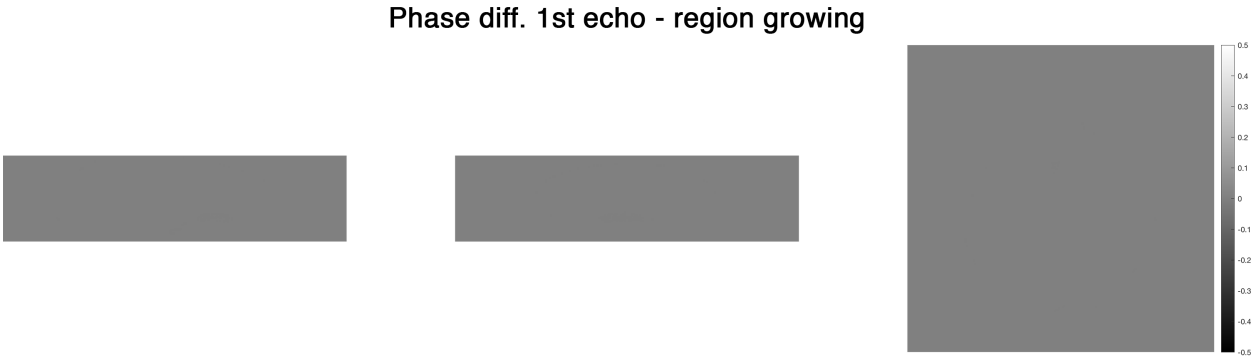


Both methods can unwrap the 1st echo phase successfully, but they show different brightness under the same dynamic range (-2pi,2pi). **Where does this difference come from?**

Let's compare the unwrapped phase with the original wrapped phase. In Matlab this can be done by: `phase_residual = angle(exp(1i*wrappedPhase) .* conj(exp(1i*unwrappedPhase)));`

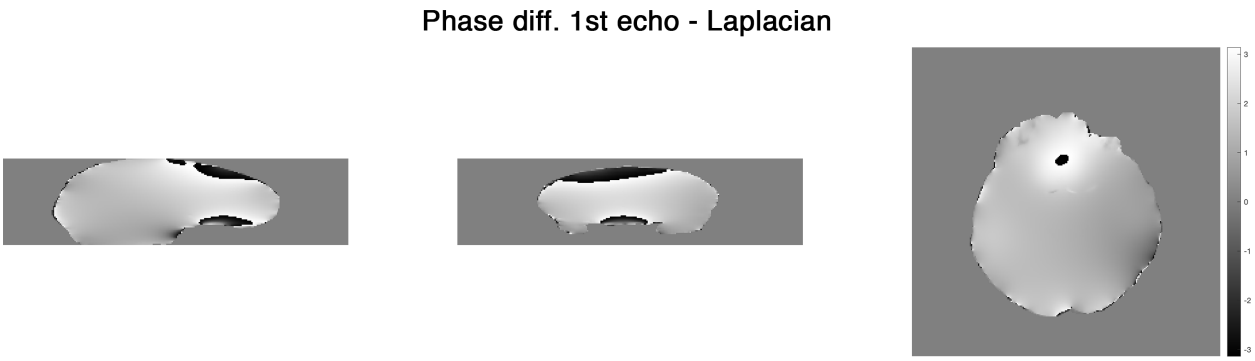
Here are the residual phase maps form

Region growing



The residual map shows 0s everywhere. This means that the unwrapped phase image is just a summation of 2xpixN (N is an integer) of the original wrapped phase map (which is unwrapped in this case).

Laplacian

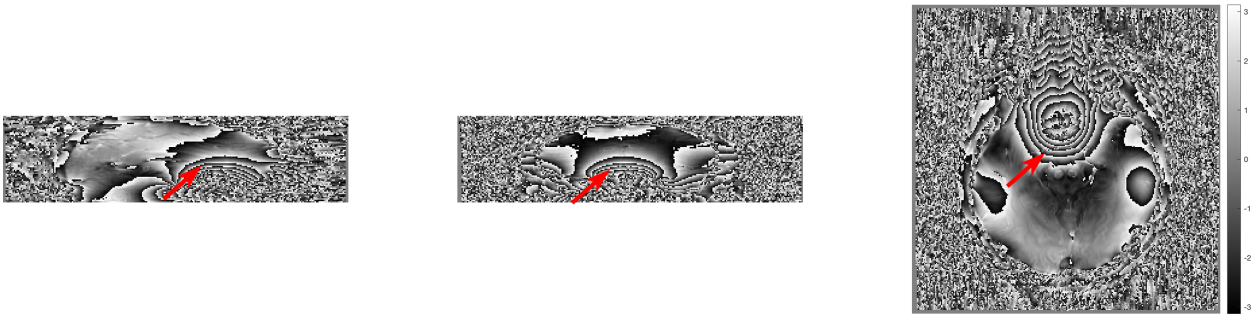


This shows that the unwrapped phase from the Laplacian method is not an exact summation of 2xpixN to the original phase. It is due to the fact that the Laplacian operator removes some of the harmonic field components from the original phase image (which is the slowly-varying field in this image). Since the harmonic field will also be removed from later on processing steps in QSM processing, Laplacian unwrapping is still a good method to unwrap the signal phase in this case (we didn't see any brain structure in the residual map except veins).

Test 2: Unwrapping the phase with a large amount of wrapping effect

From Eq.1, we can see that phase accumulated with an increase of time. Therefore when we look at the later echo(es), we can see more phase wrapping appeared and here is an example at the 8th echo (TE=45ms) of the testing data

Wrapped phase 8th echo



We can use the same unwrapping methods as we did before and look at the results again:

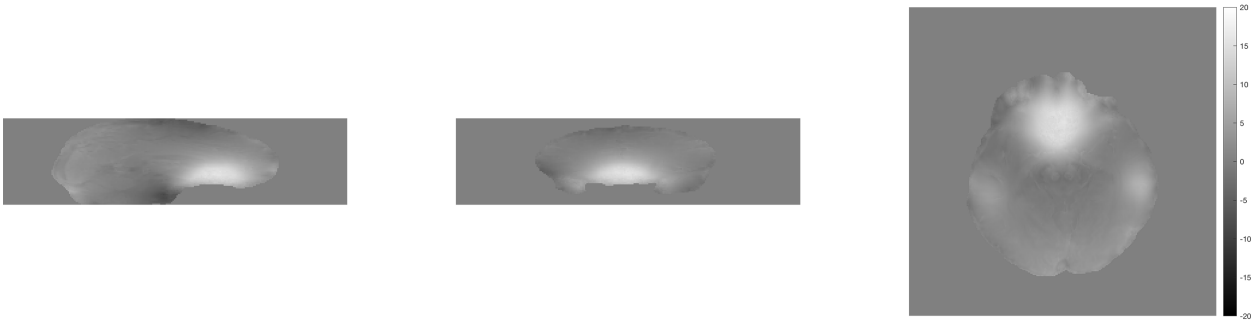
Region growing

Unwrapped phase 8th echo - region growing

The region growing method can still produce a reliable result in most of the brain regions even in the presence of a large wrapping effect. Yet, unwrapping errors can be observed in the posterior blood vessel and in the temporal lobe (see arrows). It is apparently caused by the relatively large susceptibility field together with the low SNR in these regions.

Laplacian

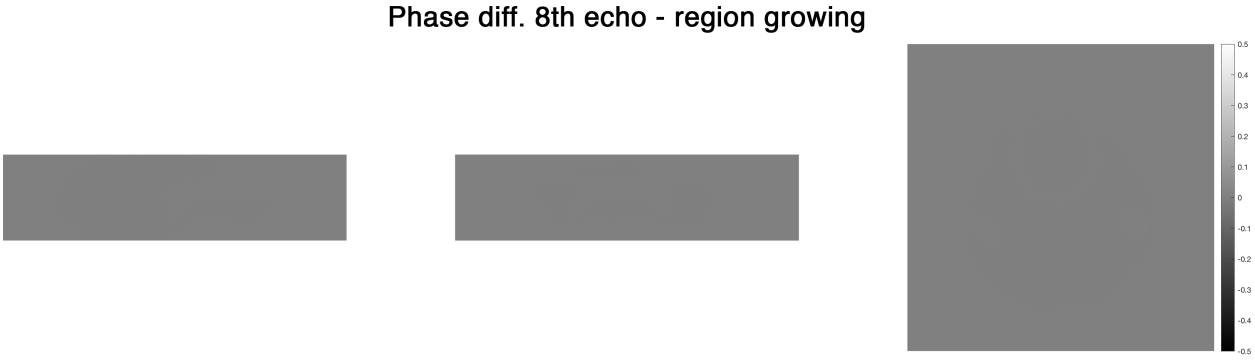
Unwrapped phase 8th - Laplacian



The unwrapped phase is smooth from the Laplacian method but is it truly better than that of the region growing method? Let's compare the unwrapped phase with the original phase to see more detail!

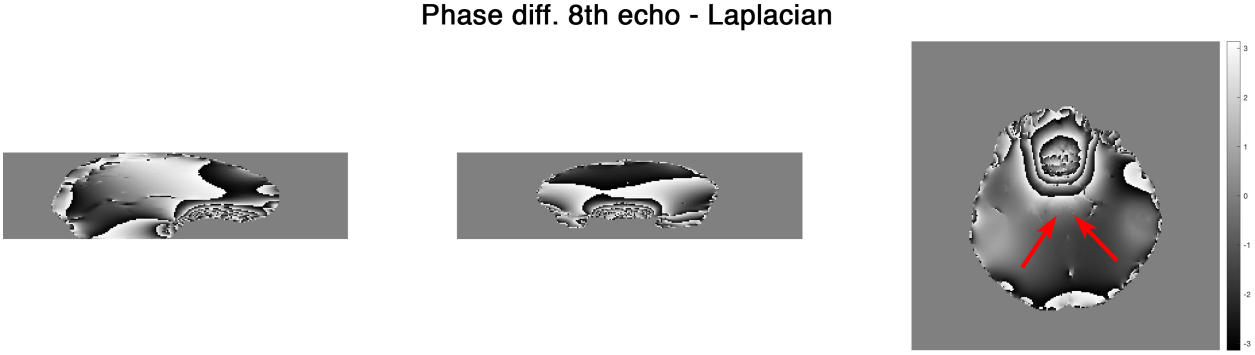
Here are the residual phase maps form

Region growing



Once again, the residual map shows 0s everywhere, suggesting that the unwrapped phase is a summation of 2xpixN of the original phase even though it didn't unwrap the phase correctly in some regions.

Laplacian



More harmonic components were removed compared to the case of 1st echo. In addition, we can start seeing brain structures (arrows indicating the red nuclei) in the residual phase which is not optimal for QSM. Nonetheless, the incorrect unwrapped phase generated by region growing method can actually create more problems in later QSM processing.

Summary

Region growing method should be used in the first attempt to unwrap the raw phase image. It is reliable for a small amount of wrapped effect and represents the true phase value (for successful unwrapping). Laplacian method is very robust to unwrap a large amount of phase wrapping by trading a small degree of accuracy in the unwrapping results.

1.19 SEPIA 101 - First QSM with SEPIA

1.19.1 Objectives

- Understanding the background of magnetic susceptibility contrast
- Gaining experience in basic QSM post-processing in SEPIA framework

Target Audience

- who is new to SEPIA
- who wants to know some basic knowledge about QSM

Estimated Time

About 1 hour

1.19.2 Prerequisite

Please install SEPIA in the Matlab environment. You can find the information about the installation procedures in *Installation/Setp up*.

1.19.3 Introduction

In this tutorial, we will go through the standard processing pipeline for quantitative susceptibility mapping (QSM), a novel contrast mechanism that uses to study tissue magnetic properties, in SEPIA framework.

Theory: QSM physics

Briefly, there are two main types of magnetic property (a.k.a magnetic susceptibility, χ) we can measure with MR:

- **Paramagnetism**: substances with paramagnetic property generate a secondary magnetic field that **enhances** the existing magnetic field generated by the MRI scanner. A typical example is iron either in blood or stored in ferritin;
- **Diamagnetism**: substances with diamagnetic property generate a secondary magnetic field that **reduces** the existing magnetic field strength, for example, myelin and calcification.

Because of the secondary magnetic field generated by (both paramagnetic and diamagnetic) tissues, the overall magnetic field experienced by water protons will no longer be the same across the whole brain. The strength of this magnetic field inhomogeneity will depend on local magnetic susceptibility sources: sources with stronger magnetic susceptibility can create a stronger inhomogeneity effect. As a result, the water protons will resonate in different frequencies across the whole brain and the frequency difference at each location is depended on the strength of the neighbouring sources. Measuring the frequency shift can, therefore, compute the magnetic susceptibility of brain tissues and reveal their cellular environment.

Note: Water protons are the main sources of MRI signal. In QSM, they act as our little magnetic field detectors to reveal the local differences of the magnetic field caused by their surrounding.

Back to *SEPIA 101 - First QSM with SEPIA*.

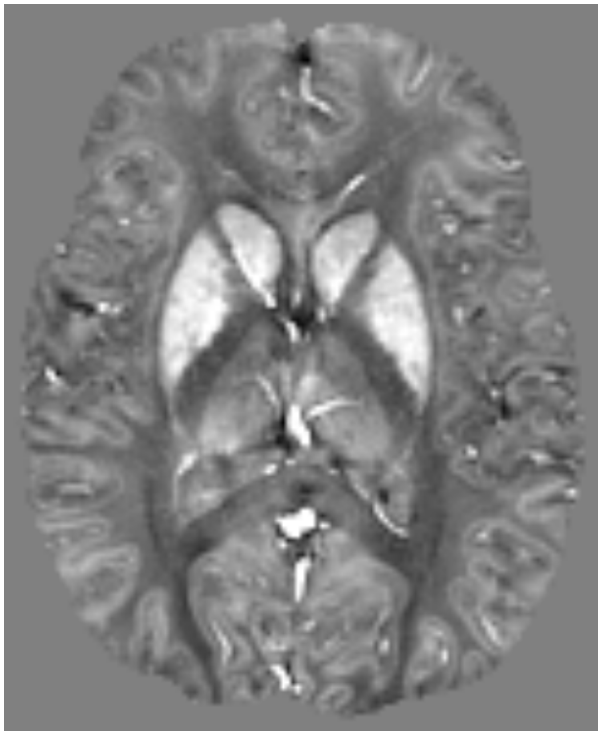


Fig. 1.1: Figure 1: A QSM map. Deep grey matter such as globus pallidus and putamen with high iron concentration is bright (positive magnetic susceptibility) while white matter, which is a myelin-rich tissue, is dark (negative magnetic susceptibility). Studies have shown that the susceptibility values in the deep grey matter are highly correlated with iron staining histology result.

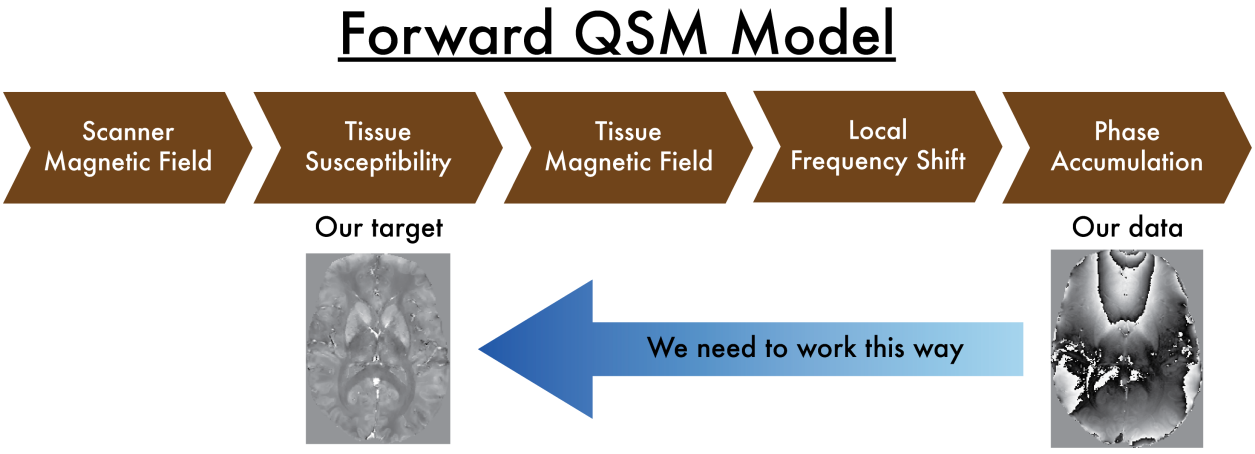


Fig. 1.2: Figure 1: QSM theory. A magnetic source generates a secondary magnetic field inside the MRI scanner, which will eventually alter the signal phase we measured. Decoding the phase data allows us to detect the molecular environment of the brain tissues.

1.19.4 Exercises

Let's begin with the following exercises that will take you from the phase data to the susceptibility maps!

Exercise 1

Objectives

- Understanding the data required for QSM
- Understanding why we need to correct the phase data before mapping the magnetic susceptibility

Data Required

Data	Description
mag.nii.gz	magnitude of complex-valued multi-echo GRE data with 4 dimenions, [spatial_x,spatial_y,num_of_slices,num_of_echoes]
phase.nii.gz	phase of complex-valued multi-echo GRE data with 4 dimenions, [spatial_x,spatial_y,num_of_slices,num_of_echoes]
mask.nii.gz	3D signal mask
sepia_header.mat	contains important information such as the echo times (TE) and magnetic field strength (in Tesla), and orientation of the acquisition regarding the physical coordinates of the scanner. These are important to compute the magnetic susceptibility with the correct units and ensure the physical model is correct.

Estimated time

About 15 min.

Understanding multi-echo GRE data

To compute a magnetic susceptibility map, multi-echo gradient-echo (mGRE) images are usually used because it can provide phase images related to the induced tissue magnetic field.

Theory: MR Phase

As mentioned in the Introduction, water protons resonate at different frequencies in the brain because of the tissue magnetic susceptibility. The frequency difference in brain tissues, f, can be detected as the difference in phase accumulation, θ, over time, t ($\theta(t) = f \times t$). Therefore, the phase measurement of the MRI signal allows us to directly map the effect of the magnetic susceptibility of brain tissues.

It should be noted that the phase can only be measured in the range of [-180, 180] degrees (or $[-\pi, \pi]$ in radian).

Back to *Exercise 1*.

Go to the data directory `~/sepia_tutorial/sepia101_data/` and have a look.

You should be able to see three compressed NIfTI images (.nii.gz) and a SEPIA header file (.mat) in the directory. The NIFTI data is generated from a phantom dataset used in QSM Challenegs 2.0. It simulates data acquired with a 3D mGRE sequence at 7T. Both magnitude and phase images are 4D, with the first 3 dimensions containing spatial information (i.e. image of a brain) and **echo time in the 4th dimension**.

Note: You can check *Installation/Setp up* to see what information is needed in the SEPIA header file and how to generate this file from standard DICOM/NIfTI conversion tools.

SEPIA is a QSM porcessing pipeline tool developed in Matlab. To use SEPIA, please first start a Matlab session. Once Matlab is open, go to the tutorial data directory (`~/sepia_tutorial/sepia101_data/`) in Matlab environment.

We will first give a brief explaination about our tutorial data:

Magnitude images

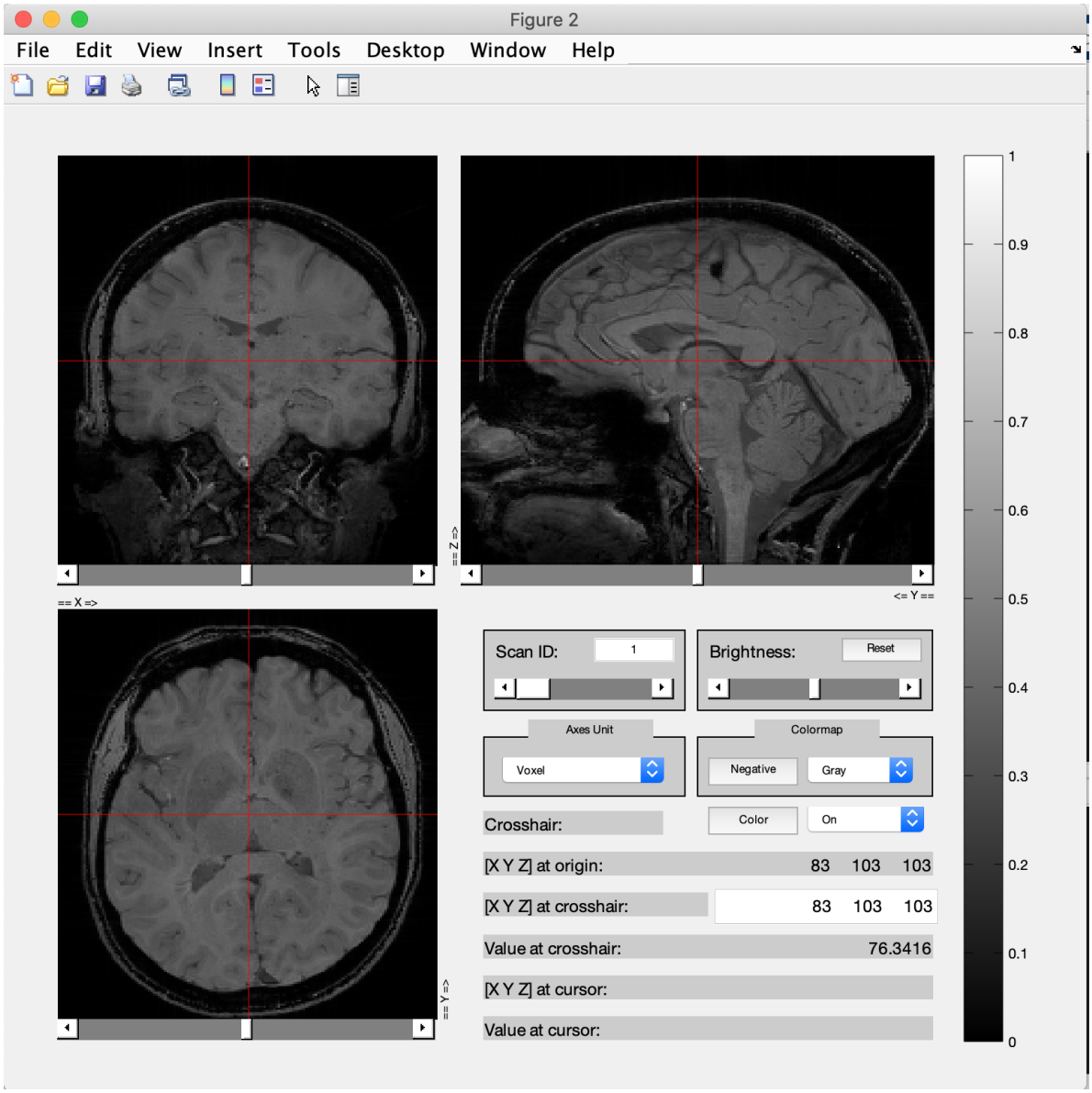
1. Take a look of the magnitude images. You can do it in Matlab using the following commands:

   `sepia_addpath`, and then

   `view_nii(load_nii('mag.nii.gz'))`

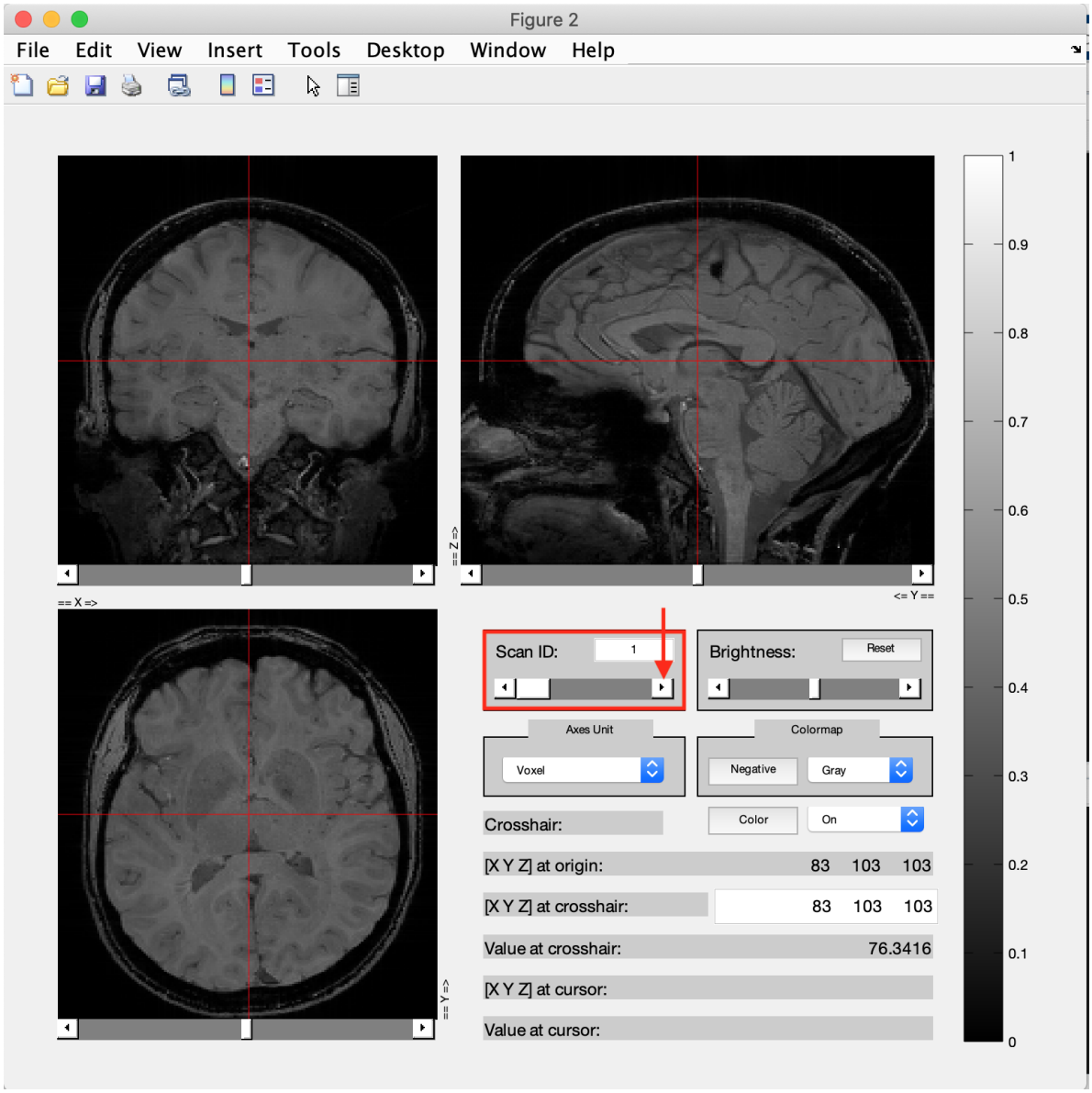
   The first command will load all non-external functions in SEPIA, including the NIfTI support for Matlab from Jimmy Shen.

   The second command will call a NIfTi viewer to display the magnitude image. The display contrast is automatically adjusted in the viewer.



Tip: You can also do this with your favourite NIfTI viewer that allows visualising 4D data (e.g. `FSLeyes`). If you use `FSLeyes` then adjust the display window to 'Min 0' and 'Max 150'.

2. By default, the 1st echo data is displayed in the viewer. Use the slider in 'Scan ID' to see how the images change in time (time between echoes = 8 ms).



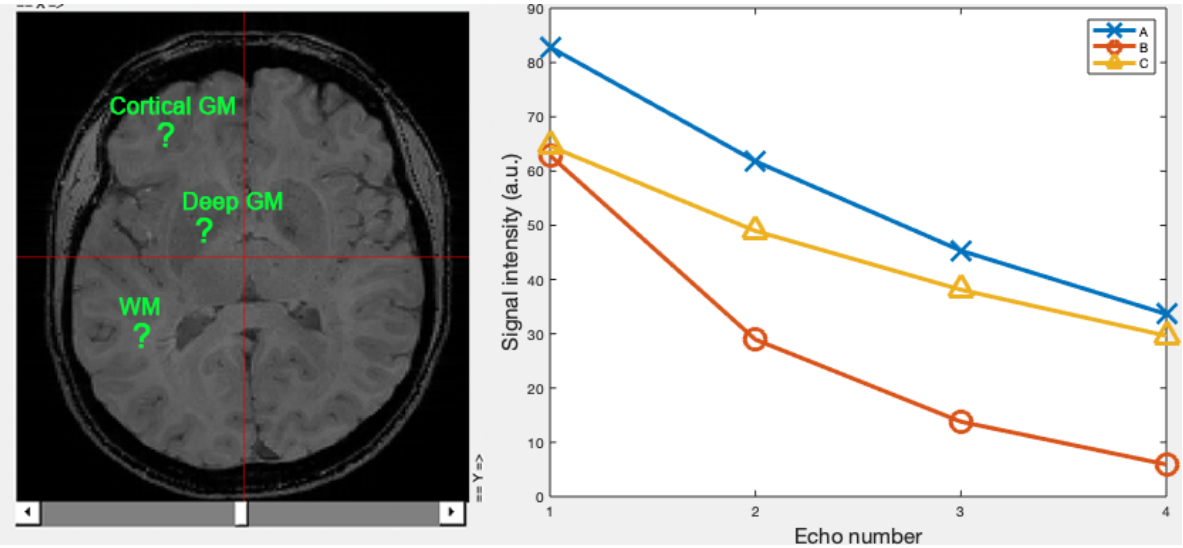
You should be able to see the overall brightness of the images becomes dimmer in later echoes. It is because water protons was losing phase coherence (dephasing) over time. Roughly speaking, the greater is the dephasing phenomenon, the weaker is the signal. One of the causes of dephasing effect in GRE data is related to the field inhomogeneity caused by the local magnetic field from tissues. Tissues that have a lot of strong magnetic sources (e.g. iron stored in ferritin) will decay much faster than those with fewer/weaker sources.

Tip: If you are using `FSLeyes`, you can also click the the movie button  to see how the brain contrast changes with respect to the echo time in action.

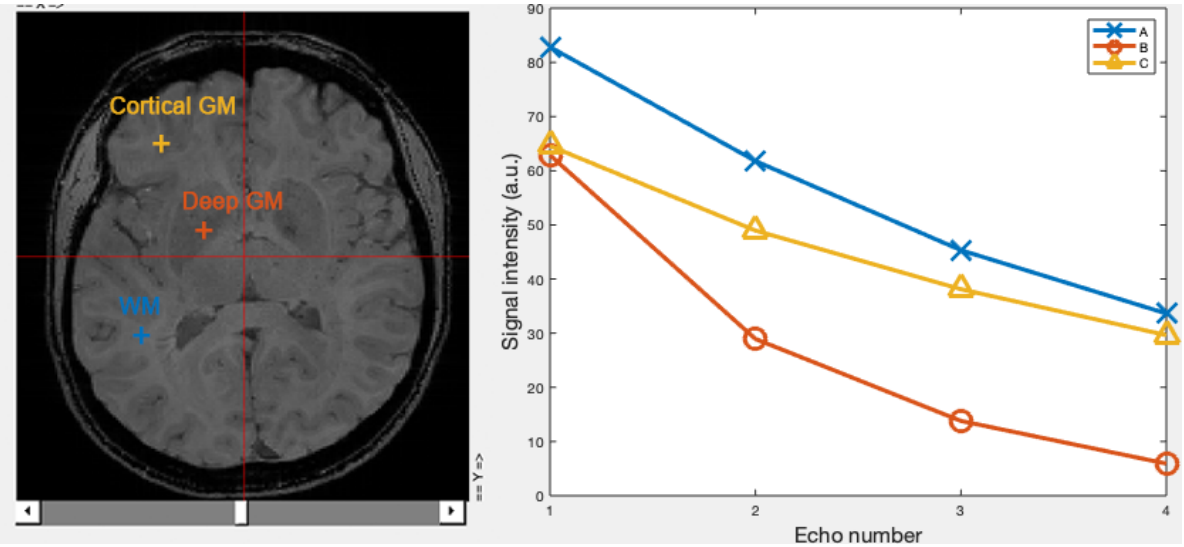
---

3. Signals change in time on three locations on slice #103 (A, B, C) were plotted randomly representing 3 different types of tissue (deep grey matter, corical gray matter and white matter) as shown below.

   Question 1: Can you identify the locations A, B and C to the corresponding type of tissue? Observe the change of intensity differences of the tissues over time in the image viewer



Answer of Question 1



White mater (WM) has the strongest signal intensity at the 1st echo corresponding to curve A. Cortical grey matter (cortical GM) and deep grey matter (deep GM) have similar signal intensity at the 1st echo, yet the signal intensity of deep GM at the later echoes are the weakest among all locations so it corresponds to curve B and cortical GM corresponds to curve C.

The plot of signal variation over time in the right actually tells us some information about magnetic susceptibility. It is well-known that deep grey matter such that globus pallidus has high iron content. Those irons, stored in ferritin, are paramagnetic substances that can create a relatively strong (local) magnetic field, in turns speeding up the dephasing effect of water protons in the tissue and making the darker appearance in the images. Given the contrast observed from the magnitude data can also be magnetic susceptibility related, though QSM mainly works with phase images, magnitude images can be very useful because they share similar (but not always the same!) contrasts, therefore, have the potential to improve the quality of QSM maps.

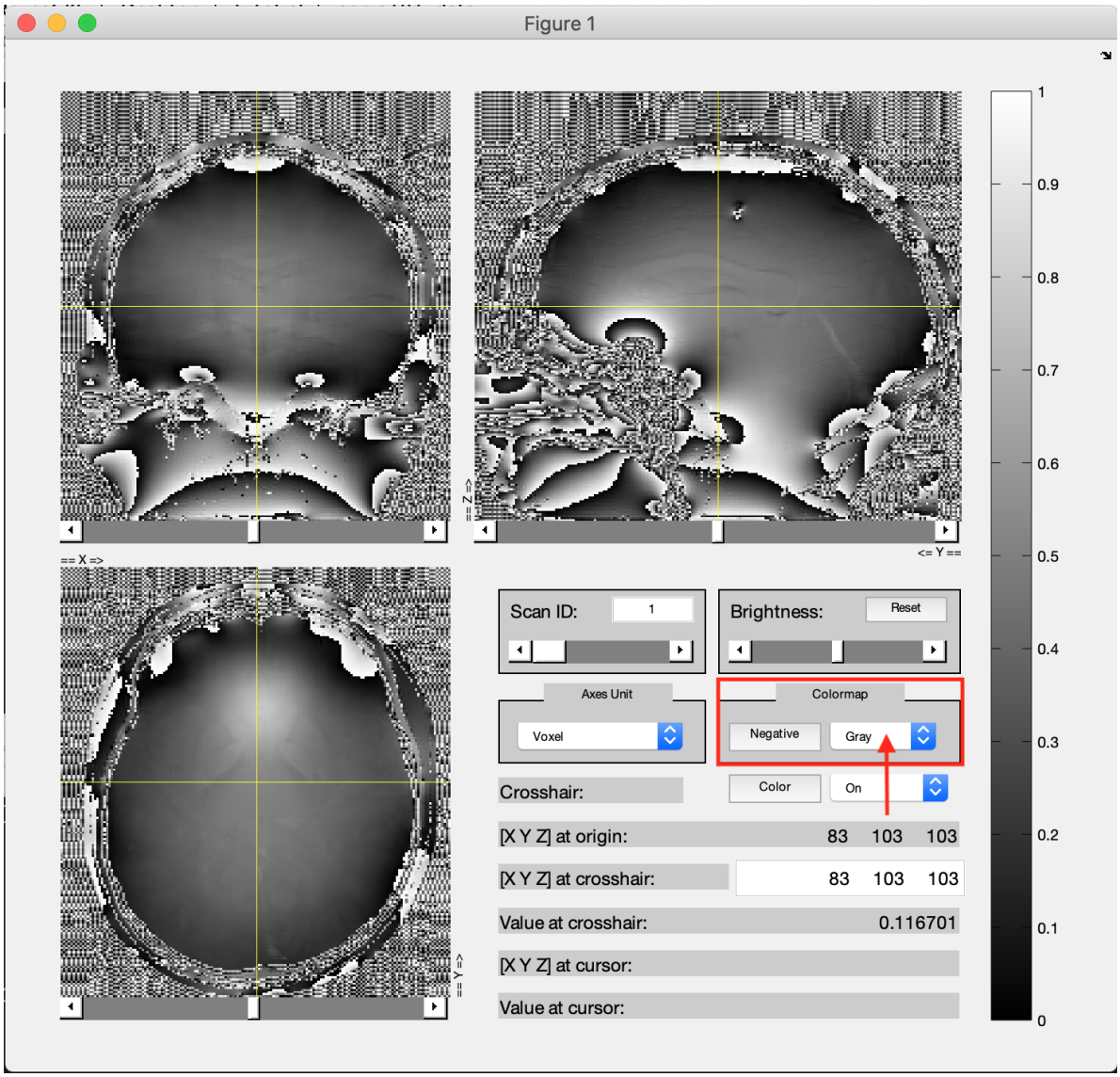
Back to *Exercise 1*.

Phase images

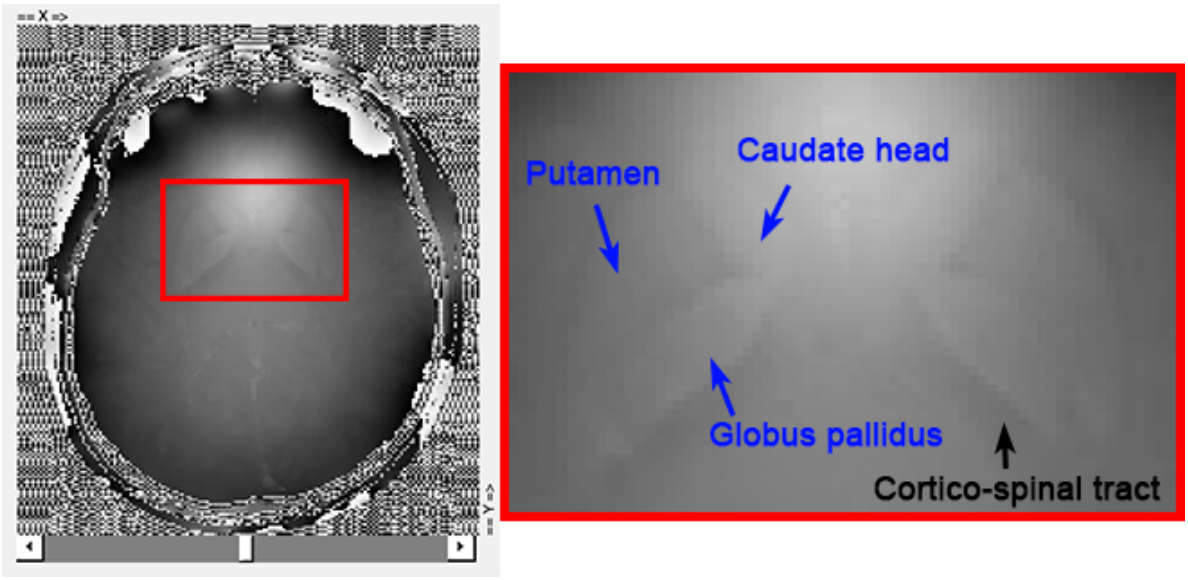
1. Now close the viewer. Open the phase images and change the colormap to 'Gray':

   ```
   view_nii(load_nii('phase.nii.gz'))
   ```

   The phase images look quite different from the magnitude images.



   Even in the 1st echo phase, it is still possible to (vaguely) identify some such as globus pallidus, putamen, head of caudate nucleus and corticospinal tracts on slice #103. The structures also become clearer in later echoes (you can use the slider in 'Scan ID' as in the magnitude image exercise above).

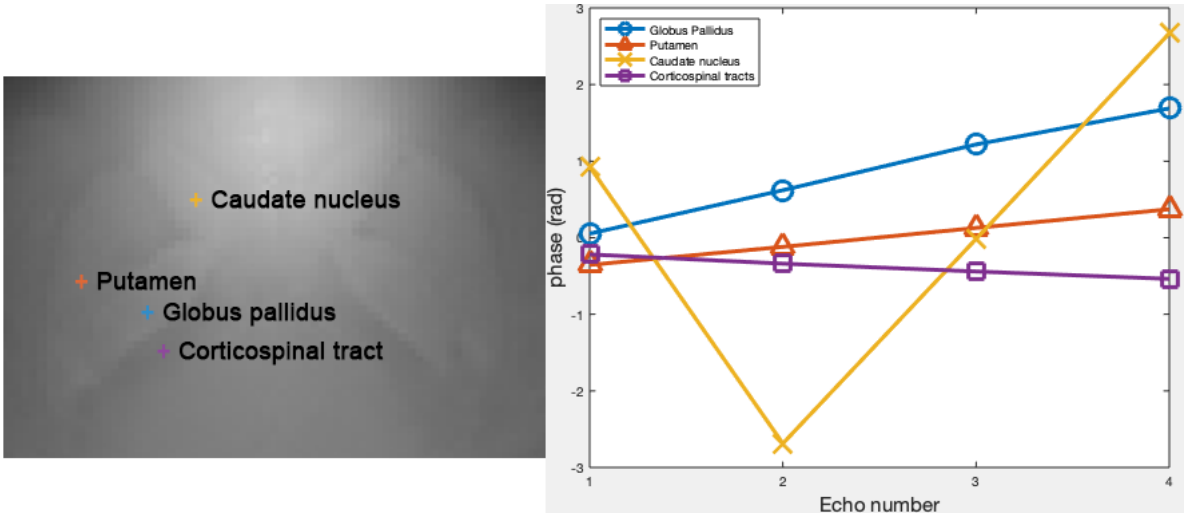


2. In QSM, the phase of the GRE signal is assumed to be directly proportional to the frequency induced by local susceptibility sources in time, i.e.:

$$phase = frequency \times time \tag{1.16}$$

In other words, we should observe the phase contrasts become greater in the later echoes (i.e. bright brighter and dark darker in the next echo).

This is true when we focus on some structures mentioned in Step 1 (e.g. globus pallidus, putamen and corticospinal tracts) but not always true for the caudate nucleus. Here is the plot of the phase over time in these structures:



In the figure, you can see the phase development over time of the globus pallidus, putamen and corticospinal tracts are roughly linear but it is not the case with the head of caudate nucleus from which you can see the phase value first dropped from first echo to the second echo and then increased from the second echo to the last echo. The decrease of the phase value at the 2nd echo can be seen as the growing zebra-line pattern coming from the inferior frontal lobe toward the centre of the brain (location [75,128,103]). The pattern, if we watch it closely, is spreading not only from the edge to the centre of the brain but also as with the increase of echo time.

Question 2: Which of the following is the cause of the problem?

A. Somebody messed up the acquisition

Unfortunately, all phase images of mGRE data have a similar problem. So it is unlikely that everyone messed their acquisition up.

Back to *Exercise 1*.

B. The subject moved during the scan

Subject motion does induce changes in the phase images but certainly not responsible for the phase changes across echoes (each echo is separated by 8 ms only!). Therefore, it is not the reason we have this observation.

Back to *Exercise 1*.

C. The phase is bounded to certain values

Yes, this is the correct answer! There is nothing wrong about the signal phase generated by the tissues. However, when we sampled the signal, the phase values are constrained to the range $-\pi \leq phase < \pi$. When the true phase is outside of this range, a π will be added or subtracted to the true phase such that the *apparent* phase we observed from the data will still be inside the range.

Note: Think of an analogue clock with one hand only representing the hour. Every 12 hours the hand starts a new cycle again. If you look at the clock 13 hours after the first watch, it apparently advanced 1 hour only but the actual time difference between the two observations is 13 hours.

Back to *Exercise 1*.

D. Fast switching gradient introduced extra phase

The instability of the fast switching gradient could induce extra phase offsets to the data. However, it is not the reason we have this observation in the phase images.

Back to *Exercise 1*.

E. I don't know. I'm here to learn some fMRI analysis so just show me the answer

There is nothing wrong about the signal phase generated by the tissues. However, when we sampled the signal, the phase values are constrained to the range $-\pi \leq phase < \pi$. When the true phase is outside of this range, a π will be added or subtracted to the true phase such that the *apparent* phase we observed from the data will still be inside the range.

Note: Think of an analogue clock with one hand only representing the hour. Every 12 hours the hand starts a new cycle again. If you look at the clock 13 hours after the first watch, it apparently advanced 1 hour only but the actual time difference between the two observations is 13 hours.

Back to *Exercise 1*.

In order to estimate the frequency shift correctly using Eq. (1.16), this phase problem has to be addressed which is called phase unwrapping.

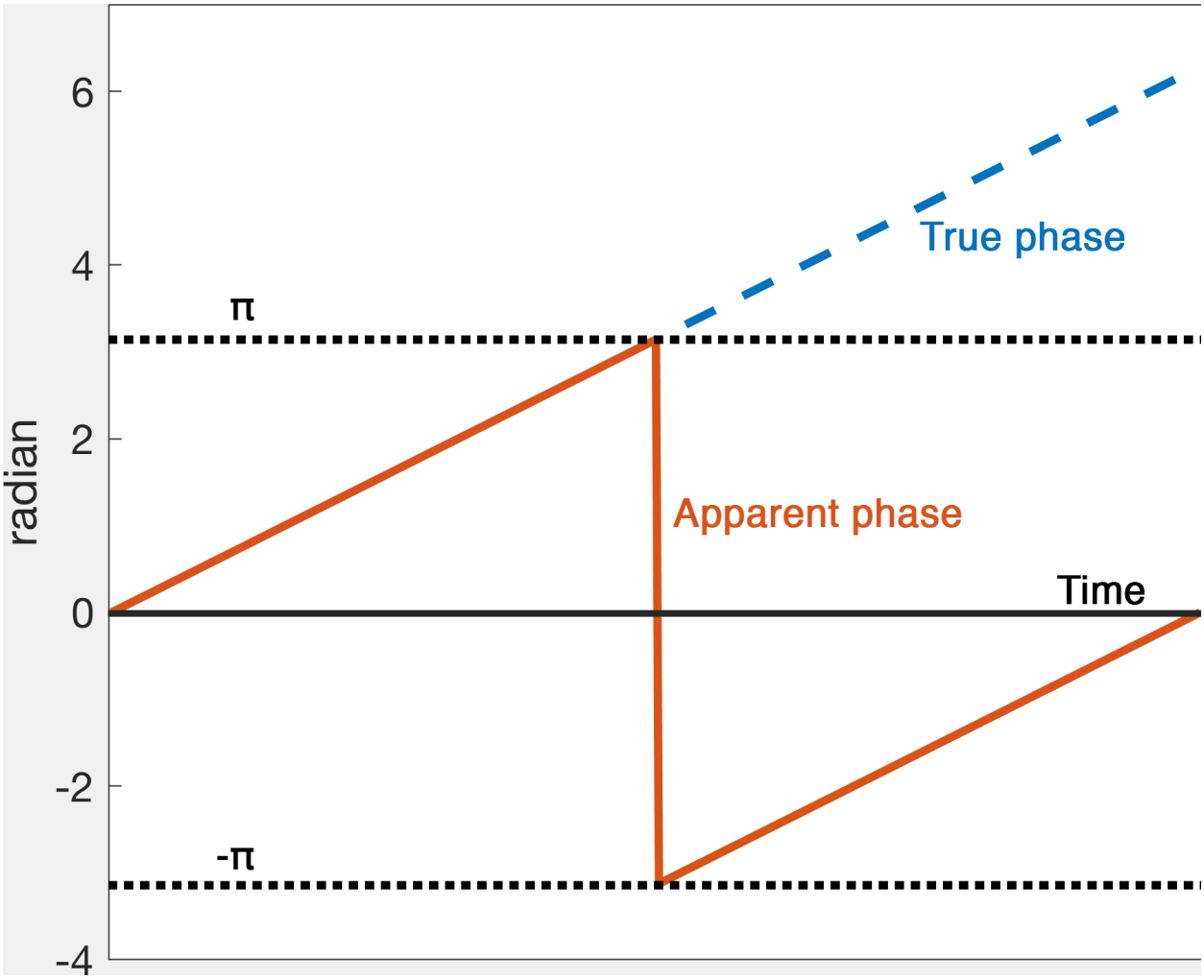


Fig. 1.3: Figure 1: An illustration of phase wrapping. The actual phase developed outside the range $-\pi \leq phase < \pi$ over time but our measurement will only show the apparent phase.

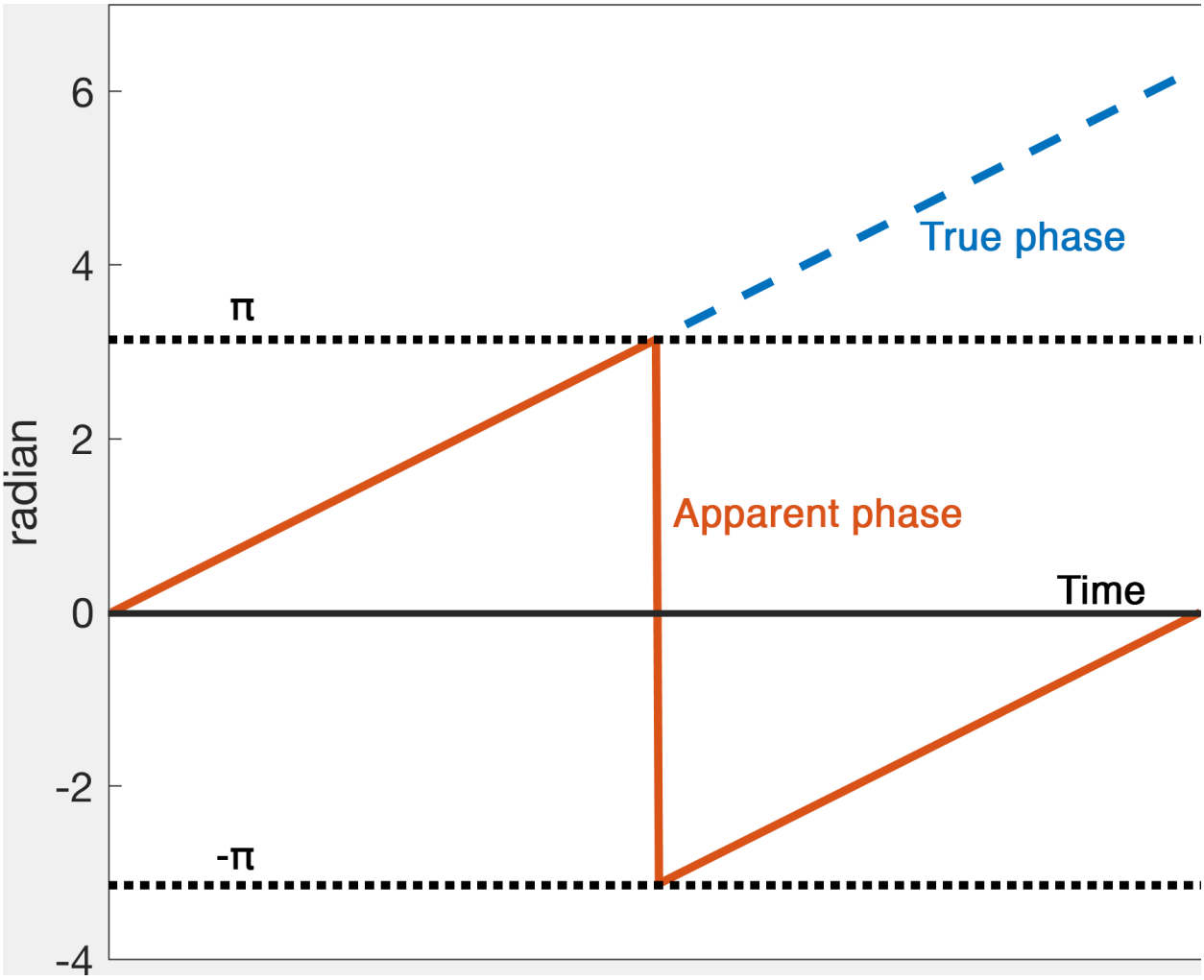


Fig. 1.4: Figure 1: An illustration of phase wrapping. The actual phase developed outside the range $-\pi \leq phase < \pi$ over time but our measurement will only show the apparent phase.

To unwrap the phase and to map back to the correct values, SEPIA provides several algorithms to do the job, and this is what we are going to do in the next exercise.

You can close all the image viewer(s) now.

Proceed to *Exercise 2*.

Exercise 2

Objectives

- Gaining experience in using SEPIA
- Understanding how to perform phase unwrapping

Data Required

Data	Description
mag.nii.gz	magnitude of complex-valued multi-echo GRE data with 4 dimenions, [spatial_x,spatial_y,num_of_slices,num_of_echoes]
phase.nii.gz	phase of complex-valued multi-echo GRE data with 4 dimenions, [spatial_x,spatial_y,num_of_slices,num_of_echoes]
mask.nii.gz	3D signal mask
sepia_header	contains important information such as the echo times (TE) and magnetic field strength (in Tesla), and orientation of the acquisition regarding the physical coordinates of the scanner. These are important to compute the magnetic susceptibility with the correct units and ensure the physical model is correct.

Estimated time

About 20 min.

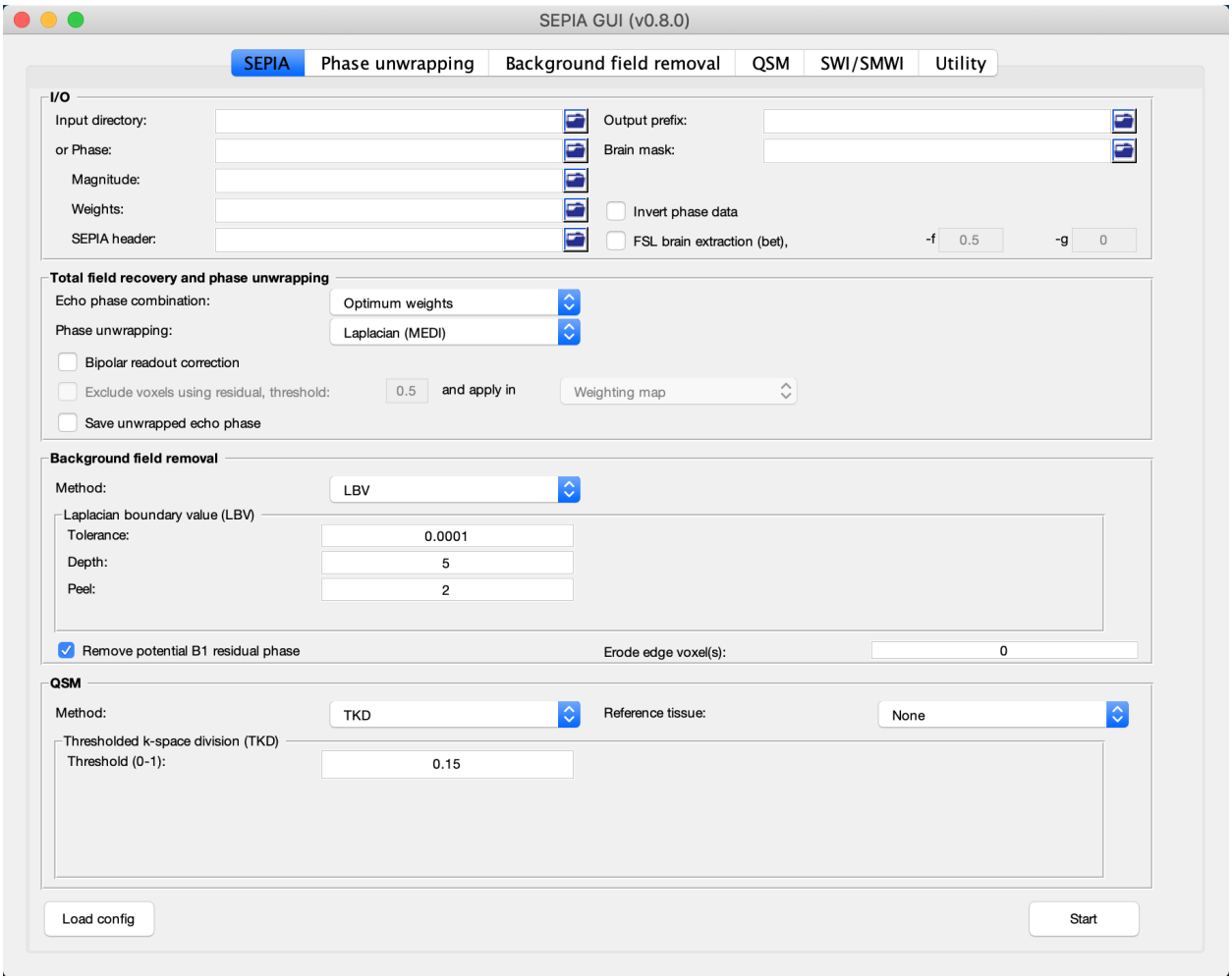
SEPIA

Now, go the data directory in the Matlab's command window and start sepia:

```
cd ~/sepia_tutorial/sepia101_data/
```

```
sepia
```

A graphical user interface (GUI) should appear right away.



There are several tabs in SEPIS corresponding to usage of SEPIA in various tasks. The first tab in SEPIA provides a one-step application to process QSM from the raw phase data to a magnetic susceptibility map. Alternatively, we can break down the processing pipeline into several steps and SEPIA also supports this approach.

In the following exercises, we will go for the second approach in this tutorial such that we can explain the QSM processing step by step.

Phase Unwrapping and Total Field Computation

From exercise 1, we understand the raw phase GRE data is affected by the phase wrapping issue which stops us from computing the frequency shift correctly using the phase data.

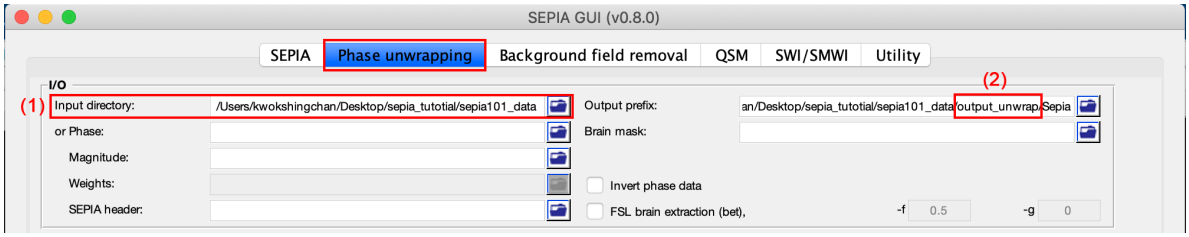
To correct the wrapped phase in the raw images, go the **Phase unwrapping** tab (next to **SEPIA** tab).

You will see two panels under the tab: the **I/O** panel is for specifying data input and output and the **Total field recovery and phase unwrapping** panel is for selecting phase unwrapping and true phase estimation algorithms.

Tip: SEPIA supports two data input routines: (1) If your data follows the SEPIA naming structure, you can select the directory containing all the input data as your input in the first row of **I/O** panel. (2) Alternatively, you can specify the input files separately by following the instruction of the second row of the **I/O** panel.

In the **I/O** panel:

1. Select the **Input directory**: *~/sepia_tutorial/sepia101_data/*
2. Change the **Output prefix** to: *~/sepia_tutorial/sepia101_data/output_unwrap/Sepia*



In the **Total field recovery and phase unwrapping** panel:

1. Keep the **Echo phase combination** method as 'Optimum weights'.

   This option is to determine how the phase information in time will be combined for multi-echo data. Here we decided to combined the phase information based on SNR weighting.

2. Change the **Phase unwrapping** method to 'SEGUE'.

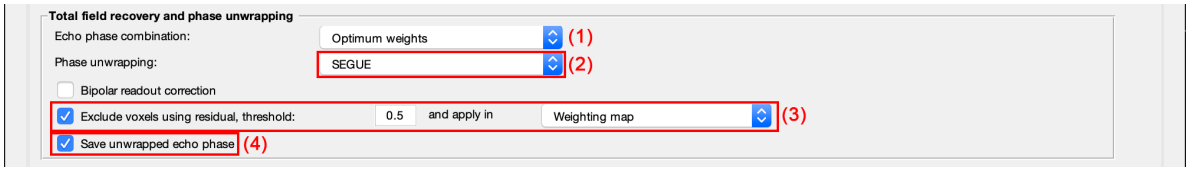
   This option is to determine the algorithm to spatially unwrap the phase. 'SEGUE' is a region growing based method.

3. Check **Exlcude voxels using residual, threshold:** option.

   This optino allows us to create a new weighting map based on how closely the signal evolves like a simple linear model.

4. Check **Save unwrapped echo phase** option.

   This option allows us to save the unwrapped phase for each echo.



Then click the **Start** button at the bottom of the GUI.

You should now see some messages regarding the general information of your input data and the overview of the selected method(s) displaying on the Matlab's command window. Once the process finishes (a few minutes), you will see the message meaning the processing is finished.

'*Processing pipeline is completed!*'.

Tip: All the output messages of SEPIA will be displayed on the Matlab command window. Make sure you check the command window before clicking the **Start** button again!

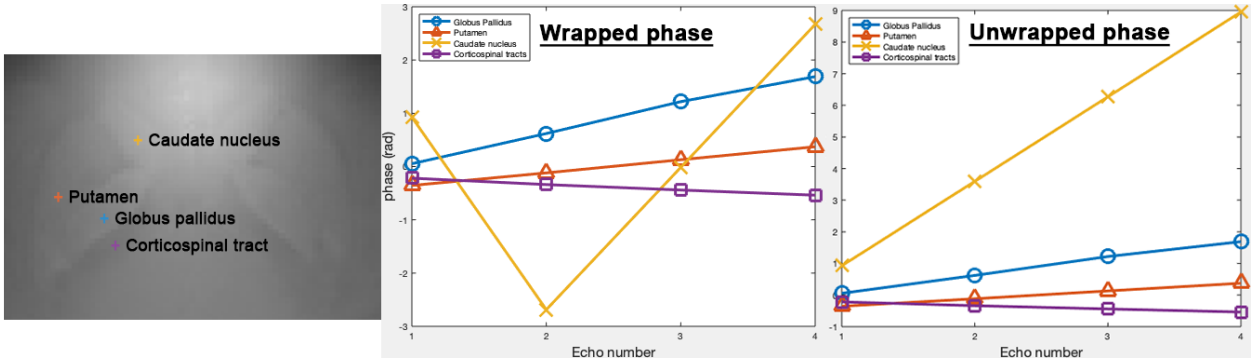
Once the process is finished, you should be able to see the following output in the output directory (*~/sepia_tutorial/sepia101_data/output_unwrap/*)

Output data	Description
sepia_config.m	Automatic generated script by the GUI of SEPIA containing all user specified parameters
run_sepia.log	Event log file of the Matlab's command window output
Sepia_total-field.nii.gz	Unwrapped total frequency shift in Hz
Sepia_relative-residual.nii.gz	Relative residual derived using mono-exponential model with a single frequency shift (if voxel exclusion is selected)
Sepia_mask-reliable.nii.gz	Derived from thresholding relative-residual map using user-defined value
Sepia_noise-sd.nii.gz	Estimated standard deviation of noise in the phase data
Sepia_unwrapped-phase.nii.gz	Unwrapped phase data in radian
Sepia_weights.nii.gz	SNR-weighted image derived from standard deviation of noise in phase data

Let's have a look of the unwrapped phase first (*Sepia_unwrapped-phase.nii.gz*), assuming you are still in the data directory (*~/sepia_tutorial/sepia101_data/*) in Matlab.

```
view_nii(load_nii('output_unwrap/Sepia_unwrapped-phase.nii.gz'))
```

Try to see the phase of each echoes using the slider of 'Scan ID'. Now you shall see that all the zebra-line pattern and phase jumps are gone in the later echo images. If we plot the phase of the brain structure in Exercise 1, the phase of the caudate nucleus also evolves linearly after phase unwrapping.



With the correctly unwrapped phase, we can compute the total frequency shift (*Sepia_total-field.nii.gz*) in the tissue from the phase using the following equation:

$$frequency = \frac{phase}{time} \tag{1.17}$$

Open the total frequency shift (or field) map and see how it looks like:

```
view_nii(load_nii('output_unwrap/Sepia_total-field.nii.gz'))
```

In the total field map, we can vaguely see some brain structures but they seems to be hidding behind something. It is because the total field map has contributions from not only the tissues but also background sources such as air/tissue interface which have strong magnetic susceptibility creating magnetic fields that can affect the whole brain. To be able to compute tissue magnetic susceptibility, the field effect from background (non-tissue) sources has to be removed from the total field.

You can close the image viewer now.

Proceed to *Exercise 3*.

Back to *Exercise 1*.

Exercise 3

Objectives

- Understanding why we need to remove the background magnetic field contributions before QSM
- Fine-tuning method parameters to improve the background field removal results

Data Required

Data	Description
Sepia_totalfield.nii.gz	Unwrapped total frequency shift in Hz, in *~/sepia_tutorial/sepia_data/output_unwrap/*
mask.nii.gz	3D signal mask, in *~/sepia_tutorial/sepia_data/*
sepia_header.mat	contains important information such as the echo times (TE) and magnetic field strength (in Tesla), and orientation of the acquisition regarding the physical coordinates of the scanner. These are important to compute the magnetic susceptibility with the correct units and ensure the physical model is correct , in *~/sepia_tutorial/sepia_data/*

Estimated time

About 10 min.

Background Field Removal

The total frequency map we obtained from the last exercise contains magnetic fields generated by not only the brain tissue but also the scanner hardware imperfection and fields generated at air/tissue interfaces. Therefore, we have to remove the non-tissue field contributions, or background field, before computing the susceptibility map.

One more step before computing QSM but why?

In the last exercise, it seems like the brain structures are hidding behind something. Why are these tissue contrasts 'hidden' in our images?

The data contains not only the phase generated by the brain tissues but also by the scanner hardware imperfection and fields generated at air/tissue interfaces such as sinuses and ear canals.

These non-tissue fields, or so-called background fields, can be one or two order(s) of magnitude stronger than the tissue fields and affects the global brain tissues.

Since we are only interested in the magnetic fields generated by the brain tissues, we have to remove these background fields before computing the QSM map. However, the background fields and tissue fields co-exist across the whole brain.

Luckily, the background fields have different mathematical properties and researchers have successfully developed various algorithms to separate the tissue fields from the background fields.

Back to *Exercise 3*.

SEPIA provides 7 methods to remove the background magnetic fields. Today we will use the so-called Sophisticated Harmonic Artifact Reduction for Phase (SHARP) algorithm to do this job.

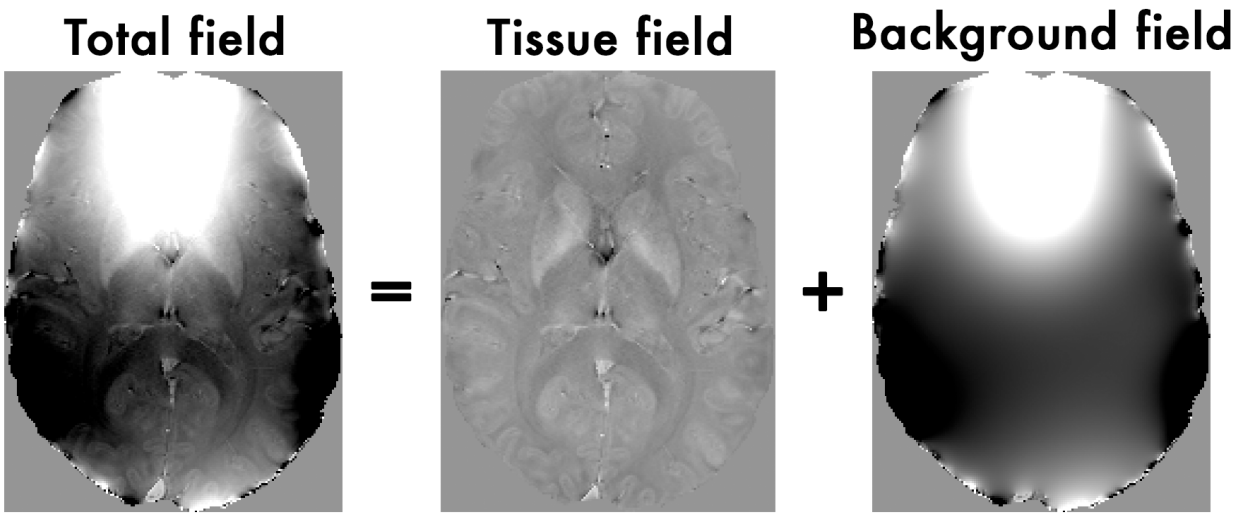


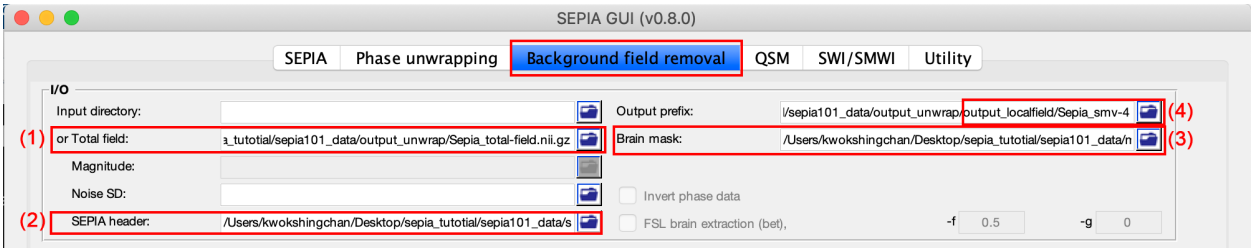
Fig. 1.5: Figure 1: The total field we obtained from the last exercise is the summation of tissue and background magnetic fields. In order to compute the magnetic susceptibility of the brain tissue correctly, the background field contributions have to be removed before mapping the tissue susceptibilities.

Exercise 3.1

Go to the **Background field removal** tab. You will see two panels as in the **Phase unwrapping** tab.

In the **I/O** panel, specify the required files by using the  buttons:

1. **or Total field**: *Sepia_total-field.nii.gz* (in *~/sepia_tutorial/sepia_data/output_unwrap/*),
2. **SEPIA Header**: *Sepia_header.mat* (in *~/sepia_tutorial/sepia_data/*),
3. **Brain mask** *mask.nii.gz* (in *~/sepia_tutorial/sepia_data/*),
4. Change the **Output prefix** to: *~/sepia_tutorial/sepia101_data/output_unwrap/output_localfield/Sepia_smv-4*



Second, in the **Background field removal** panel, change the **Method** to 'SHARP'. You can have two parameters to adjust.

- 'SMV radius (voxel)': radius of a spherical mean value (SMV) kernel, in number of voxels
- 'Threshold': threshold used in Truncated SVD.

1. Press the **Start** button.

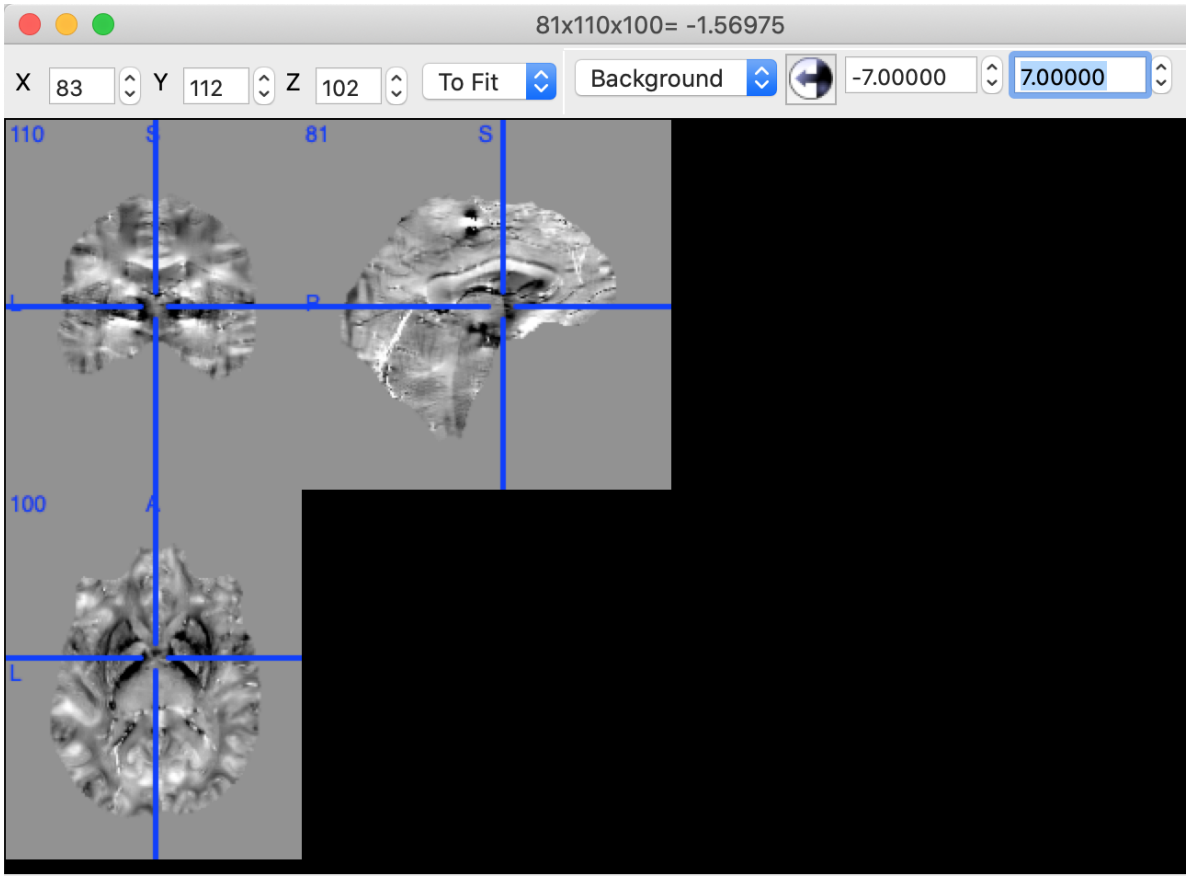
   Again, when the process is finished, you will see the message: '*Processing pipeline is completed!*'.

2. Once the process is finished, you should be able to see the following output in the output directory (*~/sepia_tutorial/sepia101_data/output_unwrap/output_localfield/*)

Output data	Description
sepia_config.m	Automatic generated script by the GUI of SEPIA containing all user specified parameters
run_sepia.log	Event log file of the Matlab's command window output
Sepia_smv-4_local-field.nii.gz	Local (tissue) field map in Hz
Sepia_smv-4_mask-qsm.nii.gz	Signal mask for QSM step

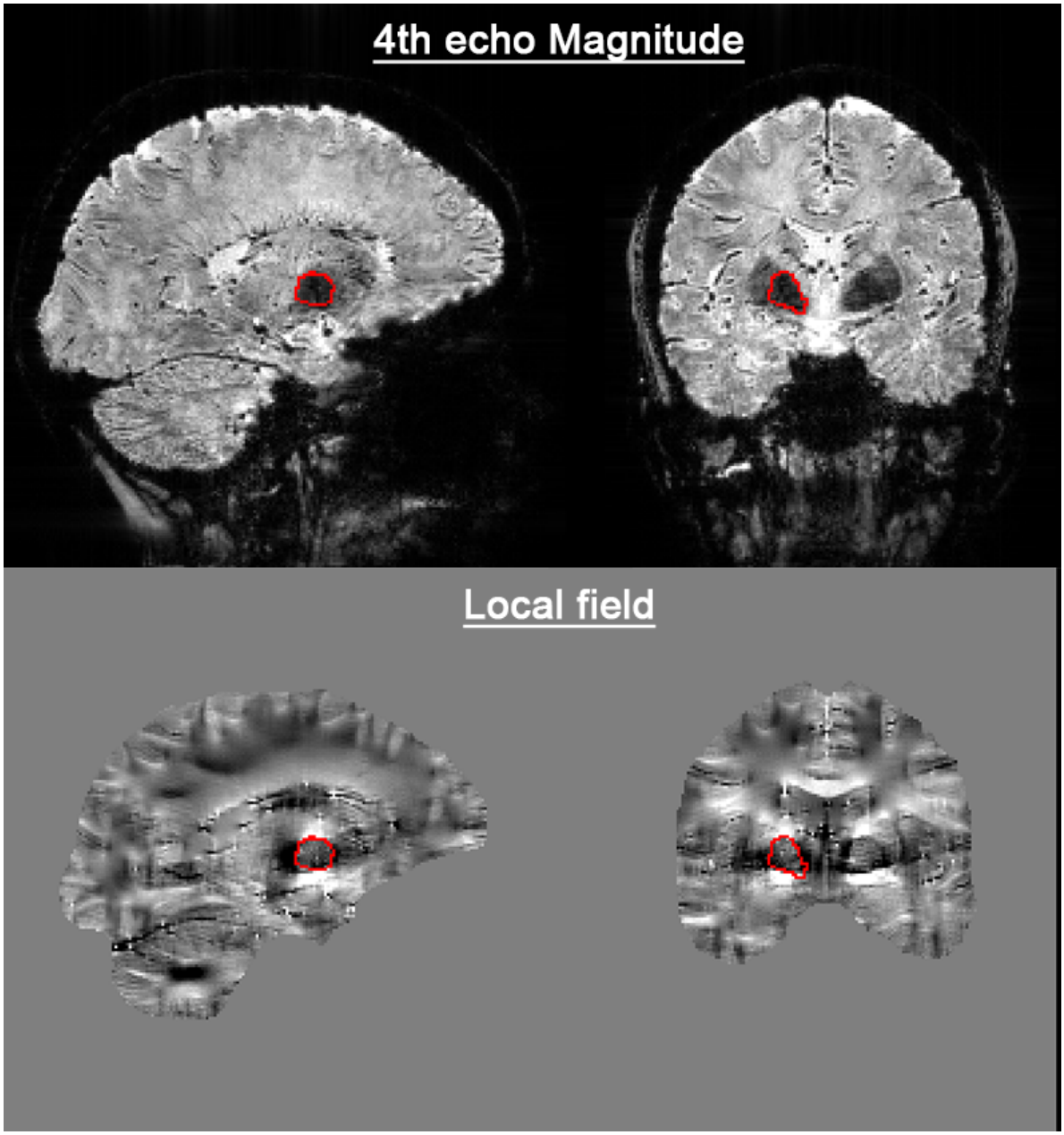
   Open the local field map with any NIfTI image view (e.g. `FSLeyes` or `mricron`). Adjust the display window to 'Min -7' and 'Max 7'.

   Note you can clearly see the contrast between grey and white matter, and veins and tissue now. Other structures as the globus pallidus, red nuclei and substantia nigra are visible... but not quite normal.



3. The figure below shows the last echo of magnitude data, where the globus pallidus is circled by red line, based on the image intensity, and the corresponding local field map. It is known that globus pallidus has a high iron content which can generate a strong induced magnetic field.

   Can you guess the shape and sign of the magnetic properties of the globus palllidus?



You can close all the NIfTI viewers now.

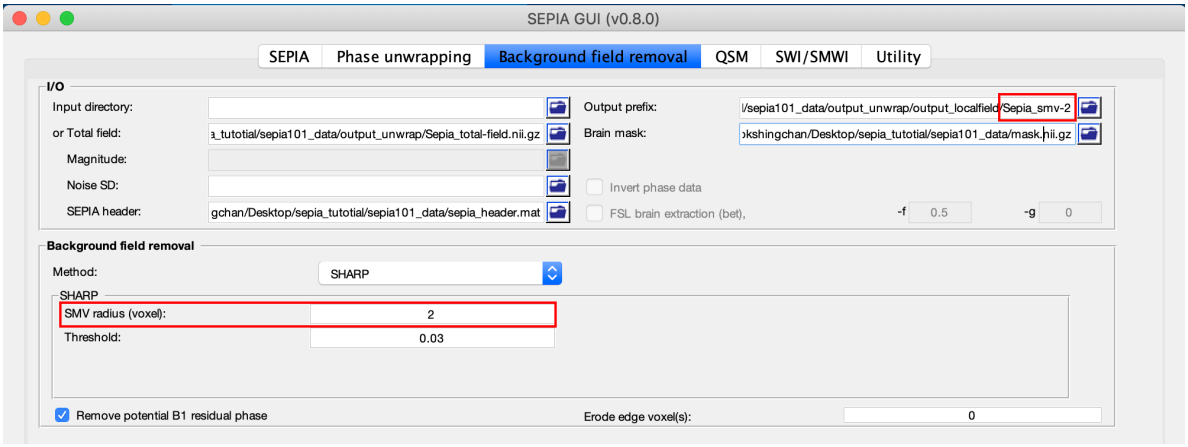
Proceed to *Exercise 4*.

Exercise 3.2 (Advanced)

Note: If you still have enough time, follow the exercise below.

In this exercise, we will focus on the effect of using a different 'SMV radius (voxel)' value.

1. Change the **Output prefix** to: *~/sepia_tutorial/sepia101_data/output_unwrap/output_localfield/Sepia_smv-2*
2. Set **SMV radius (voxel)** in the **Background field removal** panel to: 2



3. Press the **Start** button. Again, when the process is finished, you will see the message: '*Processing pipeline is completed!*'.
4. Try open *Sepia_smv-2_local-field.nii.gz* and *Sepia_smv-4_local-field.nii.gz* at the same time. Adjust the display window to 'Min -7' and 'Max 7' for both images. What differences can you see between the two results?

Proceed to *Exercise 4*.

Back to *Exercise 2*.

Exercise 4

Objectives

- Understanding QSM dipole inversion
- Gaining experience to use QSM algorithms

Data Required

Data	Description
Sepia_local-field.nii.gz	Local (tissue) field map in Hz , in *~/sepia_tutorial/sepia_data/output_unwrap/output_localfield/*
Sepia_smv-4_mask-qsm.nii.gz	Signal mask for QSM step , in *~/sepia_tutorial/sepia_data/output_unwrap/output_localfield/*
sepia_header.mat	contains important information such as the echo times (TE) and magnetic field strength (in Tesla), and orientation of the acquisition regarding the physical coordinates of the scanner. These are important to compute the magnetic susceptibility with the correct units and ensure the physical model is correct , in *~/sepia_tutorial/sepia_data/*

Estimated time

About 15 min.

The Last Step

The last step of QSM processing is to deconvolute the local (tissue) field by the unit dipole field, such that the tissue magnetic susceptibility can be revealed. This can be described by:

$$\chi = F^{-1}[\frac{F(Tissuefield)}{F(d)}] \tag{1.18}$$

where F and F^{-1} are the Fourier and inverse Fourier transform operators.

This is the so-called dipole inversion of QSM, which is just the element-wise division between the Fourier transforms of the two images.

Theory: Dipole Inversion

When looking at the local field you can see some beautiful contrasts between brain tissues. These local fields represent all the secondary magnetic field induced by the tissues (which are our magnetic susceptibility sources). The tissue is suddenly behaving like a small magnet and the effect of the magnetic field extends beyond the source itself.

Note: Think of a magnet that can attract a metal towards it even when the two objects are not intact. It is because the magnetic field generated by the magnet extends outside the magnet body itself.

The local fields generated by the magnetic susceptibility sources (χ), can be computed using the following equation:

$$Tissuefield = \chi * d \tag{1.19}$$

where d is a unit (dipole) field that the source generated and has the following shape:

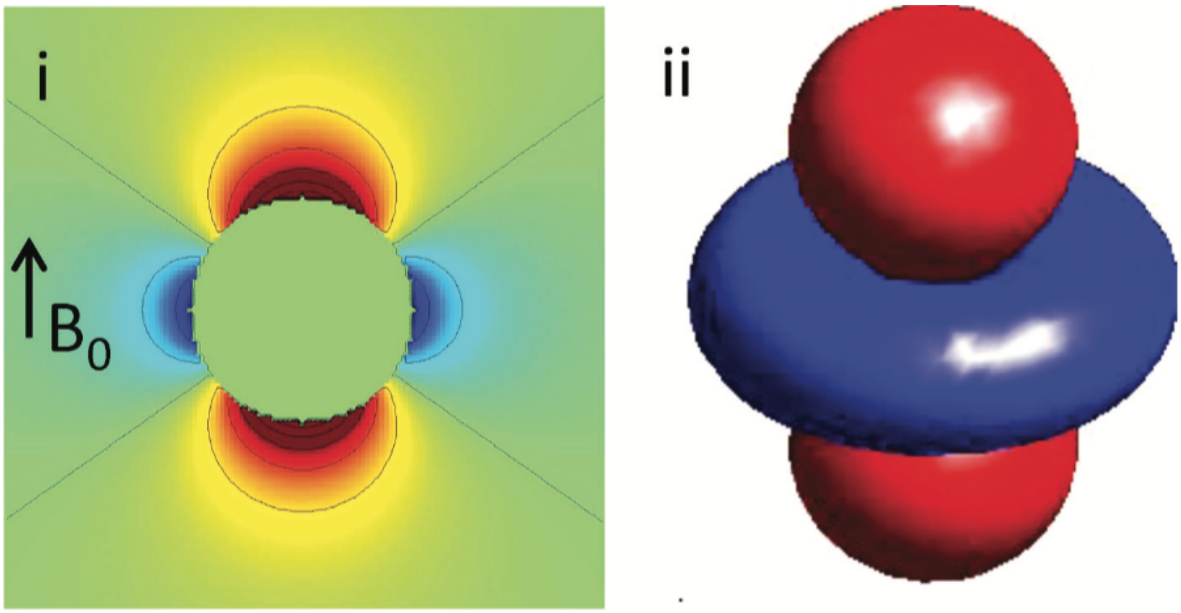


Fig. 1.6: Figure 1: An illustration of a unit dipole field in (i) sagittal section and (ii) surface rendered contour. Red colour represents a positive magnetic field and blue colour represents a negative magnetic field. (Reproduced from Wang & Liu MRM 2014, Wiley)

Eq. (1.19) basically means that the secondary magnetic field experienced by the tissue at each location is the summation of the fields generated by all other (surrounding) sources. Since we have the prior knowledge about the shape of

the magnetic field generated by a unit source (which is d , the unit dipole field) and the field generated by the tissues (output of exercise 3, *local-field.nii.gz*), mapping the magnetic susceptibility of the tissue is just the deconvolution of these two spatial maps (a.k.a. dipole inversion).

Back to *Exercise 4*.

Sounds simple, isn't it? Let's try it out!

QSM: Dipole inversion

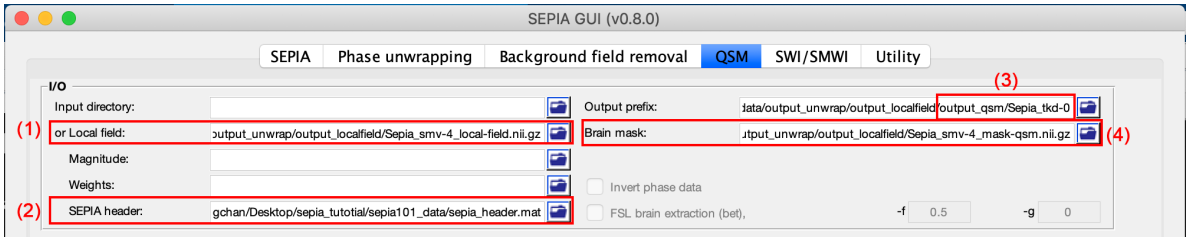
Move to the **QSM** tab of SEPIA.

Exercise 4.1

I/O panel:

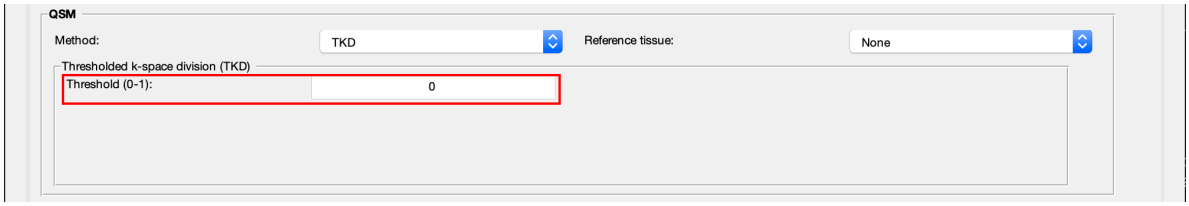
1. Select the **or Local field** input: *Sepia_smv-4_local-field.nii.gz* (in *~/sepia_tutorial/sepia_data/output_unwrap/output_localfield/*),
2. Select the **SEPIA Header**: *Sepia_header.mat* (in *~/sepia_tutorial/sepia_data/*),
3. Change the **Output prefix** of the output to: *~/sepia_tutorial/sepia_data/output_unwrap/output_localfield/output_qsm/Sepia_tkd-0*,
4. Select the **Brain mask**: *Sepia_smv-4_mask-qsm.nii.gz* (in *~/sepia_tutorial/sepia_data/output_unwrap/output_localfield/*).

Note: An updated brain mask has to be used here because some edge voxels were excluded from the original brain mask in the last operation.



QSM panel:

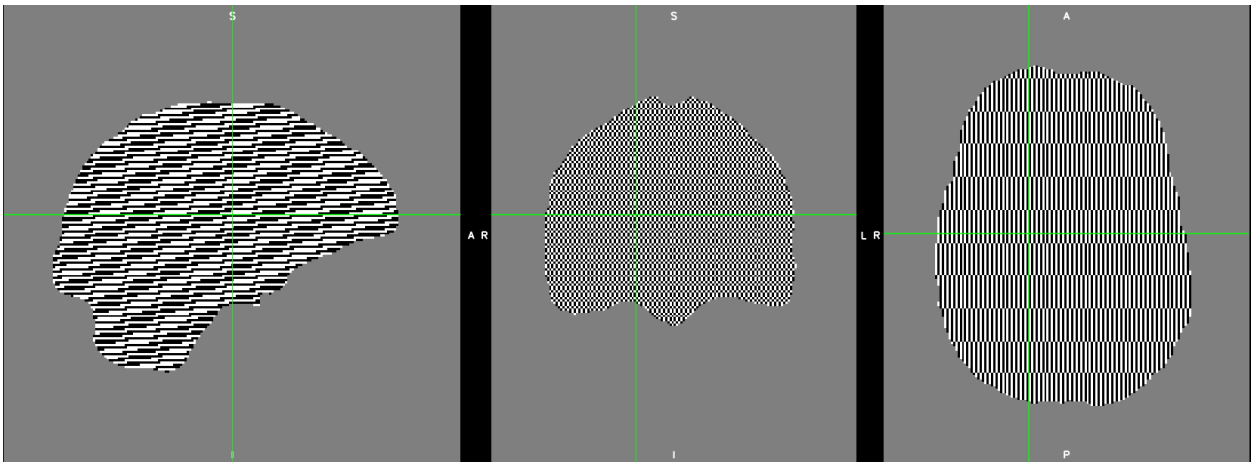
5. To do exactly the operation as in Eq. (1.18), set the threshold of the **TKD** algorithm to '0' and press **Start**.



6. Check the result *Sepia_tkd-0_QSM.nii.gz* in the output directory. Set the display window to 'Min. -0.1' and 'Max. 0.1' (ppm). Does it look like the QSM map as we expected?

Answer: Exercise 4.1

The result should look like the image below. We followed exactly the operation described in the dipole inversion equation, but what's wrong?



To understand the problem, let's have a look at the denominator of the dipole inversion equation, which is the Fourier transform of the dipole field (so-called dipole kernel).

$$\chi = F^{-1}[\frac{F(Tissuefield)}{F(d)}]$$

$$Dipolekernel, D = F(d)$$

Here is the image representation of the magnitude of the dipole kernel $|D|$

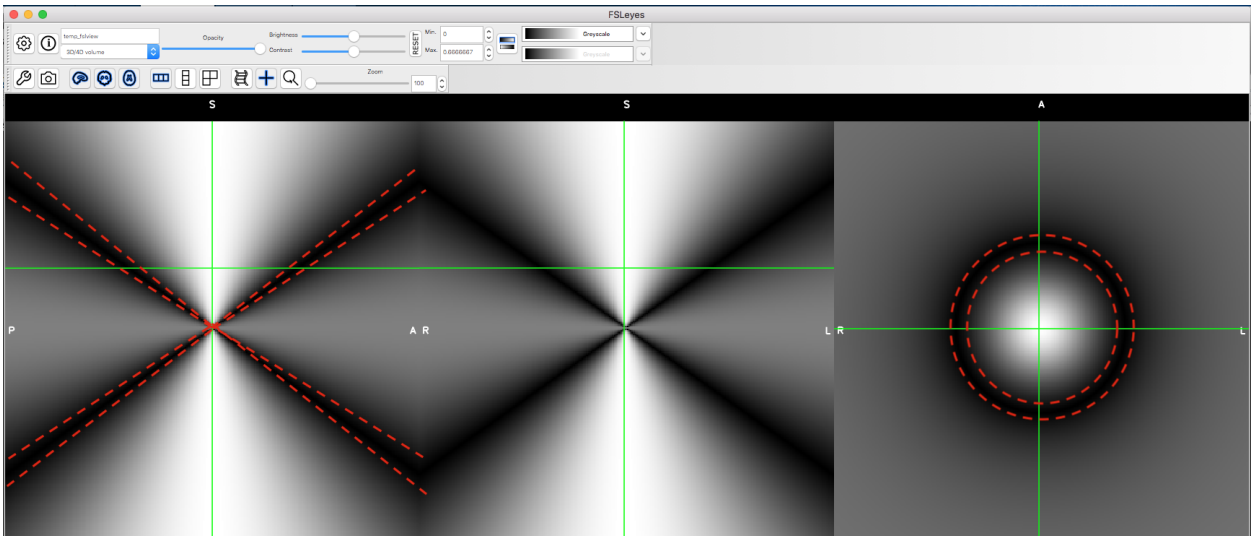
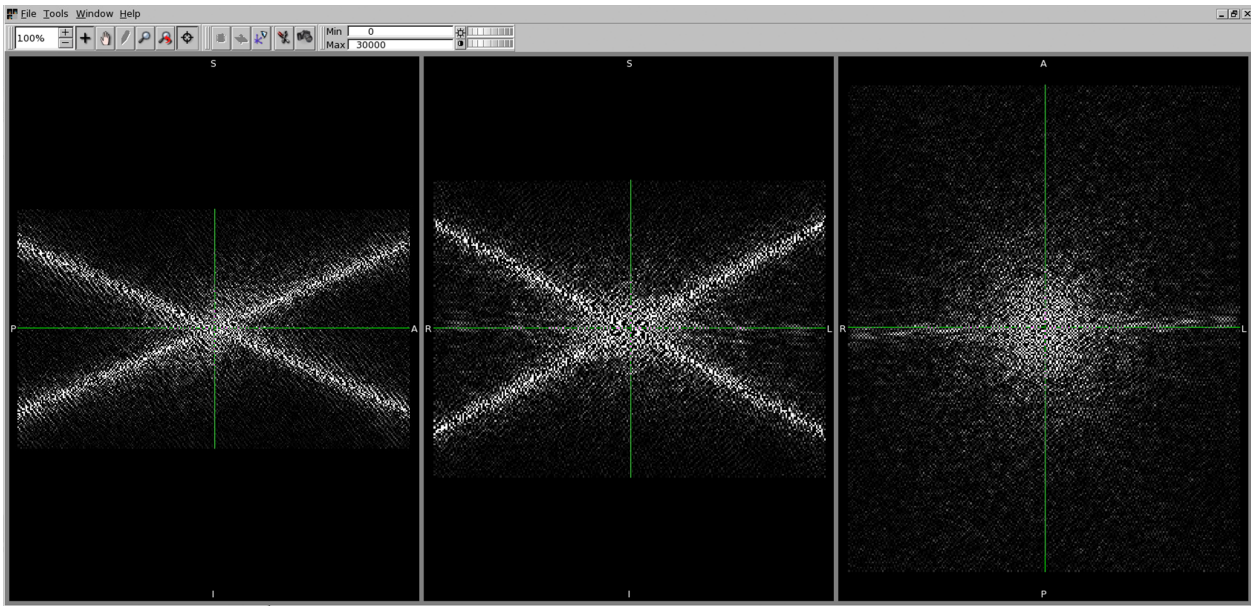


Fig. 1.7: Figure 1: The magnitude of the Fourier transform of the dipole field.

The dashed lines outlines the regions where the values in the dipole kernel are equal or very close to zero (which can be represented as a cone surface). This leads to the ill-defined division of zero problem! After dividing the two images, these regions will contain either undefined value (when values in dipole kernel equal to zeros) or unrealistic numbers (when values close to zeros), making the QSM map unusable:

Figure 2: Fourier transform of the QSM map (i.e. in k-space). It is clear that the values on/near the cone surface in k-space are amplified.

Back to *Exercise 4*.



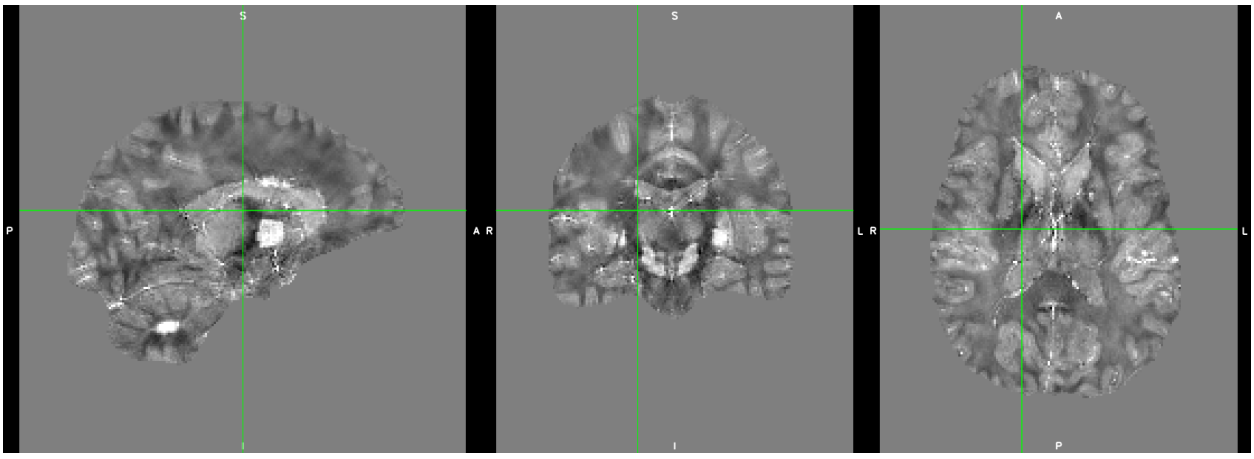
Exercise 4.2

To avoid the previous QSM map we can increase the threshold value of the **TKD**.

1. Change the **Output orefix** to: *~/sepia_tutorial/sepia_data/output_unwrap/output_localfield/output_qsm/Sepia_tkd-0p15*.
2. Change the threshold of the **TKD** algorithm to 0.15 and press **Start**.
3. Check the result *Sepia_tkd-0p15_QSM.nii.gz* in the output directory. Display it along with the *Sepia_tkd-0_QSM.nii.gz*. Set the display window to 'Min. -0.1' and 'Max. 0.1' (ppm). Do you see any improvement?

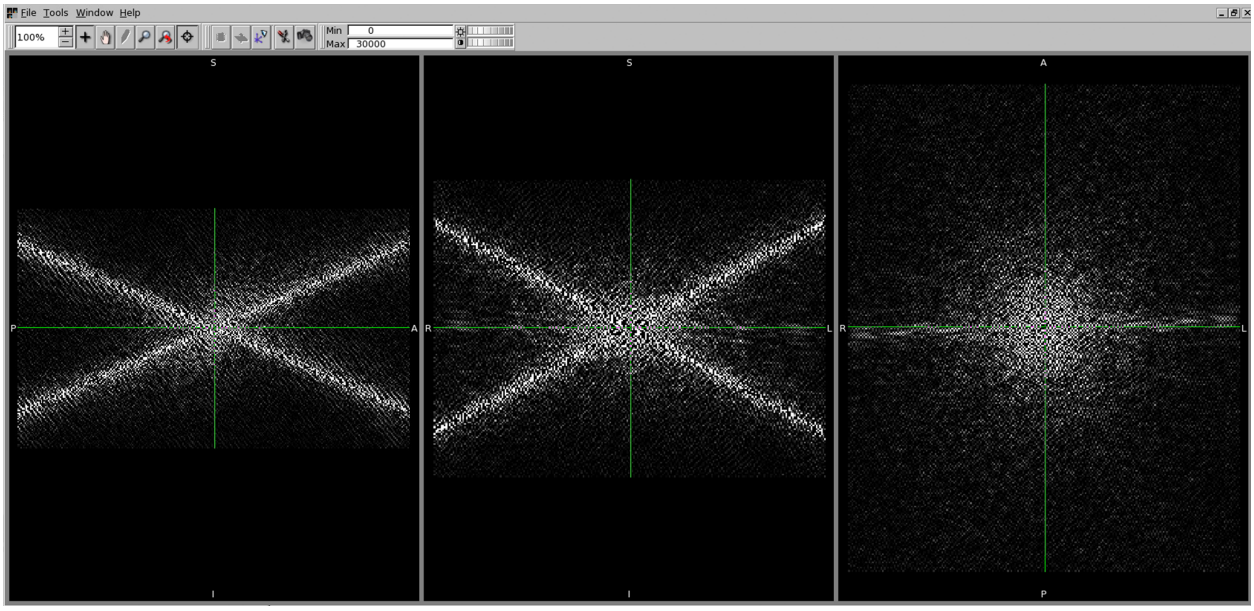
Answer: Exercise 4.2

You should now see some brain structures in the QSM map.

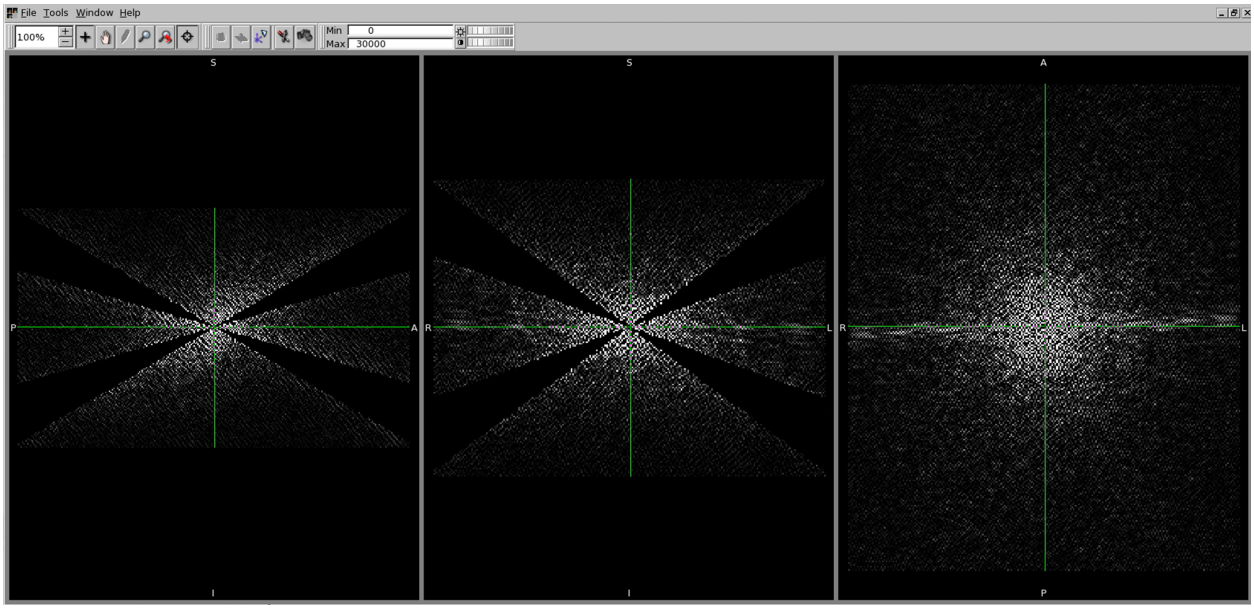


The idea of the TKD method is very straightforward. Since we know the location in which the division results are unreliable (i.e. when D equal/close to zero), we can discard the information in these regions by replacing their values to zero after the element-wise division.

For example, without thresholding the QSM k-space:



and we can threshold the above k-space when the magnitude of D is smaller than, e.g. 0.15, leading to



The Fourier transform of the above image is the QSM map we obtained in this exercise.

Tip: However, the larger is the threshold value, the more is the information being discarded. Therefore, increasing the threshold can improve the appearance of the resulting QSM map (as artefacts reduced), we are also losing the accuracy between our input (i.e. local field map) and output (i.e. QSM map).

Back to *Exercise 4*.

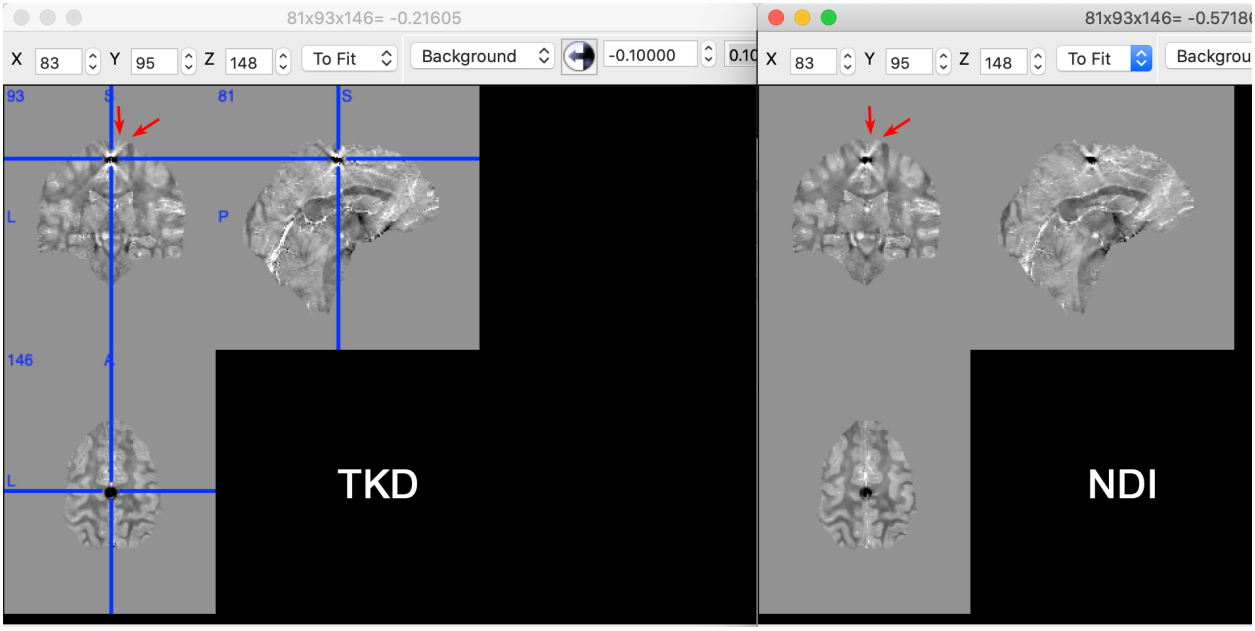
Congratulations! You have finished all the exercises in this tutorial. If you still have time left, finish the advanced exercises or experiment with different QSM methods/methods' parameters.

Advanced Exercise 4.3

To further improve the quality of the QSM map, some methods, such as non-linear dipole inversion (NDI), incorporate additional information, e.g. SNR weighting, with advanced processing algorithm.

1. Select the **Weights** input: *Sepia_weights.nii.gz* (in *~/sepia_tutorial/sepia_data/output_unwrap/*),
2. Change the **Output prefix** to: *~/qsm_tutorial/data/output_qsm/Sepia_ndi*.
3. Change the **QSM** method to 'NDI' and keep the default setting. Press **Start**. It will take about a few minutes to finish.

Open the result *Sepia_tkd-0p15_QSM.nii.gz* and *Sepia_ndi_QSM.nii.gz* togther. If you are using `mricron`, go to location [83,95,148], which is the location of calcification in the phantom data. Can you see that the NDI's result has less artefact?



Back to *Exercise 3*.

1.20 Integration of New Phase unwrapping/BFR/QSM Method in SEPIA: Part 1

1.20.1 Objectives

- Learn how to add a new method to SEPIA framework
- Understanding structure of SEPIA processing backend

Target Audience

- who has some experience with SEPIA
- researchers who want to add their method(s) to the SEPIA framework

Estimated Time

About 30 minutes

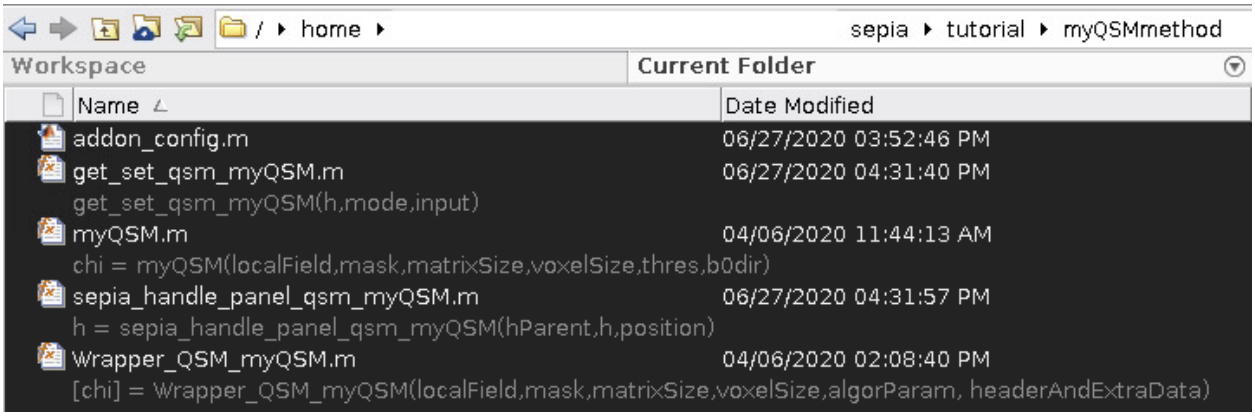
1.20.2 Introduction

In this tutorial, we will practice how to integrate a new method to the SEPIA processing backend. The new method can be used in phase unwrapping/backgorund field removal/QSM dipole inversion step.

Note: This tutorial only demonstrates the way of adding new methods to the processing backend. To be able to have your method also shown in the SEPIA frontend GUI, please visit Part 2 of the tutorial.

1.20.3 Exercise

To begin with, let's go to the tutorial directory `$SEPIA_HOME/tutorial/`. There is a folder called *myQSMmethod*. You should be able to see there are four Matlab scripts in the folder:



We will need `myQSM.m`, `Wrapper_QSM_myQSM.m` and `addon_config.m` these three files in this tutorial.

We are going to explain these files in detail.

myQSM.m

Let's have a look at `myQSM.m`

`myQSM.m` is basically a thresholded k-space method to perform the QSM dipole inversion process. It requires 4 essential and 2 optional input variables:

Essential variables

- *localField*: 3D matrix of a local (tissue) field map, unit is unimportant in this function
- *mask*: 3D matrix of a signal mask
- *matrixSize*: 1-by-3 array to indicate the matrix size of the local field map
- *voxelSize*: 1-by-3 array to indicate the spatial resolution of the local field map, in mm

Optional variables

```
function chi = myQSM(localField,mask,matrixSize,voxelSize,thres,b0dir)
% if the last two input variables are not provided, set to a default value
if nargin < 6
    b0dir = [0,0,1];
end
if nargin < 5
    thres = 0.15;
end

% display message
fprintf('Threshold for k-space division is %f \n',thres);

%% Core
% dipole kernel
kernel = DipoleKernel(matrixSize,voxelSize,b0dir);

% initiate inverse kernel with zeros
kernel_inv = zeros(matrixSize, 'like', matrixSize);

% get the inverse only when value > threshold
kernel_inv( abs(kernel) > thres ) = 1 ./ kernel(abs(kernel) > thres);

% direct dipole inversion method
chi = real( ifftn( fftn(localField) .* kernel_inv ) ) .* mask;

end
```

- *thres*: a threshold of k-space cooridate to avoid division-by-zero problem
- *b0dir*: main magnetic field direction with respect to the local field map

The function returns one output variable which is the magnetic susceptibility map, *chi*, and has the same unit as the local field map input.

Note: This function is equivalent to your own QSM function to perform dipole field inversion.

Wrapper_QSM_myQSM.m

`Wrapper_QSM_myQSM.m` is a wrapper function to connect `myQSM.m` to the SEPIA framework. It belongs to the comminication level of SEPIA. We will go through the function step by step to understand how it works:

```
% You can change the name of the function but DO NOT change the input/output variables
function [chi] = Wrapper_QSM_myQSM(localField,mask,matrixSize,voxelSize,algorParam, headerAndExtraData)
% load some constants
sepia_universal_variables;

% get algorithm parameters, if user doesn't specify them then set some default values
algorParam = check_and_set_algorithm_default(algorParam);
% method     = algorParam.qsm.method;
thre_tkd   = algorParam.qsm.threshold;  % here you define how can SEPIA store the user input in the 'algorParam' variable

% get extra data such as magnitude/weights/B0 direction/TE/etc.
headerAndExtraData = check_and_set_SEPIA_header_data(headerAndExtraData);
b0dir = headerAndExtraData.b0dir;
b0    = headerAndExtraData.b0;
% magn  = headerAndExtraData.magn;  % you can access the magnitude and/or other data from the 'headerAndExtraData' variable

% add path
sepia_addpath;

%% Display algorithm parameters
disp('The following parameter is being used...');
disp(['K-space threshold value  = ' num2str(thre_tkd)]);

%% main
% you can change the unit before your method if you wish
% localField = localField/(b0*gyro); % convert from Hz to ppm

chi = myQSM(localField,mask,matrixSize,voxelSize,thre_tkd,b0dir);

% make sure the output susceptibility map is in 'ppm' which is the default
% unit in SEPIA
chi = chi/(b0*gyro); % convert from Hz to ppm

end

%% set default parameter if not specified
function algorParam2 = check_and_set_algorithm_default(algorParam)

algorParam2 = algorParam;

try algorParam2.qsm.threshold = algorParam.qsm.threshold; catch; algorParam2.qsm.threshold = 0.15; end

end
```

Anatomy of Wrapper_QSM_myQSM.m

```
function [chi] = Wrapper_QSM_myQSM(localField,mask,matrixSize,voxelSize,algorParam,
↪headerAndExtraData)
```

First of all, you can define the function's name with your own preference but the format of input and output variables in this wrapper function are fixed and you should not make any changes on them.

```
% load some constants
sepia_universal_variables;
```

Some constant terms such as the gyromagnetic ratio of 1H are used in various occasions and can be called using the `sepia_universal_variables` function.

```
% get algorithm parameters, if user doesn't specify them then set some default values
algorParam = check_and_set_algorithm_default(algorParam);
thre_tkd   = algorParam.qsm.threshold;  % here you can define how SEPIA will store␣
↪the user input in the 'algorParam' variable
```

In this example, we need the threshold value defined by the user to threshold the k-space coordinate in `myQSM.m`. All the user-defined parameters of the chosen method(s) are stored in *algorParam* input in SEPIA. The variable name (e.g. *algorParam.qsm.threshold* here) is defined by the developer and used in the pipelin configuration file. `check_and_set_algorithm_default` is a nested function to make sure the required variable is set (either by user or using the default value) before it is used.

```
% get extra data such as magnitude/weights/B0 direction/TE/etc.
headerAndExtraData = check_and_set_SEPIA_header_data(headerAndExtraData);
b0dir = headerAndExtraData.b0dir;
b0    = headerAndExtraData.b0;
% magn  = headerAndExtraData.magn;  % you can access the magnitude and/or other data␣
↪from the 'headerAndExtraData' variable
```

To create a dipole kernel with correct orientation, the algorithm needs to know the main magnetic field direction which can be obtained from the *headerAndExtraData* variable. If the multi-echo magnitude data and/or SNR-weighted map are needed, they can also be accessed in this variable as well.

```
% add path
sepia_addpath;
```

You can add the required path(s) in the function.

```
%% Display algorithm parameters
disp('The following parameter is being used...');
disp(['K-space threshold value  = ' num2str(thre_tkd)]);
```

You can also provide some feedback to user by displaying the algorithm parameters/other information in the function.

```
%% main
% you can change the unit before your method if you wish
% localField = localField/(b0*gyro); % convert from Hz to ppm

chi = myQSM(localField,mask,matrixSize,voxelSize,thre_tkd,b0dir);

% make sure the output susceptibility map is in 'ppm' which is the default
% unit in SEPIA
chi = chi/(b0*gyro); % convert from Hz to ppm
```

Once all input are ready, you can call your method to compute the susceptibility map (or local field map, depended on the objective of the method). Feel free to adapt the data for the needs of the method. The only requirement is to return the susceptibility map, *chi*, with unit of ppm.

With these two files, the method is almost ready for SEPIA. Before we can use this method in SEPIA, we need to tell SEPIA there is a new method available. To do so, we need the `addon_config.m` file.

addon_config.m

A new method in SEPIA can only be detected when the `addon_config.m` file is available together with the method itself. It provides crucial information such as script names and method name for SEPIA to support its functionality.

```
1  %% addon_config
2  %
3  % Description: Addon script to incorporate new method to the SEPIA framework
4  %
5  % Kwok-shing Chan @ DCCN
6  % k.chan@donders.ru.nl
7  % Date created: 27 June 2020
8  % Date modified:
9  %
10 % DO NOT change the variable names, update the variable value instead
11 %
12 %% main
13
14 % This name will be used thorough the SEPIA framework for usage
15 addons.method = 'myQSM';
16
17 % Specify the filename of the wrapper function (without extension)
18 addons.wrapper_function = 'Wrapper_QSM_myQSM';
19
20 % Specify the filename of the GUI method panel function (without
21 % extension)
22 % (optional): if no GUI support is provided, then set it to empty, i.e.
23 % addons.gui_method_panel    = [];
24 addons.gui_method_panel = 'sepia_handle_panel_qsm_myQSM';
25
26 % Specify the filename of the function to export/import algorithm
27 % parameters (without extension) (optional)
28 % (optional): if no tunable algorithm parameter, then set it to empty, i.e.
29 % addons.config_function     = [];
30 addons.config_function  = 'get_set_qsm_myQSM';
```

Anatomy of addon_config.m

```
% This name will be used thorough the SEPIA framework
addons.method = 'myQSM';
```

You need to specify the name of your method (i.e. 'myQSM' here). This name will be used thorough SEPIA.

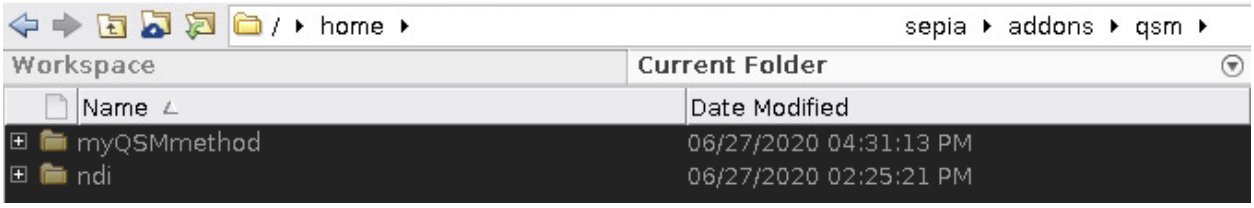
Note: Space is allowed in the name.

```
% Specify the filename of the wrapper function (without extension)
addons.wrapper_function      = 'Wrapper_QSM_myQSM';
```

Here we need to tell SEPIA what's the filename of myQSM wrapper function file.

These two variables will allow SEPIA to access myQSM in the processing backend. We will explain the remaining two in the next tutorial.

Now everything is ready! SEPIA only detects new methods available in the *addons* direction, i.e. `$SEPIA_HOME/addons/`. There are three sub-directories: 'phase_unwrap', 'bfr', and 'qsm'. New method for a specific task must be added to the corresponding sub-directory. Therefore, we need to copy and past the whole folder (i.e. `$SEPIA_HOME/tutorial/myQSMmethod/`) to the QSM addons directory `$SEPIA_HOME/addons/qsm/`



The method is now available in SEPIA! However, the method is only available in the processing backend. You can use the method only in command-based operation such as `SEPIAIOWrapper.m` and `QSMIOWrapper.m`, e.g. :

```
sepia_addpath

% Input/Output filenames
input(1).name = '/input_dir/Sepia_local-field.nii.gz' ;
input(2).name = '' ;
input(3).name = '' ;
input(4).name = '/input_dir/sepia_header.mat' ;
output_basename = '/input_dir/output/Sepia' ;
mask_filename = ['/input_dir/Sepia_mask-qsm.nii.gz'] ;

% General algorithm parameters
algorParam.general.isBET       = 0 ;
algorParam.general.isInvert    = 0 ;
% QSM algorithm parameters
algorParam.qsm.reference_tissue = 'None' ;
algorParam.qsm.method = 'myQSM' ;
algorParam.qsm.threshold = 0.15 ;

QSMMacroIOWrapper(input,output_basename,mask_filename,algorParam);
```

Note: For phase unwrapping method, the structure of `addon_config.m` is slightly different. Please check the `addon_config.m` file in `$SEPIA_HOME/addons/phase_unwrap/segue/` for more information.

Note: The next part of the tutorial is about adding GUI feature to the method (i.e. method panel). For phase unwrapping method, currently no method panel is supported. Having `addon_config.m`, `Wrapper_???_???.m` and your method (e.g. can be Matlab function or compiled library, etc.) would be enough for both frontend and backend of SEPIA.

1.21 Integration of New Phase unwrapping/BFR/QSM Method in SEPIA: Part 2

1.21.1 Objectives

- Learn how to add a new method to SEPIA GUI

Target Audience

- who has completed Part 1 of the tutorial
- researchers who want to add their method(s) to SEPIA framework

Estimated Time

About 1 hour

1.21.2 Introduction

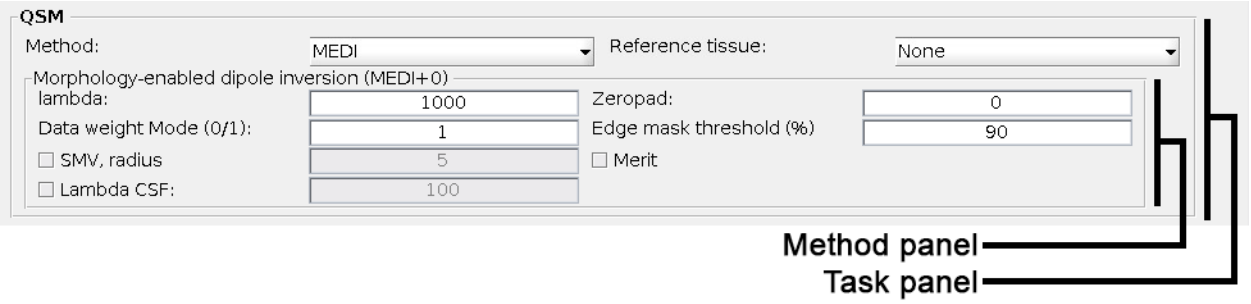
In this tutorial, we will practice how to add a new method to the SEPIA GUI. you should complete the tutorial Part 1 before proceeding to this tutorial.

GUI is a major component of SEPIA. It provides the most straightfoward way to access all avaiable resources of QSM processing in SEPIA. The main goal of the GUI is to generate a pipeline configuration file (sepia_config.m) containing all the processing tasks, methods and algorithm parameters specified by the users and used to trigger the QSM processing. The configuration file can also be executed without initializing the GUI since it directly accesses the processing backend.

The full QSM processing pipeline in SEPIA can be summarised into 4 task panels, including:

1. Data input/output panel
2. Total field recovery and phase unwrapping panel
3. Background field removal panel
4. QSM panel

For each processing task, there are multiple methods available to perform the task. Generally, each method has it own method panel in the GUI to obtain information from the users.



The method panel will be switched from one to another based on the current selected method in the task panel.

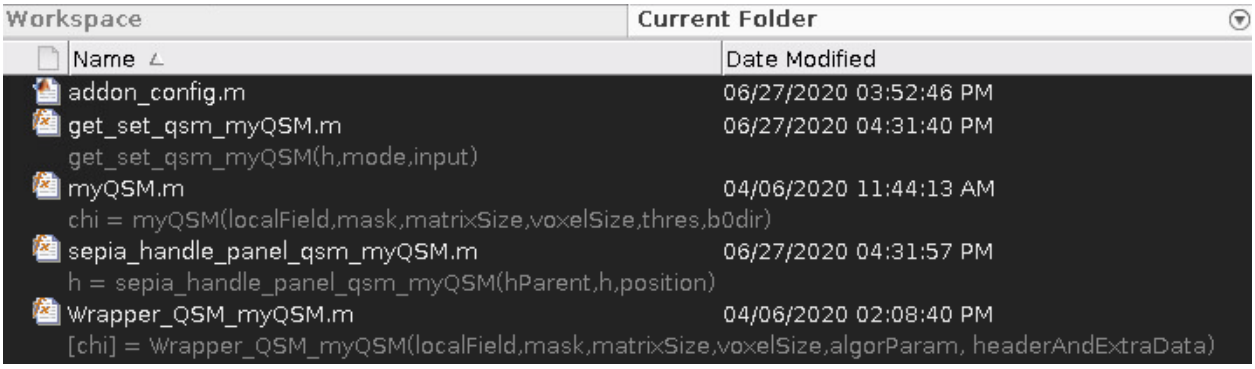
There are two main objectives we need to accomplish in this tutorial:

1. design a method panel that can obtain information from the user, and
2. export and import the information to/from a pipeline configuration file.

1.21.3 Exercise

If you already complete Part 1 of the tutorial, go to the addons directory `$SEPIA_HOME/addons/qsm/myQSMmethod/`. If not, we strongly recommand go through the Part 1 of the tutorial first.

In the addons directory `$SEPIA_HOME/addons/qsm/myQSMmethod/`, you should see there are five Matlab scripts in the folder:



In Part 1, we demonstrated how to connect `myQSM.m` to the SEPIA processing backend using `Wrapper_QSM_myQSM.m` as a connector and `addon_config.m` for SEPIA to load your method in the framework. In this tutorial, we will use the remaining two files: `sepia_handle_panel_qsm_myQSM.m` and `get_set_qsm_myQSM.m`

sepia_handle_panel_qsm_myQSM.m

In Part 1 of this tutorial, we decided that our method `myQSM.m` has one adjustable parameter called `thres`, which is a threshold of the dipole kernel. In the processing backend, users can adjust the threshold value via the user algorithm input structure 'algorParam', e.g.:

```
algorParam.qsm.method = 'myQSM' ;
algorParam.qsm.threshold = 0.15 ;

QSMMacroIOWrapper(input,output_basename,mask_filename,algorParam);
```

We therefore need to design a panel for myQSM such that users can input their desired threshold value in the GUI.

Each method in SEPIA GUI has it own panel to obtain information from users. This information can be a value (e.g. tolerance), a decision (e.g. true/false), a selection (given choices) and many others. We will go through the script to understand how a panel can be designed in SEPIA GUI.

Anatomy of sepia_handle_panel_qsm_myQSM.m

```
function h = sepia_handle_panel_qsm_myQSM(hParent,h,position)
```

For every new panel you can decide a new name of the function. However, the input and output variables are fixed and should not be changed.

```
%% set default values
defaultThreshold = 0.15;
```

We first decide the default input value that will be showed in the GUI.

```
%% Tooltips
tooltip.qsm.myQSM.threshold  = 'K-space threshold';
```

You can also add tooltips to further explain the information the method required.

```
function h = sepia_handle_panel_qsm_myQSM(hParent,h,position)

%% set default values
defaultThreshold = 0.15;

%% Tooltips
tooltip.qsm.myQSM.threshold = 'K-space threshold';

%% layout of the panel
nrow        = 4;
rspacing    = 0.01;
ncol        = 2;
cspacing    = 0.01;
[height,bottom,width,left] = sepia_layout_measurement(nrow,rspacing,ncol,cspacing);

%% Parent handle of TKD panel children

h.qsm.panel.myQSM = uipanel(hParent,...
    'Title','Thresholded k-space division (TKD)',...
    'position',position,...
    'backgroundcolor',get(h.fig,'color'),'Visible','on');

%% Children of TKD panel

    panelParent = h.qsm.panel.myQSM;

    % width of each element in a functional column, in normalised unit
    wratio = 0.5;

    % row 1, col 1
    % text|edit field pair: threshold
    [h.qsm.myQSM.text.threshold,h.qsm.myQSM.edit.threshold] = sepia_construct_text_edit(...
        panelParent,'Threshold (0-1):', defaultThreshold, [left(1) bottom(1) width height], wratio);


%% set tooltips
set(h.qsm.myQSM.text.threshold, 'Tooltip',tooltip.qsm.myQSM.threshold);

%% set callbacks
set(h.qsm.myQSM.edit.threshold, 'Callback', {@EditInputMinMax_Callback,defaultThreshold,0,0,1});

end
```

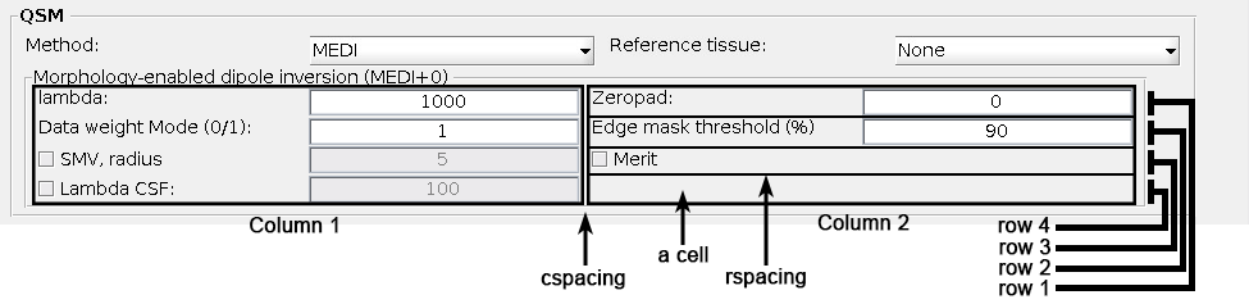
```
%% layout of the panel
nrow        = 4;
rspacing    = 0.01;
ncol        = 2;
cspacing    = 0.01;
[height,bottom,width,left] = sepia_layout_measurement(nrow,rspacing,ncol,cspacing);
```

In principle develops can design the layout of the method panel with their own style. In SEPIA, the `sepia_layout_measurement` function can help standardise the panel layout by creating a evenly distributed grid. It requires the following input:

- *nrow*: number of rows in the grid
- *rspacing*: spacing between consecutive rows, in normalised unit
- *ncol*: number of columns in the grid
- *cspacing*: spacing between consecutive columns, in normalised unit

It returns four variables that specify the position of each cell in the grid:

- *height*: height of the cell, in normalised unit
- *bottom*: 1-by-*nrow* array indicating the bottom position of the cell, starting from the top of the panel
- *width*: width of the cell, in normalised unit
- *left*: 1-by-*ncol* array indicating the left position of the cell, starting from the left



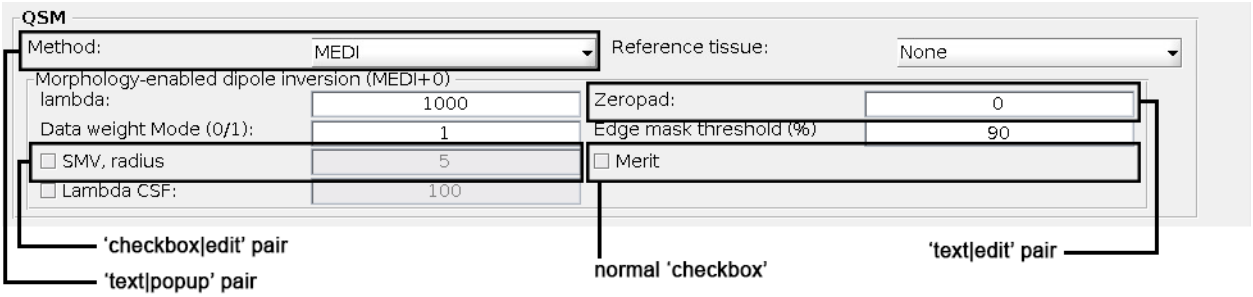

```
h.qsm.panel.myQSM = uipanel(hParent,...
   'Title','My QSM dipole inversion',...
   'position',position,...
   'backgroundcolor',get(h.fig,'color'),'Visible','on');
```

Firstly, we create a panel in SEPIA. This panel belongs to the QSM task panel which is specified in the *hParent* input. The only thing you can change is the '*Title*' value here.

```
panelParent = h.qsm.panel.myQSM;

 % width of the first element in a cell, in normalised unit
 wratio = 0.5;

 % row 1, col 1
 % text|edit field pair: threshold
 [h.qsm.myQSM.text.threshold,h.qsm.myQSM.edit.threshold] = sepia_construct_text_edit(.
↪..
     panelParent,'Threshold (0-1):', defaultThreshold, [left(1) bottom(1) width␣
↪height], );
```

Secondly, we can start adding operational functions to the method panel. There are many operations you can add to the method panel in order to obtain input from users. SEPIA provides three functions to simplify the work of adding operations to the panel, including:

1. `sepia_construct_text_edit`: create a 'text|edit' pair to obatin (numerical) input from users;
2. `sepia_construct_text_popup`: create a 'text|popup' pair to obatin predefined input from users by selection;
3. `sepia_construct_checkbox_edit`: create a 'checkbox|edit' pair to obatin a logical decision (true or false) from users plus an optional numerical input.



These three functions cover most of the operations in SEPIA. For detail description of how the functions work please check the header of the functions. In this tutorial, we only use the `sepia_construct_text_edit` function to obatin the k-space threshold value from the user.

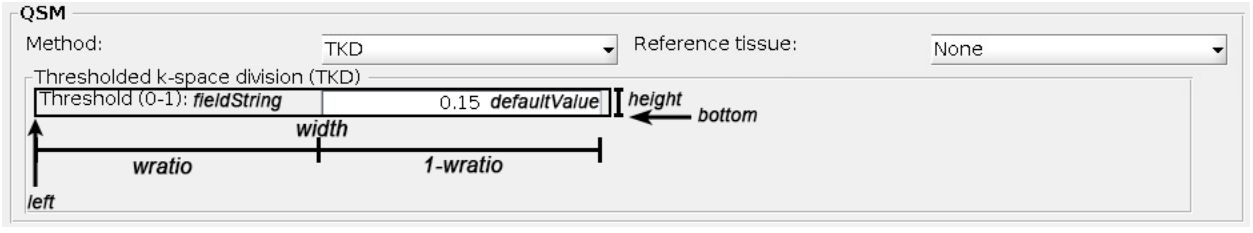
```
function [h_text,h_edit] = sepia_construct_text_edit(parent,fieldString,defaultValue,
↪pos,wratio)
```

`sepia_construct_text_edit` requires 5 input variable:

- *parent*: parent handle of the operation, which is the handle of the panel (e.g. *h.qsm.panel.myQSM*)
- *fieldString*: the text displayed in the 'text' field of the operation (e.g. *'Threshold (0-1):'*)
- *defaultValue*: the value displayed in the 'edit' field of the operation (e.g. *defaultThreshold*)
- *pos*: the position of the entire operation ('text'+'edit' fields), [left bottom width height] (e.g. [*left(1) bottom(1) width height*])
- *wratio*: the normalised width taken by the 'text' field.

The function returns two output variables:

- *h_text*: handle of the 'text' field, (e.g. *h.qsm.myQSM.text.threshold* in this tutorial)
- *h_edit*: handle of the 'edit' field, (e.g. *h.qsm.myQSM.edit.threshold*)



These three SEPIA functions are resbonsible for only creating the GUI components. The function of these operations are still missing.

```
%% set tooltips
set(h.qsm.myQSM.text.threshold, 'Tooltip',tooltip.qsm.myQSM.threshold);
```

Here we set the tooltips that was defined in the beginning of the file to the 'text' field of the panel.

```
%% set callbacks
set(h.qsm.myQSM.edit.threshold, 'Callback', {@EditInputMinMax_Callback,
↪defaultThreshold,0,0,1});
```

The callback function allows developer to control the behaviour of the user input. Here we utilise a function called `EditInputMinMax_Callback` in SEPIA to limit the range of the input value from the users. Let's have a look to this function

```
EditInputMinMax_Callback(source,eventdata,defaultValue,isIntegerInput,lb,ub)
```

Ingoring the input variables *source* and *eventdata*, this function takes three extra input from the developer:

- *defaultValue*: whenever an invalid value is entered, returns to this value (e.g. returns to *defaultThreshold* in this tutorial)
- *isIntegerInput*: whether the input is an integer or not (*true* or *1*: input needed to be integer; *false* or *0*: input can be floating number) (e.g. the input can be floating number in this tutorial)
- *lb*: lower bound of the input value (e.g. the minimum number is *0* in this example)
- *ub*: upper bound of the input value (e.g. the maximum number is *1* in this example)

Now, the method panel is ready for the GUI. Our next job is to make sure the user input can be correctly exported to the pipeline configuration file and afterward imported from the pipeline configuration file to the GUI which will be done the next section.

get_set_qsm_myQSM.m

Once the method panel is setup, our final task is to translate this information from the GUI to the pipeline configuration file (sepia_config.m) in a way the processing backend can understand. This job is done by the `get_set_qsm_myQSM.m` file.

In Part 1 of this tutorial, we defined a variable named `algorParam.qsm.threshold` in the wrapping function `Wrapper_QSM_myQSM.m` to allow user to adjust the threshold on the dipole kernel, i.e.

```
% get algorithm parameters, if user doesn't specify them then set some default values
algorParam = check_and_set_algorithm_default(algorParam);
thre_tkd   = algorParam.qsm.threshold;  % here you can define how SEPIA will store␣
↪the user input in the 'algorParam' variable
```

The usage of the variable name `algorParam.qsm.threshold` is consistent thorough all levels of the SEPIA framework. This means the name and the structure of `algorParam.qsm.threshold` are the same in the sepia_config.m file as in the `Wrapper_QSM_myQSM.m` file which is the connector between SEPIA and the main script of processing.

In a nutshell, `get_set_qsm_myQSM.m` obtains the user input value from the GUI and converts it in a correct format in the pipeline configuration file. It is also responsible to read the value from the pipeline configuration file and update the number shown in the GUI when users load the pipeline configuration file back to the GUI.

We are going to explain how to archieve all these in `get_set_qsm_myQSM.m`.

Anatomy of get_set_qsm_myQSM.m

```
function get_set_qsm_myQSM(h,mode,input)

str_pattern = {'.qsm.threshold'};

action_handle = {h.qsm.myQSM.edit.threshold};

switch lower(mode)
    case 'set'
        fid = input;

        fprintf(fid,'algorParam%s = %s ;\n' ,str_pattern{1},get(action_handle{1},   'String'));

    case 'get'

        config_txt = input;

        % first edit field
        val             = get_num_as_string(config_txt, str_pattern{1}, '=', ';');
        set_non_nan_value(action_handle{1},'String',val)

end
```

```
function get_set_qsm_myQSM(h,mode,input)
```

`get_set_qsm_myQSM.m` has three predefined input variables. No modfication is allowed here.

```
str_pattern = {'.qsm.threshold'};
```

`str_pattern` stores all the sub-structures you defined in `Wrapper_QSM_myQSM.m`, e.g. the ``.qsm.threshold` part of `algorParam.qsm.threshold`. The string pattern specified here will be printed on the pipeline configuration file and read into GUI when the configuration file is loaded. For methods with multiple input, separate the sub-structure string patterns of the corresponding input using ‘,’.

```
action_handle = {h.qsm.myQSM.edit.threshold};
```

`action_handle` contains all the GUI handles of where the information is exported/imported. In this example, the threshold is specified in `h.qsm.myQSM.edit.threshold` in the GUI function (see above section).

Note: The handle variable stored in `action_handle` must have the same position as in `str_pattern`.

```
switch lower(mode)
```

The variable `mode` here corresponds to

1. ‘set’: exporting the GUI input to a pipeline configuration file;
2. ‘get’: importing values in a pipeline configuration file to the GUI.

```
case 'set'
   fid = input;

   fprintf(fid,'algorParam%s = %s ;\n'       ,str_pattern{1},get(action_handle{1},
↪'String'));
```

In ‘set’ scenario, `get_set_qsm_myQSM.m` tries to export information from the GUI to a pipeline configuration file (sepipa_config.m). The first line `fid = input;` is just to obtain the sepia_config.m file ID in order to write something to the file in the second line.

There are two formatted text data (%s) we need to write, the first %s corresponds to the algorithm parameter sub-structure, in this case adding text specified in `str_pattern` which is '.qsm.threshold' after the text 'algorParam'. The second %s corresponds to the value in the `action_handle` (i.e. `h.qsm.myQSM.edit.threshold`) which is the value specified in the GUI. The input obtained from an 'edit' field in the GUI is in text instead of a number. Therefore %s is used.

```
case 'get'

   config_txt = input;

   % first edit field
   val              = get_num_as_string(config_txt, str_pattern{1}, '=', ';');
   set_non_nan_value(action_handle{1},'String',val)
```

In 'get' scenario, `get_set_qsm_myQSM.m` tries to import information from a pipeline configuration file back into the GUI. Noted that the input variable `input` contains file ID in the 'set' scenario while here it contains all the text in the input pipeline configuration file.

The first task here is to obatin the required value from the pipeline configuration file. This is done via the `get_num_as_string` function, which captures information stated as a number in the text to the Matlab text format. The `get_num_as_string` function has four input variables:

```
str = get_num_as_string(A, str_pattern, start_indicator, end_indicator)
```

The first input, `A`, in a variable containing all text of the pipeline configuration file. The second input, `str_pattern` is a variable that contains a specific string of text, that is '.qsm.threshold' in this example. The third and fourth input, `start_indicator` and `end_indicator` indicate that the position of the required information after `str_pattern`. In this example, the threshold is exported as `algorParam.qsm.threshold = 0.1 ;` specified in the 'set' scenario. Therefore, the threshold value can be captured between the special characters '=' and ';', corresponding to the `start_indicator` and `end_indicator` input.

Once we obtain the threshold value as text format stored in `val`, the second task is to update the corresponding value shown in the GUI, which is done in the last line here.

With these two files ready, our new QSM dipole inversion method can now work properly with the SEPIA GUI. Start the SEPIA GUI and try it out!

Here we demonstrated the simpliest way to incorporate a new method in the SEPIA framework. There are so much more options available to obtain user input for your method, as shown in the MEDI method panel and the FANSI method panel. You can also check out the `sepia_handle_panel_qsm_???.m`, `get_set_qsm_???.m` and `Wrapper_QSM_???.m` files to understand how the GUI function of other existing methods is designed.

1.22 (f)MRI Toolkit 2019

1.22.1 Objectives

- Understanding the background of magnetic susceptibility contrast
- Gaining experience in basic QSM processing

Target Audience

- who is new to SEPIA
- who wants to know some basic knowledge about QSM

Estimated Time

About 1 hour

1.22.2 Introduction

In this tutorial, we will go through the standard processing pipeline for quantitative susceptibility mapping (QSM), a novel contrast mechanism that can provide local tissue magnetic properties.

Theory: QSM physics

Briefly, there are two main types of magnetic property (a.k.a magnetic susceptibility, χ) we can measure with MR:

- **Paramagnetism**: substances with paramagnetic property generate a secondary magnetic field that **enhances** the existing magnetic field generated by the MRI scanner. A typical example is iron either in blood or stored in ferritin;
- **Diamagnetism**: substances with diamagnetic property generate a secondary magnetic field that **reduces** the existing magnetic field strength. Examples included myelin and calcification.

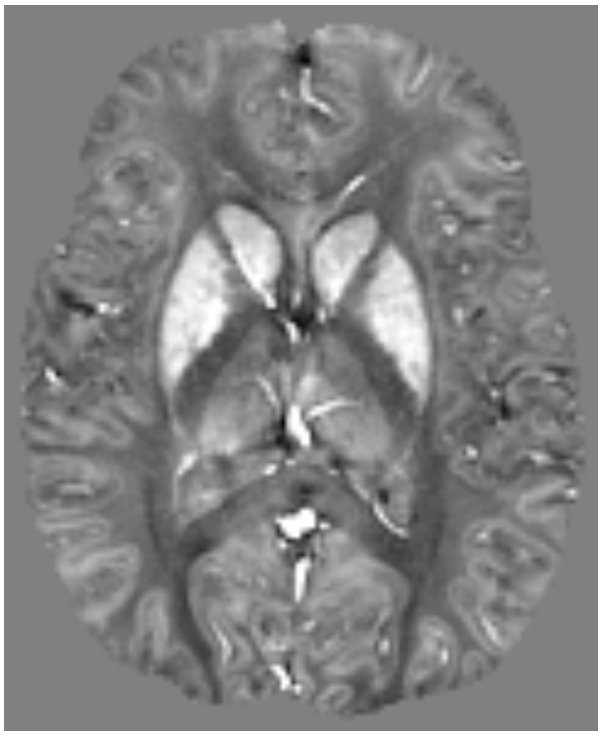


Fig. 1.8: Figure 1: A QSM map. Deep grey matter structures such as globus pallidus and putamen containing high iron concentration is bright (positive magnetic susceptibility) while white matter, which is myelin-rich tissue, is dark (negative magnetic susceptibility). Studies have shown that the susceptibility values in the deep grey matter are highly correlated with iron staining histology result.

Because of the secondary magnetic field generated by (both paramagnetic and diamagnetic) tissues, the overall magnetic field experienced by the water protons will no longer be the same across the whole brain. The strength of this magnetic field inhomogeneity will depend on the local magnetic susceptibility sources: sources with stronger magnetic susceptibility can create a stronger inhomogeneity effect. As a result, the water protons will resonate in different frequencies across the whole brain and the frequency difference at each location is depended on the strength of the

neighbouring sources. Measuring the frequency shift can, therefore, compute the magnetic susceptibility of brain tissues and reveal their cellular environment.

Note: Water protons are the main sources of MRI signal. In QSM, they act as our little magnetic field detectors to reveal the local changes of the magnetic field due to their surrounding environment.

Back to *(f)MRI Toolkit 2019*.

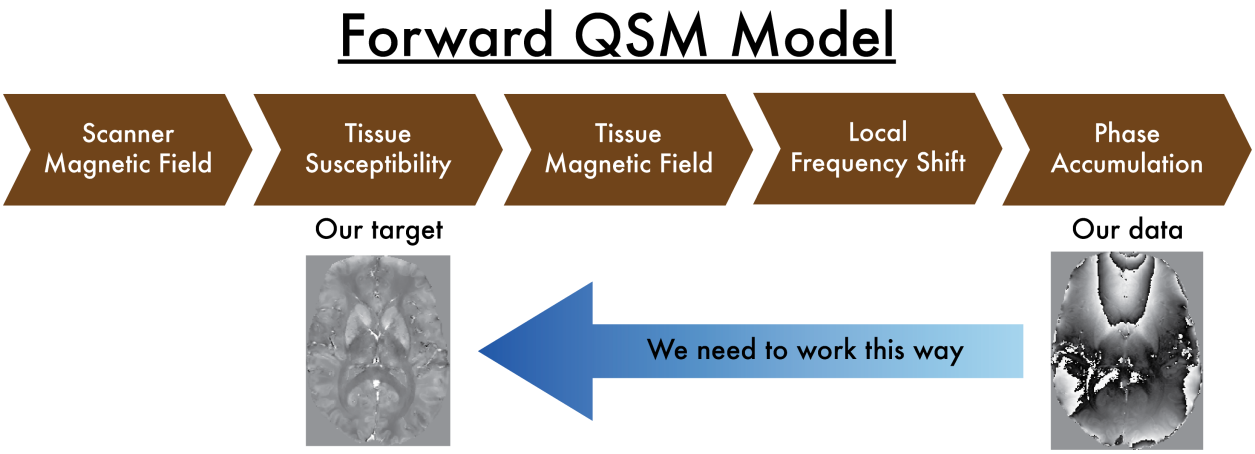


Fig. 1.9: Figure 1: The QSM theory. A magnetic source generates a secondary magnetic field inside the MRI scanner, which will eventually alter the signal phase we measured. Decoding the phase images allows us to detect the molecular environment of the brain tissues.

1.22.3 Exercises

Let's begin with the following exercises that will take you from the phase data to the susceptibility maps!

Exercise 1

Objectives

- Understanding the data required for QSM
- Understanding why we need to correct the phase data before mapping the magnetic susceptibility

Data Required

- a 4D raw phase data (*phase.nii.gz* in the input directory)
- a 4D raw magnitude data (*mag.nii.gz* in the input directory)
- JSON files generated by data conversion software (all .json in the input directory)

Estimated time

About 15 min.

Understanding multi-echo GRE data

To compute a magnetic susceptibility map, multi-echo gradient-echo images are usually used because it can provide phase images.

Theory: MR Phase

As mentioned in the introduction, water protons resonate at different frequencies because of the tissue magnetic susceptibility. The frequency difference in brain tissues can be detected as the difference in phase accumulation over time ($phase = frequency \times time$). Therefore, the phase measurement of the MRI signal allows us to map the effect of the magnetic susceptibility of brain tissues.

It should be noted that the phase can only be measured in the range of [-180, 180] degrees (or $[-\pi, \pi]$ radian).

Back to *Exercise 1*.

Go to the exercise directory which is located in ~/qsm_tutorial/.

You can use the following command in the terminal:

```
cd ~/qsm_tutorial/
```

Note: Here we assume your tutorial directory is in the home directory '~/'. If not, replace '~/' with the path containing the folder 'qsm_tutorial'.

To view the content of the directory use the command: `ls`

```
kwocha@mentat003:~/Documents/Toolkit2019/qsm_tutorial
1091 $ ls
data  sepia
```

You will see there are two folders in the directory.

- **sepia** - contains the software we will use throughout this tutorial;
- **data** - contains the multi-echo gradient echo images we will work on.

Go to the data directory using `cd data` and have a look of the content inside the folder `ls`

```
kwocha@mentat003:~/Documents/Toolkit2019/qsm_tutorial/data
1001 $ ls
echo-1.json  echo-3.json  echo-5.json  echo-7.json  echo-9.json  phase.nii.gz
echo-2.json  echo-4.json  echo-6.json  echo-8.json  mag.nii.gz
```

You should see two NIfTI images (.nii.gz) and a few JSON files (.json) in the directory:

- The NIfTI files *mag.nii.gz* and *phase.nii.gz* contain the magnitude and the phase data acquired with a multi-echo gradient echo sequence.

  Both images are 4D datasets, with the first 3 dimensions containing spatial information (i.e. the image of the brain) and **echo time in the 4th dimension**.

- The JSON files contain important information such as the echo times (TE) and magnetic field strength (in Tesla), and orientation of the acquisition regarding the physical coordinates of the scanner. These are important to compute the magnetic susceptibility with the correct units and ensure the physical model is correct.

Magnitude images

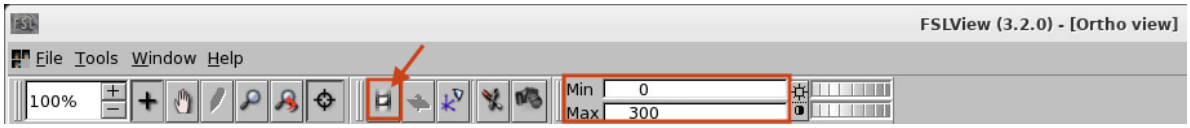
1. Take a look at the magnitude images. You can do this by calling the image viewer FSLView in the terminal:

   ```
   fslview_deprecated mag.nii.gz &
   ```

   Tip: The '&' character will enable the viewer running in the background so you can still work with the current section in the terminal.

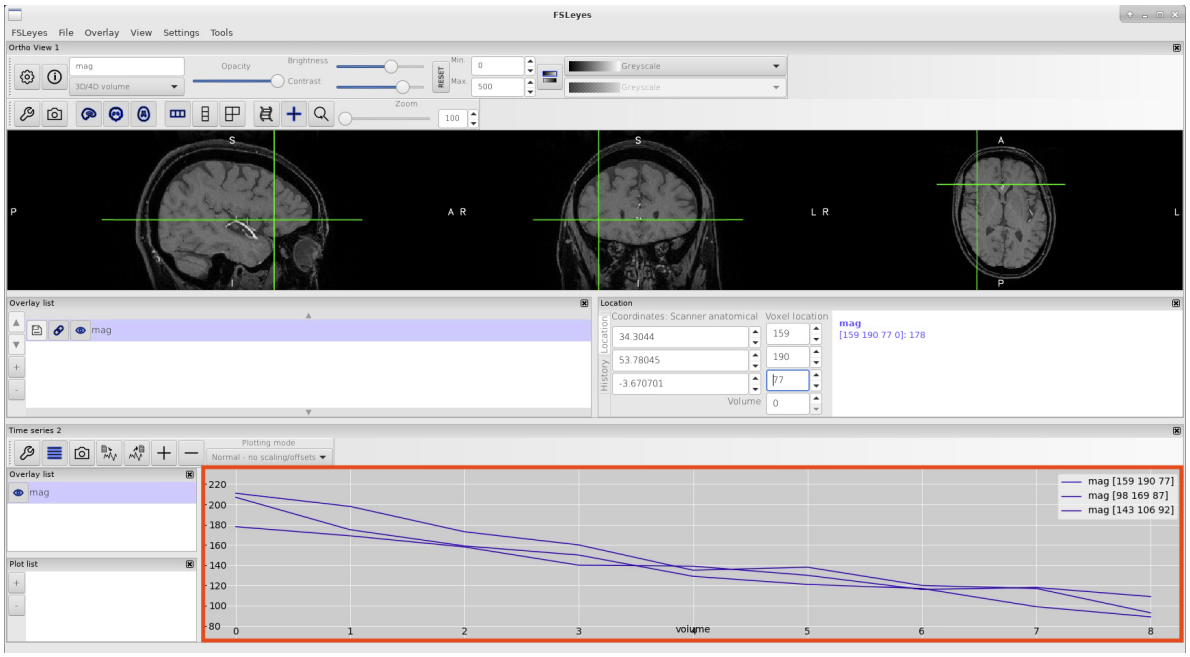
   Note: Due to the file size, it is better to view the images with FSLView instead of FSLeyes.

2. Adjust the display window to 'Min 0' and 'Max 300'.
3. Click the movie button to see how the brain contrast changes with respect to the echo time (time between echoes = 4ms).



4. Click the movie button again to stop the movie. Press `Ctrl+T` to see the plot of signal evolution at different brain tissues over time.
5. Select a few data points in the brain (e.g. caudate nucleus [98 169 87], white matter [143 106 92], and cortical grey matter [159 190 77]), how do you describe the signal change with respect to the echo time in general?

Check Your Progress 1



You should be able to observe the signals decaying over time. Precisely, this is the so-called T_2* signal decay of MR signal which results from water protons losing phase coherence over time. The mechanism of T_2*

decay is complicated because it is due to factors such as field inhomogeneity/diffusion/imperfect shimming, etc. Although QSM mainly works with phase images, magnitude images can be very useful because they share similar (but not always the same!) contrasts, therefore, have the potential to improve the quality of QSM maps.

Back to *Exercise 1*.

6. Go back to the terminal. Compute the mean magnitude image in time:

   ```
   fslmaths mag.nii.gz -Tmean mean_head.nii.gz
   ```

   This image will be used in Exercise 3.

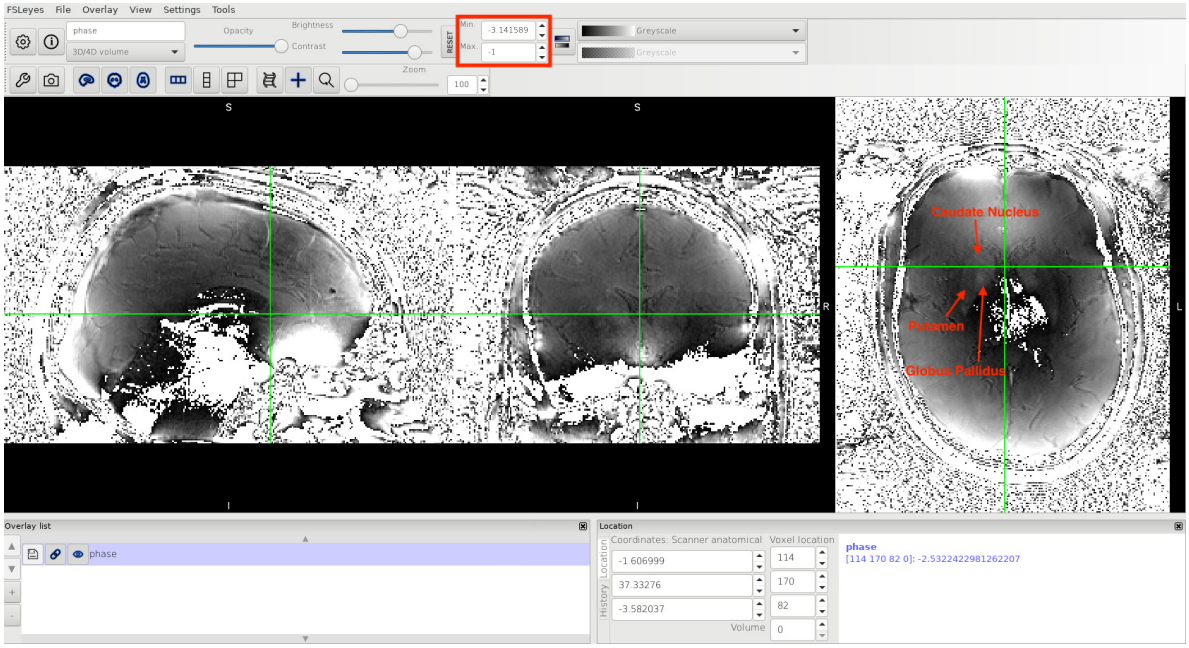
Phase images

1. Look at the phase images:

   ```
   fslview_deprecated phase.nii.gz &
   ```

   The phase images look different compared to the magnitude images and with the current display window it is hard to see any contrast in brain tissues.

2. Adjust the display window to 'Min. -3.14' and 'Max. -1' and go through different slices. You should be able to identify some brain structures (e.g. at slice 82).



3. Change the window back to 'Min. -3.14' and 'Max. 3.14'.

Based on Eq. (1.16), it is expected the phase increases/decreases monotonically. In other words, we should observe the phase contrasts become higher in the later echoes (i.e. bright brighter; dark darker).

$$phase = frequency \times time \tag{1.20}$$

1. Click the movie button to see the phase development over time. Can you make this observation?

Check Your Progress 2

In some regions the phase contrast is changing linearly with time (blue arrows, the white matter tract and gyrus are more pronounced in later echoes), but not in regions close to the prefrontal cortex and temporal lobes. In those regions, more and more replications of the zebra-line pattern (which represents phase jumps) appeared in the later echoes.

Back to *Exercise 1*.

2. Stop the movie and press `Ctrl+T` to see the phase accumulation at those problematic regions (e.g. near the prefrontal cortex [113 195 65]). Can you identify the cause of the problem?

A. Somebody screwed up the acquisition

Unfortunately, all phase images of mGRE data have a similar problem. So it is unlikely that everyone screwed their acquisition up.

Back to *Exercise 1*.

B. The subject moved during the scan

Subject motion does induce changes in the phase images but certainly not responsible for the phase changes across echoes (each echo is separated by 4ms only!). Therefore, it is not the reason we have this observation.

Back to *Exercise 1*.

C. The phase is bounded to certain values

Yes, this is the correct answer! There is nothing wrong about the signal phase generated by the tissues. However, when we sampled the signal, the phase values are constrained to the range $-\pi \leq phase < \pi$. When the true phase is outside of this range, a π will be added or subtracted to the true phase such that the *apparent* phase we observed from the data will still be inside the range.

Note: Think of an analogue clock with one hand only representing the hour. Every 12 hours the hand starts a new cycle again. If you look at the clock 13 hours after the first watch, it apparently advanced 1 hour only but the actual time difference between the two observations is 13 hours.

Back to *Exercise 1*.

D. Fast switching gradient introduced extra phase

The instability of the fast switching gradient could induce extra phase offsets to the data. However, it is not the reason we have this observation in the phase images.

Back to *Exercise 1*.

E. I don't know. I'm here to learn some fMRI analysis so just show me the answer

There is nothing wrong about the signal phase generated by the tissues. However, when we sampled the signal, the phase values are constrained to the range $-\pi \leq phase < \pi$. When the true phase is outside of this range, a π will be added or subtracted to the true phase such that the *apparent* phase we observed from the data will still be inside the range.

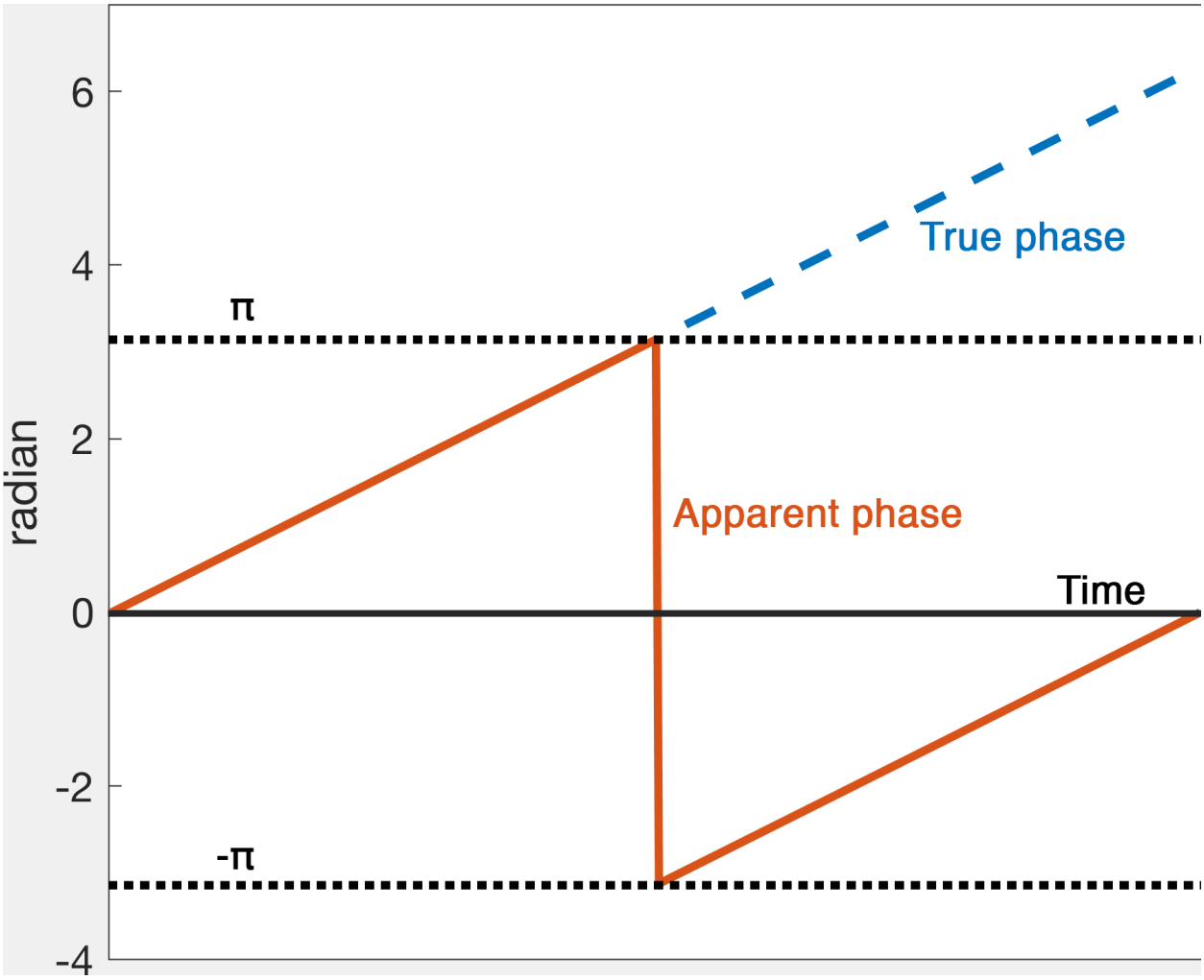


Fig. 1.10: Figure 1: An illustration of phase wrapping. The actual phase developed outside the range $-\pi \leq phase < \pi$ over time but our measurement will only show the apparent phase.

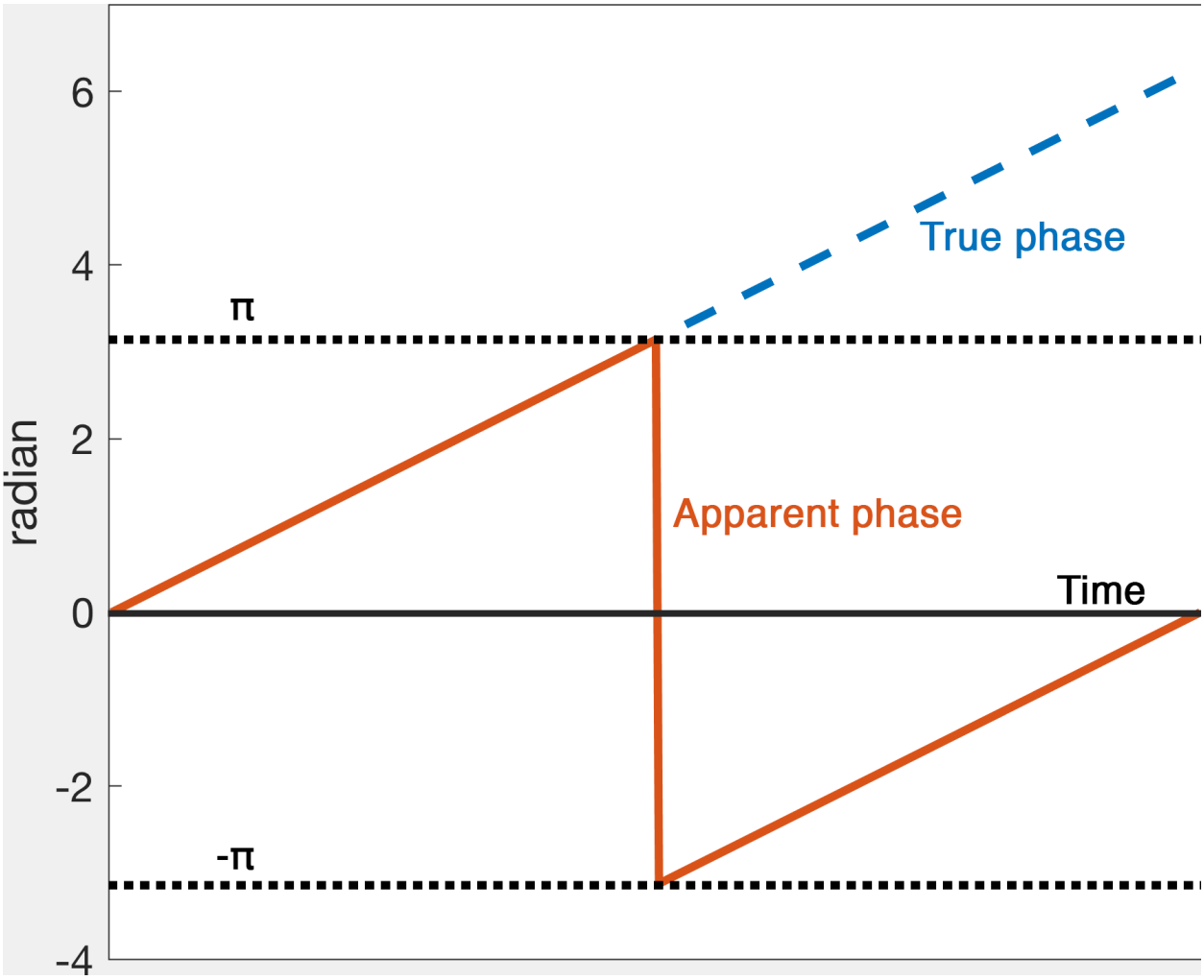


Fig. 1.11: Figure 1: An illustration of phase wrapping. The actual phase developed outside the range $-\pi \leq phase < \pi$ over time but our measurement will only show the apparent phase.

Note: Think of an analogue clock with one hand only representing the hour. Every 12 hours the hand starts a new cycle again. If you look at the clock 13 hours after the first watch, it apparently advanced 1 hour only but the actual time difference between the two observations is 13 hours.

Back to *Exercise 1*.

In order to estimate the frequency shift correctly using Eq. (1.16), this phase problem has to be addressed which is called phase unwrapping.

To unwrap the phase and to map back to the correct values, SEPIA provides several algorithms to do the job, and this is what we are going to do in the next exercise.

You can close all the FSLView window(s) now.

Proceed to *Exercise 2*.

Exercise 2

Objectives

- Gaining experience in using SEPIA
- Understanding how to perform phase unwrapping

Data Required

- a 4D raw phase data (*phase.nii.gz* in the input directory)
- a 4D raw magnitude data (*mag.nii.gz* in the input directory)
- JSON files generated by data conversion software (all .json in the input directory)

Estimated time

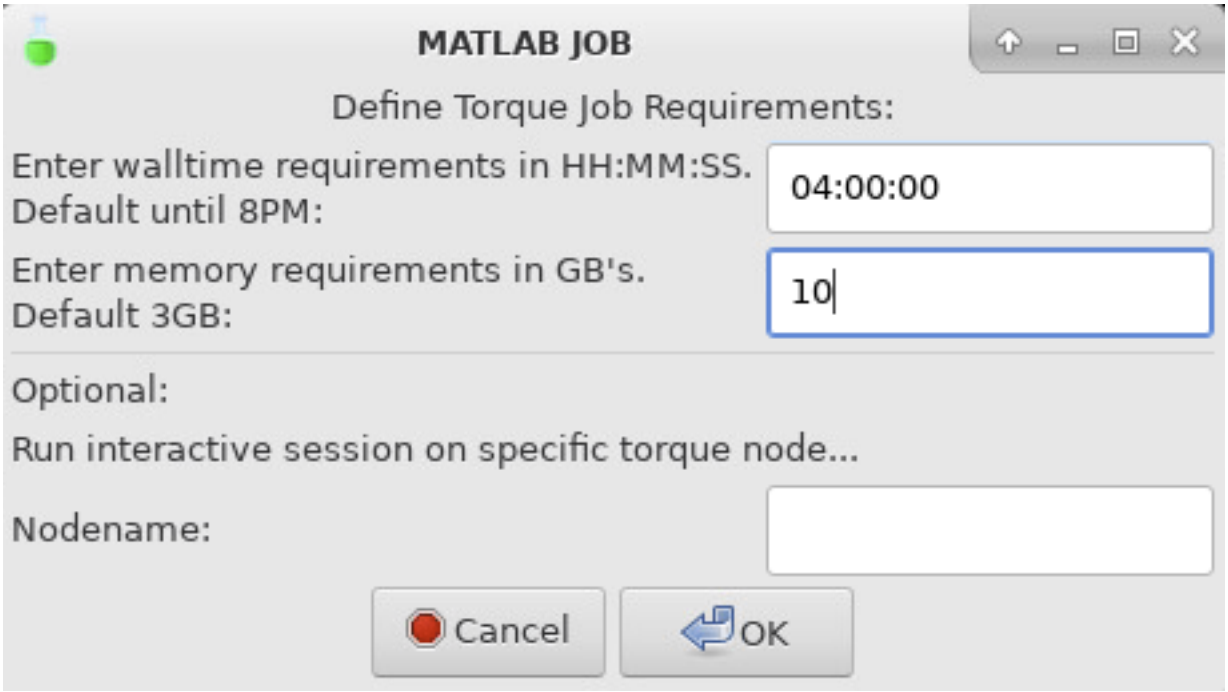
About 20 min.

SEPIA

SEPIA is a pipeline tool to process phase images in Matlab. To use SEPIA, please open a Matlab application in the cluster by typing:

`matlab2016b`,

click OK, leave the runtime as default and specify the memory requirement as 10 (GB).



Once Matlab is open, go to the tutorial directory and add the SEPIA home directory to the Matlab Path:

```
addpath('~/qsm_tutorial/sepia/');
```

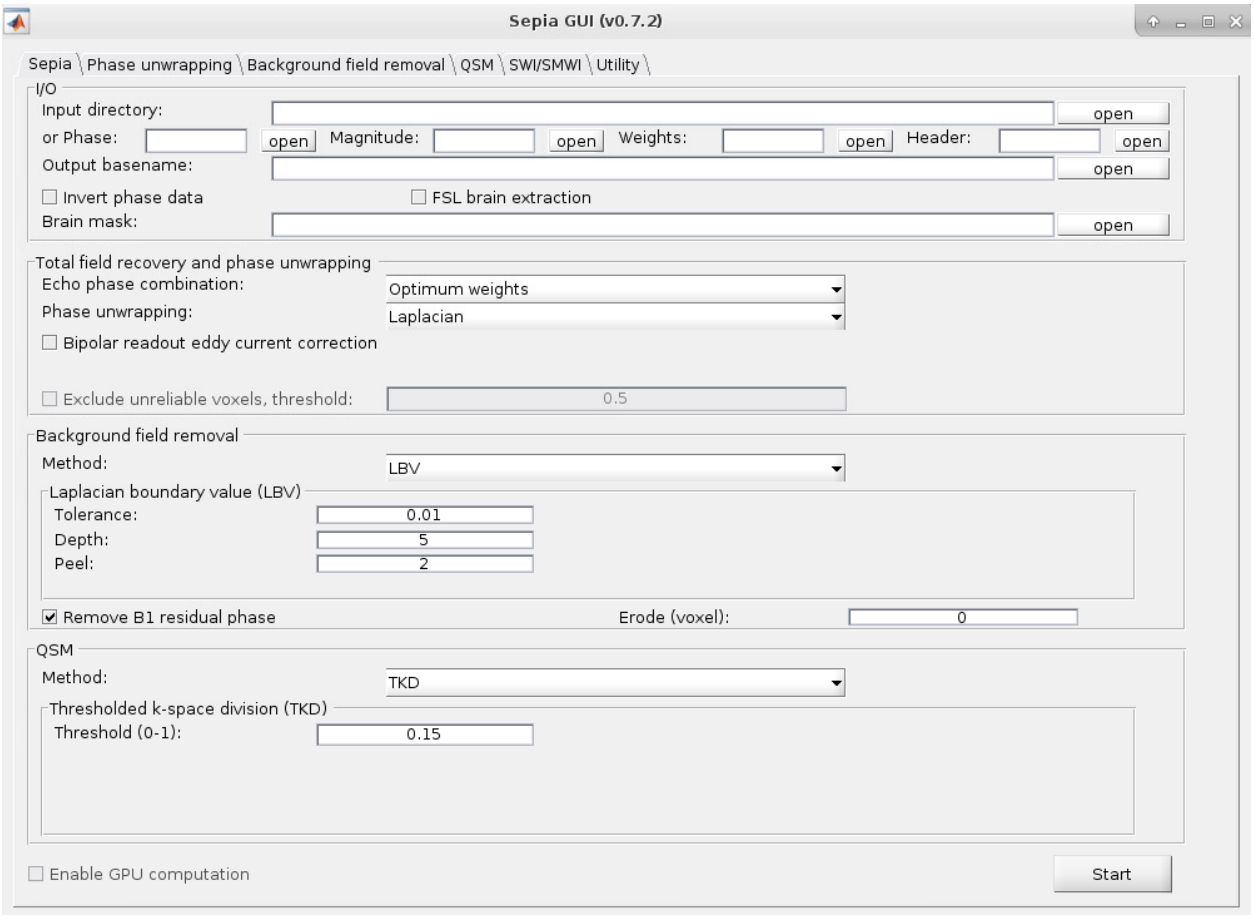
Note: The copy of SEPIA you have in the tutorial directory already includes all the external toolboxes required. If you want to know how to set up SEPIA from scratch for your research purposes, you can refer to *Installation/Setp up*.

Now, go the data directory in the Matlab command window and start sepia:

```
cd ~/qsm_tutorial/data
```

```
sepia
```

A graphical user interface (GUI) should be pop up.



The first tab in SEPIA provides a one-step application to process QSM from the raw phase data to a magnetic susceptibility map.

Alternatively, we can break down the processing pipeline into several steps and SEPIA also supports this approach.

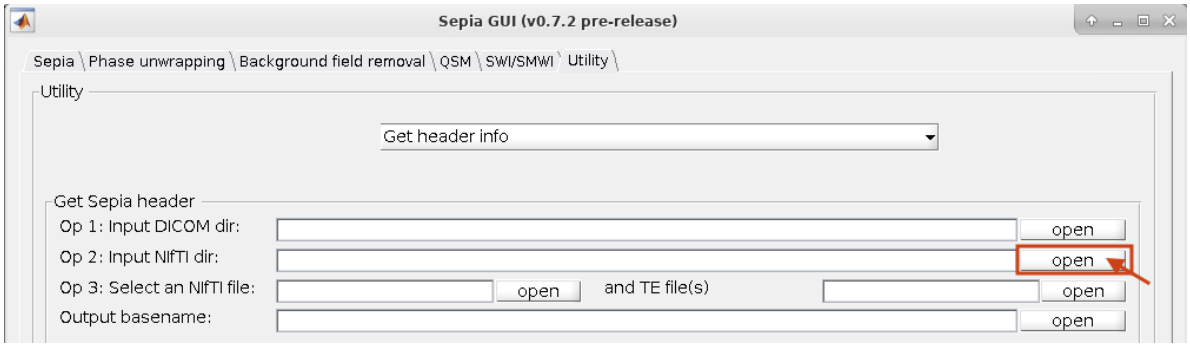
Create a SEPIA header

Before using the SEPIA, create a header file that contains all essential information regarding the data acquisition (magnetic field, resolution, echo time).

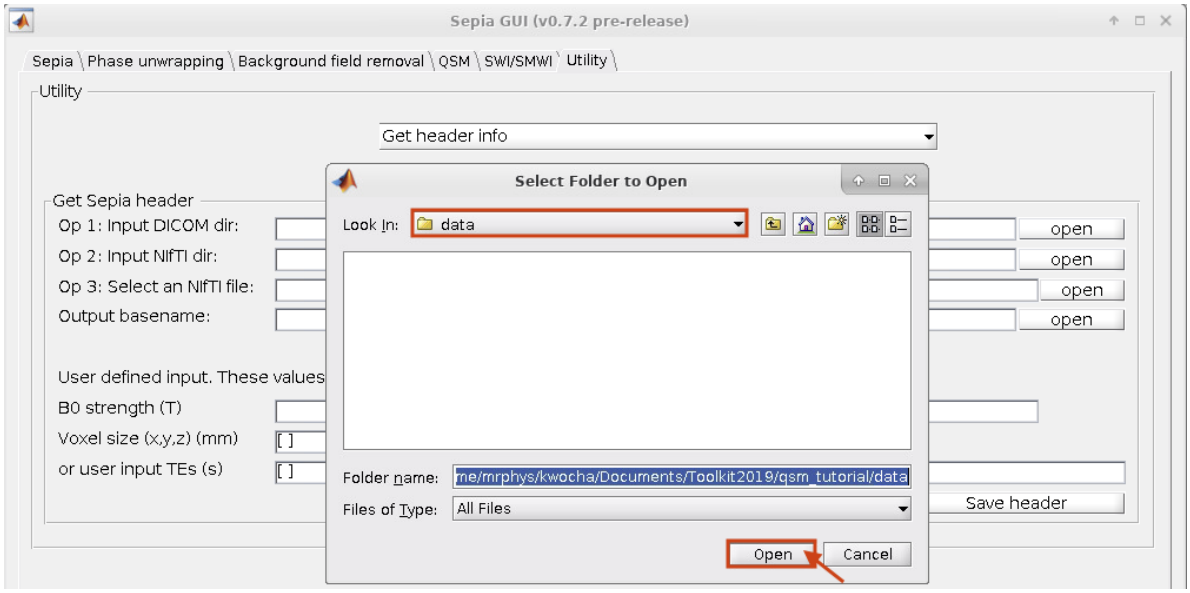
Select the **Utility** tab and then select **Get header info** in the drop-down menu. This function provides several ways to extract the header information from different files.

With all the NifTI images and JSON files stored in the same place, we can use 'Op 2' routine:

1. Click **Open** next to 'Op 2'



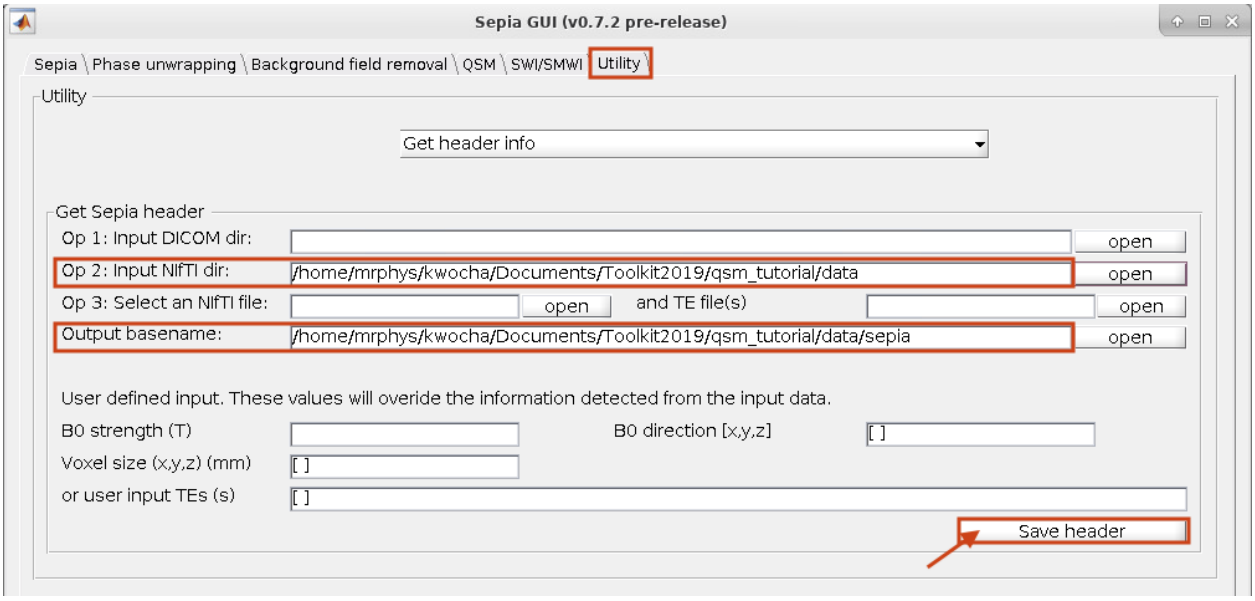
2. Select *~/qsm_tutorial/data* as the input (The dialog box will show the current directory in Matlab).



3. Click **Save header** to save the file.

   The process is done when you see the message '*SEPIA header is saved!*' in the command window. You should see a new file is generated in the input directory.

Your setting should be like:



Phase Unwrapping and Total Field Computation

To correct the wrapped phase in the raw images, go the **Phase unwrapping** tab (next to **Sepia** tab).

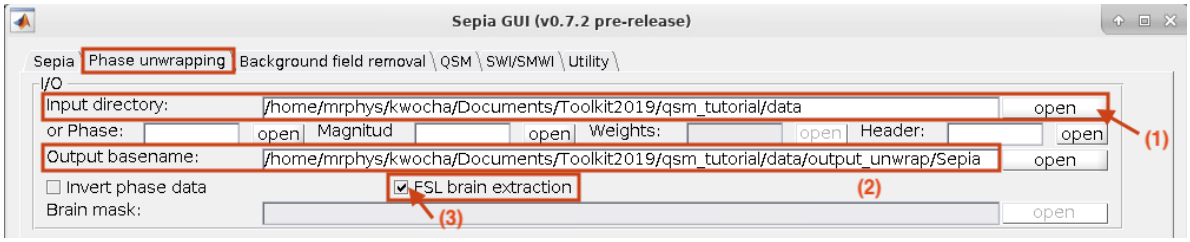
You will see two panels under the tab: the **I/O** panel is for specifying data input and output and the **Total field recovery and phase unwrapping** panel is for selecting phase unwrapping and true phase estimation algorithms.

Tip: SEPIA supports two types of data input. If your data follows the SEPIA naming structure, you can select the directory containing all the input data as your input in the first row of **I/O** panel. Alternatively, you can specify the input files separately by following the instruction of the second row of the **I/O** panel.

In the **I/O** panel:

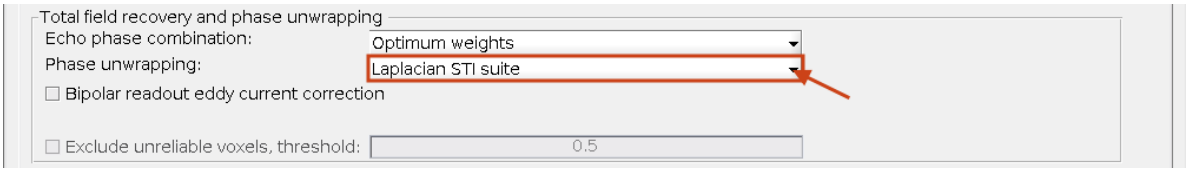
1. Select the **Input directory**: *~/qsm_tutorial/data*
2. Change the **Output basename** to: *~/qsm_tutorial/data/output_unwrap/Sepia*
3. Check the **FSL brain extraction**

   It is essential to have a brain mask to produce a high-quality QSM map.



In the **Total field recovery and phase unwrapping** panel:

1. Keep the **Echo phase combination** method as 'Optimum weights'
2. Change the **Phase unwrapping** method to 'Laplacian STI suite'.



Then click the **Start** button.

You should now see some messages displayed in the Matlab's command window. These messages give you the general information of your input data and the overview of the selected method(s). Once the process finishes (~3min), you will see the message

'*Processing pipeline is completed!*'.

Tip: All the output messages of SEPIA will be displayed on the Matlab command window. Make sure you check the command window before clicking the **Start** button again!

Check the output (should be in *~/qsm_tutorial/data/output_unwrap/*), in the terminal type:

```
fslview_deprecated Sepia_unwrapped-phase.nii.gz
```

```
fslview_deprecated Sepia_total-field.nii.gz
```

The first dataset is the unwrapped phase images (unit in radian). Play the movie to see the phase development. All the zebra-line pattern and phase jumps are gone in the later echo images (e.g. near the prefrontal cortex [113 195 65]).

Note: Besides the ability of phase unwrapping, Laplacian based operation removes some harmonic fields. Therefore, the phase values in the unwrapped phase map cannot be comparable to the raw wrapped phase.

The second dataset corresponds to the frequency (Hz) map which was computed using the unwrapped phase images at the different echo times illustrated in Eq. (1.17):

$$frequency = \frac{phase}{time} \tag{1.21}$$

The latter is the result needed in the next exercise.

You can close all the FSLView window(s) now.

Proceed to *Exercise 3*.

Back to *Exercise 1*.

Exercise 3

Objectives

- Understanding why we need to remove the background magnetic field contributions before QSM
- Fine-tuning method parameters to improve the background field removal results

Data Required

- a 3D total field image (*Sepia_total-field.nii.gz* in the previous exercise output directory)
- a 3D brain mask (*Sepia_mask.nii.gz* in the previous exercise output directory)

- a SEPIA header (*Sepia_header.mat* in the input directory)

Estimated time

About 10 min.

Background Field Removal

The frequency map we obtained from the last exercise contains magnetic fields generated by not only the brain tissue but also the scanner hardware imperfection and fields generated at air/tissue interfaces. Therefore, we have to remove the non-tissue field, or background field, contributions before computing the susceptibility map.

One more step before computing QSM but why?

In the first exercise, display window had to be adjusted to visualize some brain structures. Why are these tissue contrasts 'hidden' in our images?

The data contains not only the phase generated by the brain tissues but also by the scanner hardware imperfection and fields generated at air/tissue interfaces such as sinuses and ear canals.

These non-tissue fields, or so-called background fields, can be one or two order(s) of magnitude stronger than the tissue fields and affects the global brain tissues.

Since we are only interested in the magnetic fields generated by the brain tissues, we have to remove these background fields before computing the QSM map. However, the background fields and tissue fields co-exist across the whole brain.

Luckily, the background fields have different mathematical properties and researchers have successfully developed various algorithms to separate the tissue fields from the background fields.

Back to *Exercise 3*.

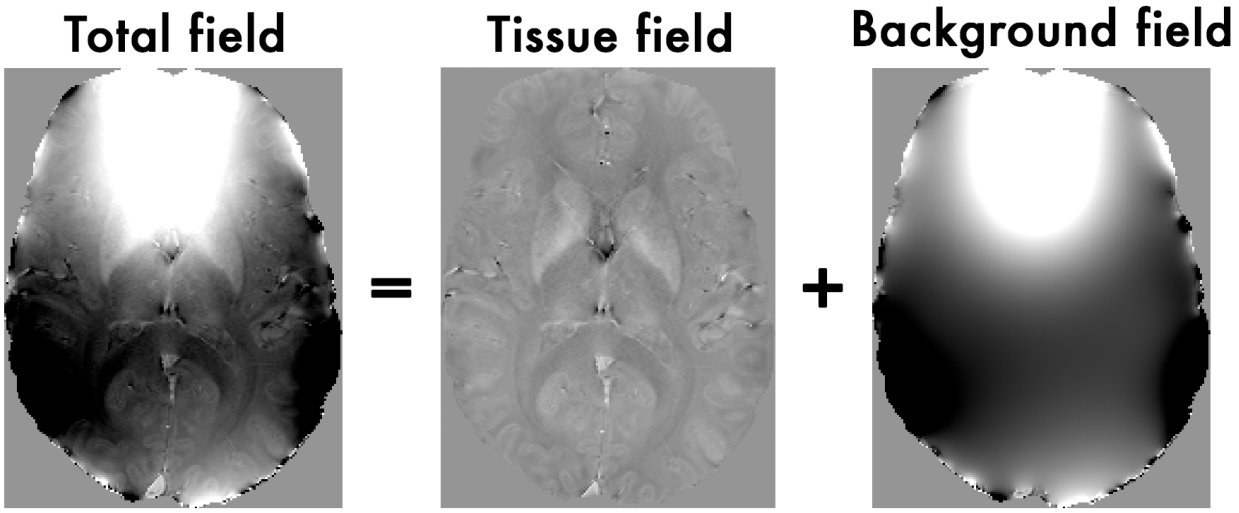


Fig. 1.12: Figure 1: The total field we obtained from the last exercise is the summation of tissue and background magnetic fields. In order to compute the magnetic susceptibility of the brain tissue correctly, the background field contributions have to be removed before mapping the tissue susceptibilities.

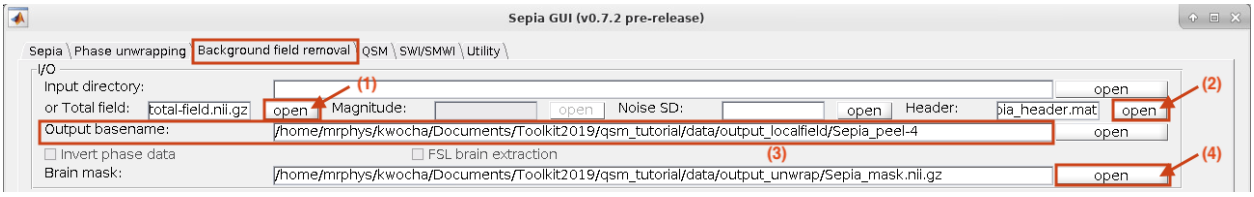
SEPIA provides 7 methods to remove the background magnetic fields. Today we will use the Laplacian boundary values (LBV) algorithm, which is, in general, quite robust.

Exercise 3.1

Go to the **Background field removal** tab. You will see two panels as in the **Phase unwrapping** tab.

In the **I/O** panel, specify the required files by using the open buttons in the second row:

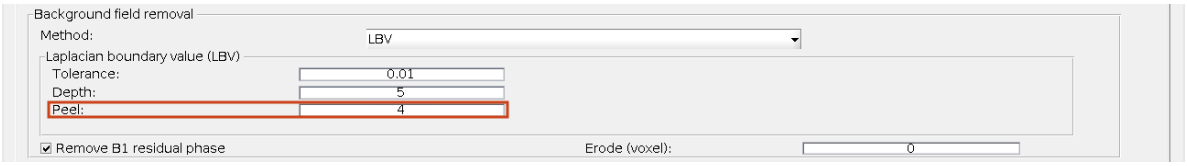
1. **Total field**: *Sepia_total-field.nii.gz* (from the output directory of the previous exercise),
2. **Header**: *Sepia_header.mat* (from the input directory),
3. Change the **Output basename** to: *~/qsm_tutorial/data/output_localfield/Sepia_peel-4*
4. **Brain mask** *Sepia_mask.nii.gz* (from the output directory of the previous exercise, which will tell the software what is not part of the background.)



Second, in the **Background field removal** panel, the 'LBV' method is shown by default. You can have three parameters to adjust.

- 'Tolerance': a threshold to stop the algorithm.
- 'Depth': multigrid level.
- 'Peel': the layer of boundary voxels to be removed after computing the tissue (or so-called local) fields.

1. Set **Peel** to: 4



2. Press the **Start** button.

   Again, when the process is finished, you will see the message: '*Processing pipeline is completed!*'.

3. Use FSLView to display the local field map (Hz) (should be in *~/qsm_tutorial/data/output_localfield/*).

   ```
   fslview_deprecated Sepia_peel-4_local-field.nii.gz &
   ```

   Adjust the display window to 'Min -5' and 'Max 5'.

Note you can clearly see the contrast between grey and white matter, and veins and tissue now. Other structures as the globus pallidus, red nuclei and substantia nigra are visible... but not quite normal.

4. Compare the location of the edges of brain structures with what you can see in the mean magnitude image computed in Exercise 1.

1. Use shortcut `Crtl+A` in the FSLView to add *mean_head.nii.gz* to the displayed local field maps.
2. Adjust the display window of *mean_head.nii.gz* to 'Min 0' and 'Max 300'.
3. Go to location [133 155 81] which is on the top edge of the globus pallidus.

4. Check/Uncheck the 'Visibility' button to turn on/off of the mean magnitude image.

5. It is known that there is a high concentration of iron deposition in the globus pallidus. There it generated a strong secondary magnetic field. Can you identify the magnetic field generated by this structure?

You can close all the FSLView window(s) now.

Proceed to *Exercise 4*.

Advanced Exercise 3.2

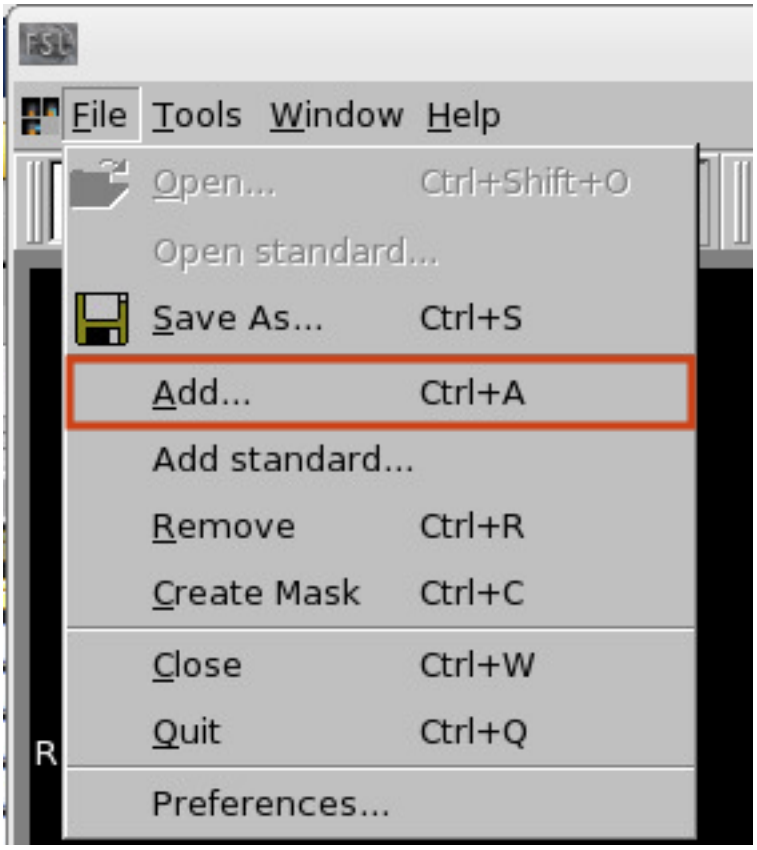
Note: If you still have enough time, follow the exercise below.

In this exercise, we will focus on the differences of using different 'Peel' values.

1. Change the **Output basename** to: *~/qsm_tutorial/data/output_localfield/Sepia_peel-2*
2. Set **Peel** to: 2
3. Press the **Start** button. Again, when the process is finished, you will see the message: '*Processing pipeline is completed!*'.
4. Use FSLView to display the local field map (Hz) when using 'Peel' value of 2 (should be in *~/qsm_tutorial/data/output_localfield/*):

   ```
   fslview_deprecated Sepia_peel-2_local-field.nii.gz &
   ```

   Load the *Sepia_peel-4_local-field.nii.gz* map in the same window. You can do this by selecting 'File Add...' (or shorcut `Crtl+A`) and select *Sepia_peel-4_local-field.nii.gz* in the window provided.



1. Adjust the display window to 'Min -5' and 'Max 5' for both images.
2. Set location as [119 142 88]. Look at the differences between the two maps globally.
3. Check/Uncheck the 'Visibility' button in the bottom of the viewer to turn on/off of the top image. What are the main differences between the two results?

Proceed to *Exercise 4*.

Back to *Exercise 2*.

Exercise 4

Objectives

- Understanding QSM dipole inversion
- Gaining experience to use QSM algorithms

Data Required

- a 3D local field image (*Sepia_peel-4_local-field.nii.gz* in the previous exercise output directory)
- a 3D refined brain mask (*Sepia_peel-4_mask-qsm.nii.gz* in the previous exercise output directory)
- a SEPIA header (*Sepia_header.mat* in the input directory)

Estimated time

About 15 min.

The Last Step

The last step of QSM processing is to deconvolute the local (tissue) field by the unit dipole field, such that the tissue magnetic susceptibility can be revealed. This can be described by:

$$\chi = F^{-1}[\frac{F(Tissuefield)}{F(d)}] \tag{1.22}$$

where F and F^{-1} are the Fourier and inverse Fourier transform operators.

This is the so-called dipole inversion of QSM, which is just the element-wise division between the Fourier transforms of the two images.

Theory: Dipole Inversion

When looking at the local-field you can see some beautiful contrasts between brain tissues. These local fields represent all the secondary magnetic field induced by the tissues (which are our magnetic susceptibility sources). The tissue is suddenly behaving like a small magnet and the effect of the magnetic field extends beyond the source itself.

Note: Think of a magnet that can attract a metal towards it even when the two objects are not intact. It is because the magnetic field generated by the magnet extends outside the magnet body itself.

The local fields generated by the magnetic susceptibility sources (χ), can be computed using the following equation:

$$Tissuefield = \chi * d \tag{1.23}$$

where d is a unit (dipole) field that the source generated and has the following shape:

Eq. (1.19) basically means that the secondary magnetic field experienced by the tissue at each location is the summation of the fields generated by all other (surrounding) sources. Since we have the prior knowledge about the shape of the magnetic field generated by a unit source (which is d , the unit dipole field) and the field generated by the tissues (output of exercise 3, *local-field.nii.gz*), mapping the magnetic susceptibility of the tissue is just the deconvolution of these two spatial maps (a.k.a. dipole inversion).

Back to *Exercise 4*.

Sounds simple, isn't it? Let's try it out!

QSM: Dipole inversion

Move to the **QSM** tab of SEPIA.

Exercise 4.1

I/O panel:

1. Select the **Local field** input: *Sepia_peel-4_local-field.nii.gz* (from the previous exercise output directory),

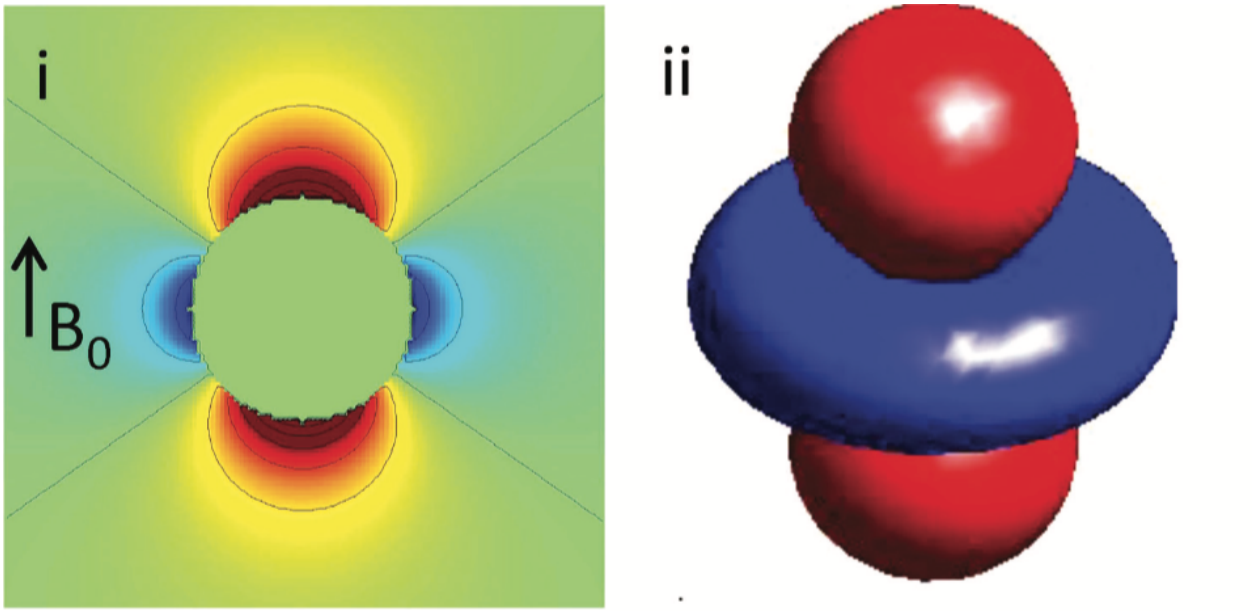
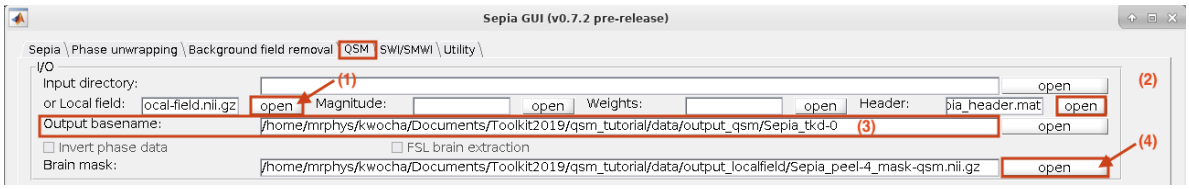


Fig. 1.13: Figure 1: An illustration of a unit dipole field in (i) sagittal section and (ii) surface rendered contour. Red colour represents a positive magnetic field and blue colour represents a negative magnetic field. (Reproduced from Wang & Liu MRM 2014, Wiley)

2. Select the **Header**: *Sepia_header.mat*,
3. Change the **Output basename** of the output to: *~/qsm_tutorial/data/output_qsm/Sepia_tkd-0*,
4. Select the **Brain mask**: *Sepia_peel-4_mask-qsm.nii.gz* (from the previous exercise output directory).

Note: An updated brain mask has to be used here because some edges were excluded from the original brain mask in the last operation.



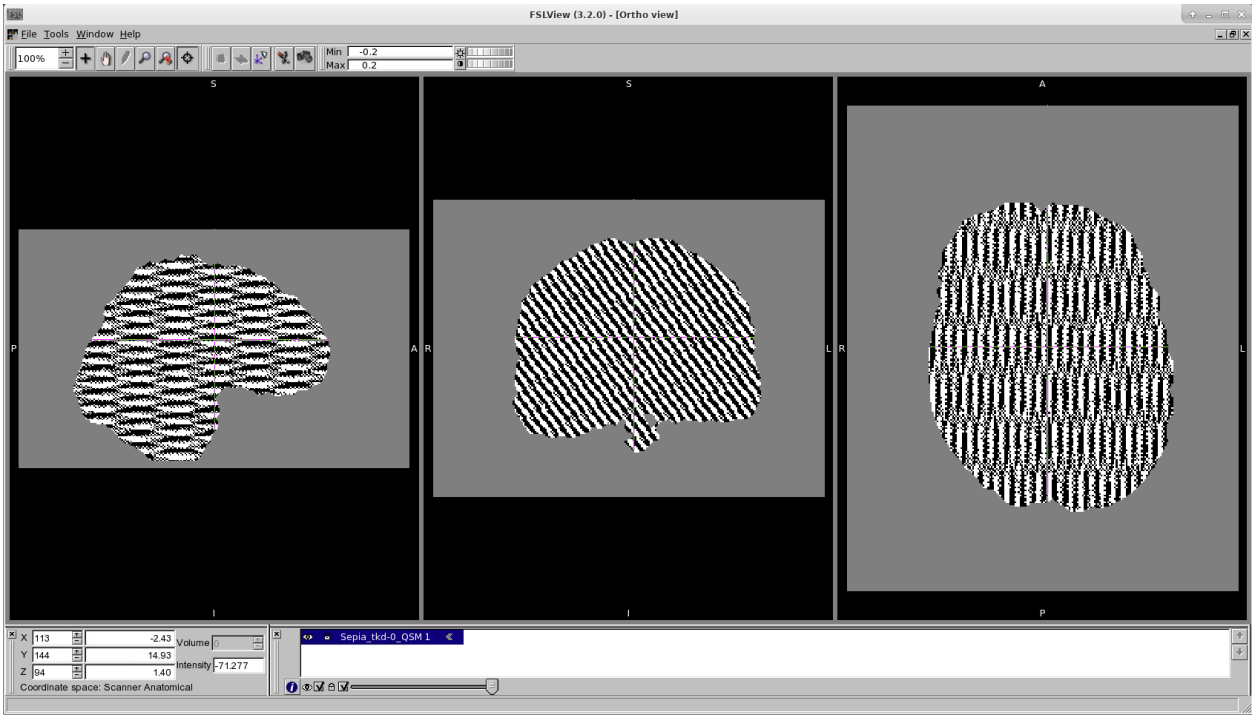
QSM panel:

5. To do exactly the operation as in Eq. (1.18), set the threshold of the **TKD** algorithm to '0' and press **Start**.

6. Check the result *Sepia_tkd-0_QSM.nii.gz* in the output directory. Set the display window to 'Min. -0.2' and 'Max. 0.2' (ppm). Does it look like the QSM map as we expected?

Answer: Exercise 4.1

The result should look like the image below. We followed exactly the operation described in the dipole inversion equation, but what's wrong?



To understand the problem, let's have a look at the denominator of the dipole inversion equation, which is the Fourier transform of the dipole field (so-called dipole kernel).

$$\chi = F^{-1}[\frac{F(Tissuefield)}{F(d)}]$$

$$Dipolekernel, D = F(d)$$

Here is the image representation of the magnitude of the dipole kernel $|D|$

The dashed lines outlines the regions where the values in the dipole kernel are equal or very close to zero (which can be represented as a cone surface). This leads to the ill-defined division of zero problem! After dividing the two images, these regions will contain either undefined value (when values in dipole kernel equal to zeros) or unrealistic numbers (when values close to zeros), making the QSM map unusable:

Figure 2: Fourier transform of the QSM map (i.e. in k-space). It is clear that the values on/near the cone surface in k-space are amplified.

Back to *Exercise 4*.

Exercise 4.2

To avoid the previous QSM map we can increase the threshold of the **TKD**.

1. Change the **Output basename** to: *~/qsm_tutorial/data/output_qsm/Sepia_tkd-0p15*.
2. Change the threshold of the **TKD** algorithm to 0.15 and press **Start**.

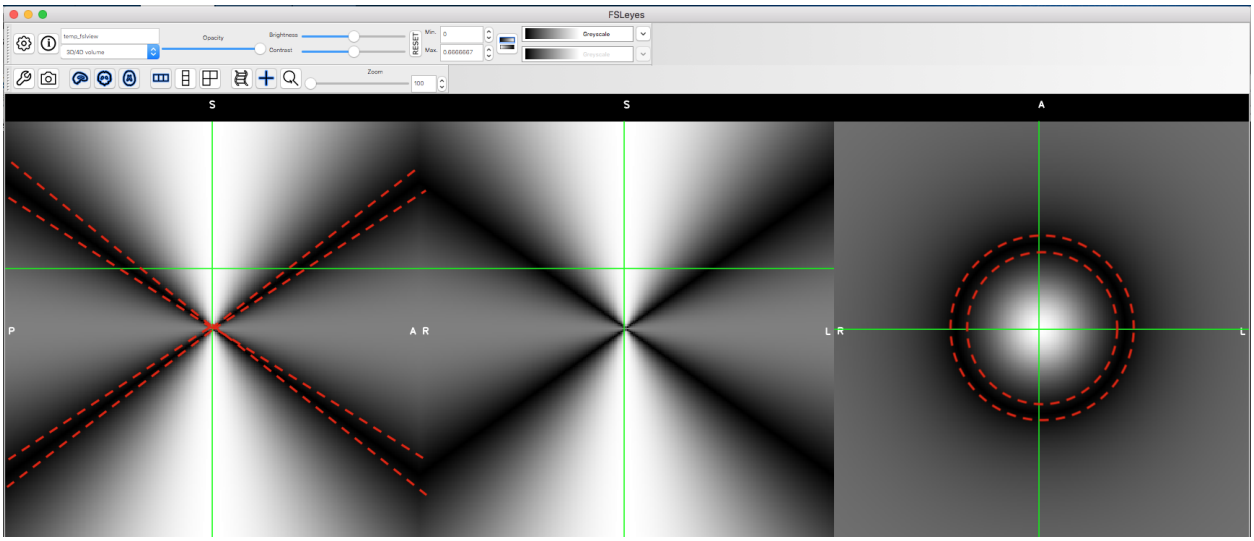
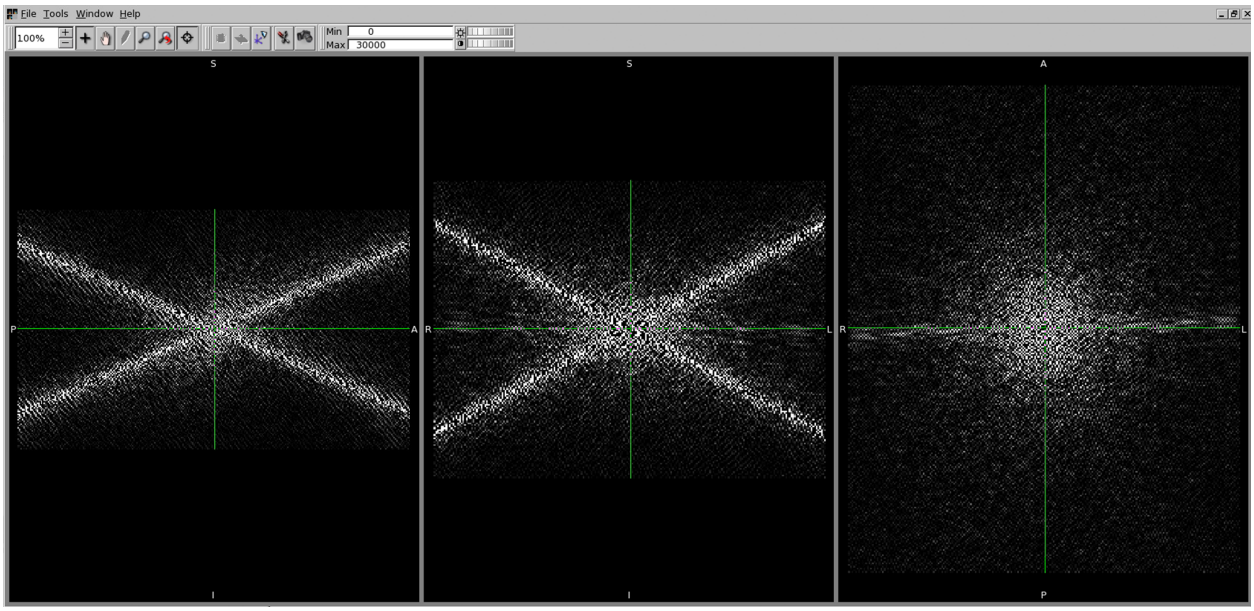


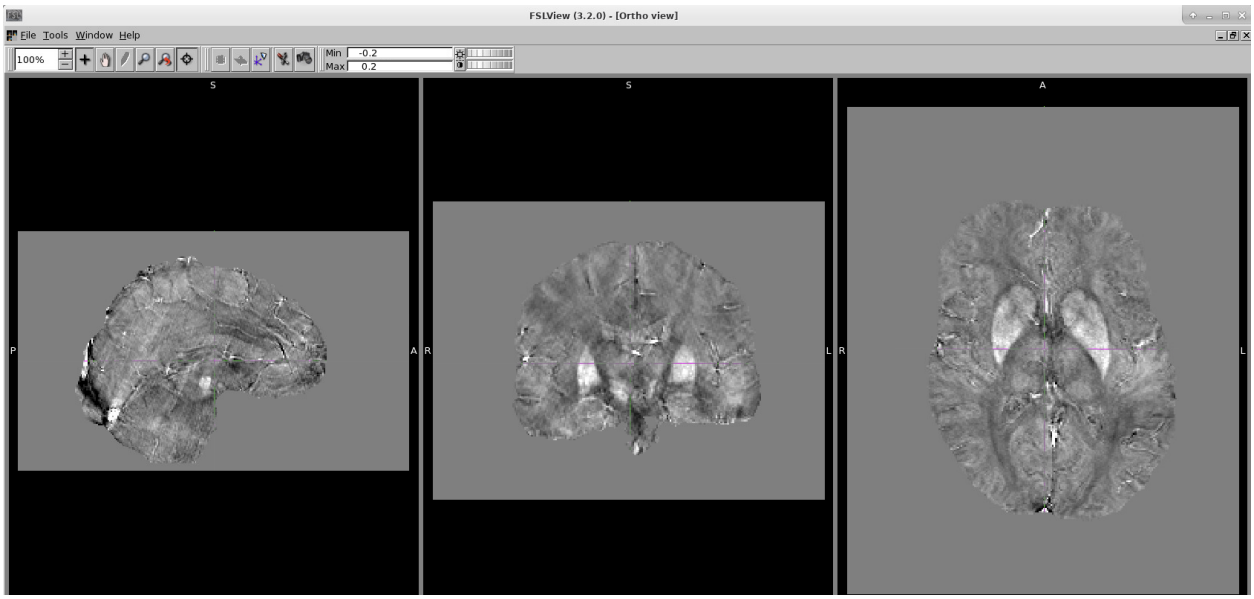
Fig. 1.14: Figure 1: The magnitude of the Fourier transform of the dipole field.



3. Check the result *Sepia_tkd-0p15_QSM.nii.gz* in the output directory. Display it along with the *Sepia_tkd-0_QSM.nii.gz* in FSLView. Do you see any improvement?
4. Compare the location on the tissue edges (e.g. [133 155 81]) in this QSM map with what you can see in the mean magnitude image *mean_head.nii.gz*. Do the edges match between the two data now?

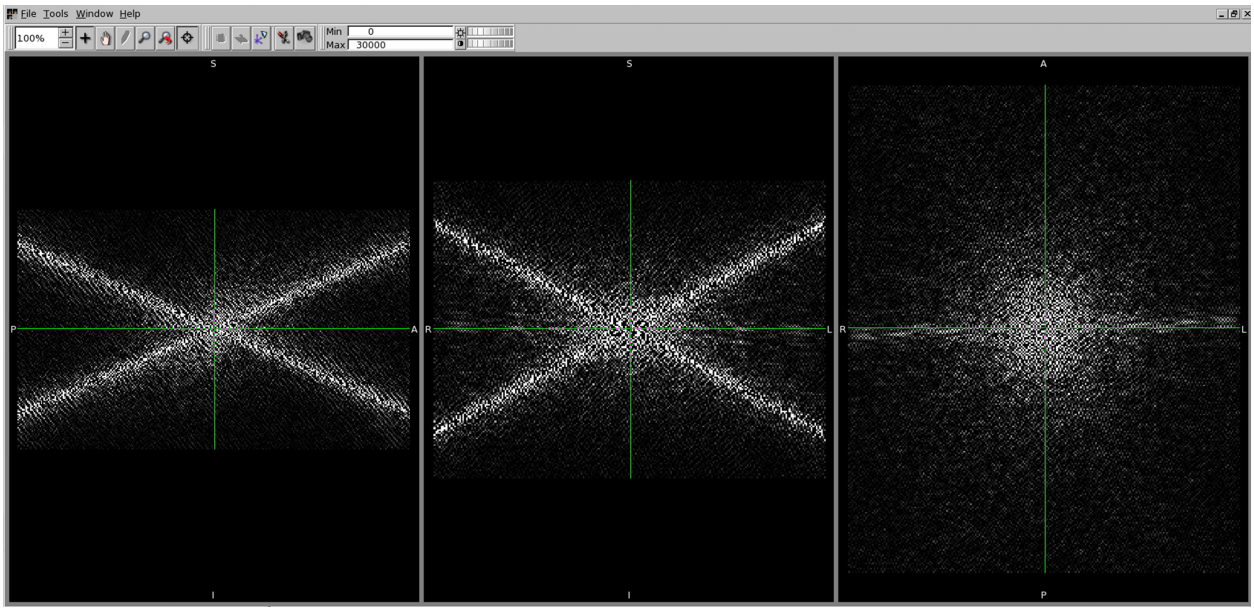
Answer: Exercise 4.2

You should now see some brain structures in the QSM map.

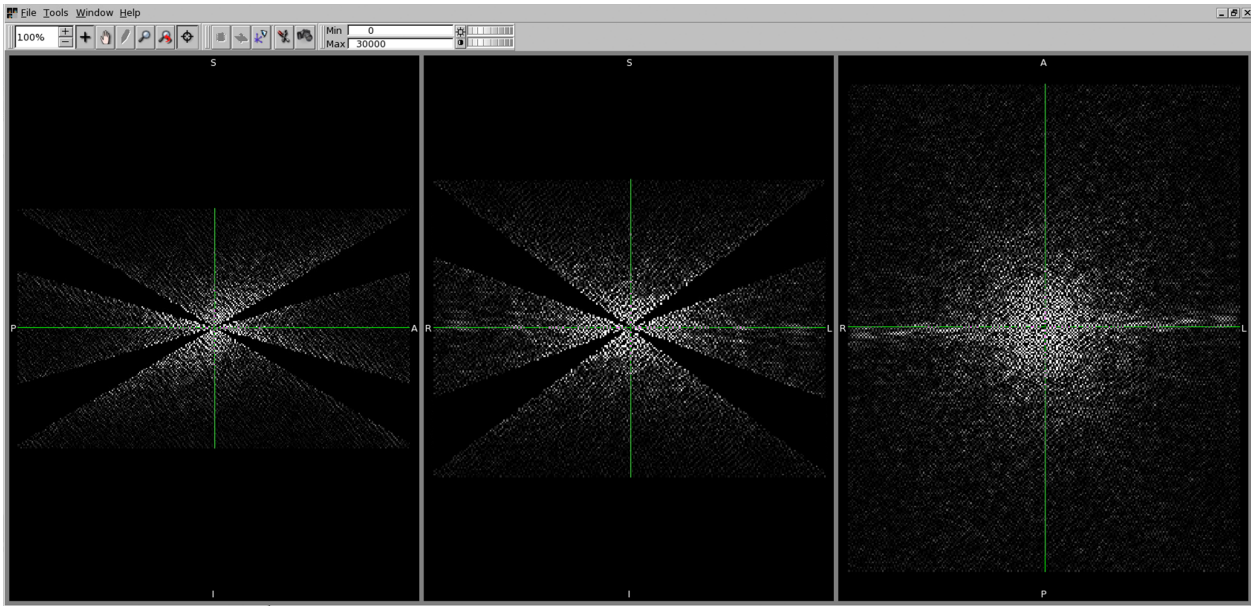


The idea of the TKD method is very straightforward. Since we know the location in which the division results are unreliable (i.e. when D equal/close to zero), we can discard the information in these regions by replacing their values to zero after the element-wise division.

For example, without thresholding the QSM k-space:



and we can threshold the above k-space when the magnitude of D is smaller than, e.g. 0.15, leading to



The Fourier transform of the above image is the QSM map we obtained in this exercise.

Tip: However, the larger is the threshold value, the more is the information being discarded. Therefore, increasing the threshold can improve the appearance of the resulting QSM map (as artefacts reduced), we are also losing the accuracy between our input (i.e. local field map) and output (i.e. QSM map).

Back to *Exercise 4*.

Congratulations! You have finished all the exercises in this tutorial. If you still have time left, finish the advanced exercises or experiment with different QSM methods/methods' parameters.

Advanced Exercise 4.3

To further improve the quality of the QSM map, some methods, such as Star-QSM, incorporate additional information, e.g. smoothness of the QSM map, during the QSM dipole inversion.

1. Change the **Output basename** to: *~/qsm_tutorial/data/output_qsm/Sepia_star-qsm*.
2. Change the **QSM** method to 'Star-QSM' and press **Start**. It will take about 2 mins to finish.

It is difficult to see the differences between the two QSM maps with the naked eyes. Subtract the two maps so that you can see the differences:

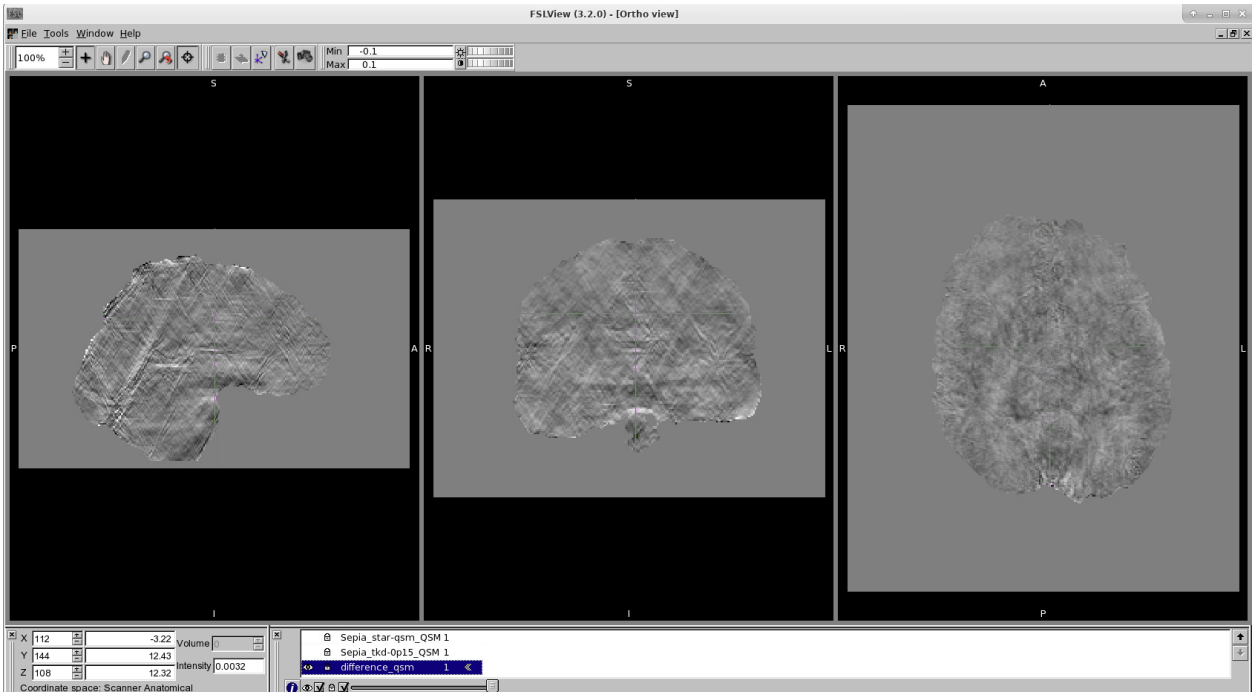
```
fslmaths Sepia_tkd-0p15_QSM.nii.gz -sub Sepia_star-qsm_QSM.nii.gz
difference_qsm
```

3. Display the *difference_qsm.nii.gz* image (dispay window [-0.1,0.1]) in FSLView with *Sepia_star-qsm_QSM.nii.gz* (dispay window [-0.2,0.2]) and *Sepia_tkd-0p15_QSM.nii.gz* (dispay window [-0.2,0.2]) in the output directory. Can you see any difference between the two QSM maps?

Answer: Exercise 4.3

You should be able to see the QSM map produced by Star-QSM is less noisy compared to the TKD result.

When looking to the difference map (see below), you can see that the main difference is the so-called more pronounced streaking artefact present in the TKD QSM map (which can be identified in the sagittal and coronal views).



Back to *Exercise 4*.

Back to *Exercise 3*.

1.23 Acknowledgments

The methods and algorithms provided with **SEPIA** are contributed by:

MEDI Toolbox developed by

Weill Cornell MRI lab

- DICOM reader
- Matlab interface of using FSL's brain extraction
- Combination of echo phase using non-linear fitting
- Laplacian phase unwrapping
- Region growing phase unwrapping
- Graphcut phase unwrapping
- Laplacian boundary value (LBV) background field removal
- Projection onto dipole field (PDF) background field removal
- Morphology enabled dipole inversion QSM
- Lateral ventricle masking

STI Suite developed by

Hongjiang Wei, PhD - University of California Berkeley, CA, USA

Wei Li, PhD, - University Texas Health Science Center at San Antonio Research Imaging Institute, San Antonio, TX, USA

Chunlei Liu, PhD - University of California Berkeley, CA, USA

- Laplacian phase unwrapping
- VSHARP background field removal
- iHARPERELLA background field removal
- iLSQR QSM
- Star-QSM

FANSI Toolbox developed by

Carlos Milovic, PhD - Biomedical Imaging Center at Pontificia Universidad Católica de Chile

Berkin Bilgic, PhD - Martinos Center for Biomedical Imaging, Harvard Medical School, MA, USA

Bo Zhao, PhD - Martinos Center for Biomedical Imaging, Harvard Medical School, MA, USA

Julio Acosta-Cabronero, PhD - Wellcome Trust Centre for Neuroimaging, Institute of Neurology, University College London, London, UK, and German Center for Neurodegenerative Diseases (DZNE), Magdeburg, Germany

Cristian Tejos, PhD - Department of Electrical Engineering, Pontificia Universidad Catolica de Chile, Santiago, Chile and the Biomedical Imaging Center at Pontificia Universidad Católica de Chile

- FANSI QSM (including both linear and non-linear solvers, and TV and TGV regularisation options)

SEGUE developed by

Anita Karsa, PhD - University College London, UK Karin Shmueli, PhD - University College London, UK

MRI Susceptibility Calculation Methods developed by

Karin Shmueli's group in UCL, UK

ROMEO developed by

ROMEO development team

1.24 References

When you use **SEPIA** in your research, please cite the method(s) that you used:

1.24.1 Brain extraction

FSL bet Smith, S. M. Fast robust automated brain extraction. Hum. Brain Mapp. 17, 143–155 (2002).

1.24.2 Phase unwrapping

Laplacian-based method Schofield, M. A. & Zhu, Y. Fast phase unwrapping algorithm for interferometric applications. Opt Lett 28, 1194–1196 (2003).

Li, W., Wu, B. & Liu, C. Quantitative susceptibility mapping of human brain reflects spatial variation in tissue composition. Neuroimage 55, 1645–1656 (2011).

3D best path Abdul-Rahman, H. S. et al. Fast and robust three-dimensional best path phase unwrapping algorithm. Applied Optics 46, 6623–6635 (2007).

Graphcut Dong, J. et al. Simultaneous phase unwrapping and removal of chemical shift (SPURS) using graph cuts: application in quantitative susceptibility mapping. IEEE Transactions on Medical Imaging 34, 531–540 (2015).

SEGUE Karsa and Shmueli. SEGUE: A Speedy rEgion-Growing Algorithm for Unwrapping Estimated Phase. IEEE Transactions on Medical Imaging 38, 1347-1357 (2018).

Echo phase combination - Optimum weights Robinson, S. D. et al. An illustrated comparison of processing methods for MR phase imaging and QSM: combining array coil signals and phase unwrapping. NMR Biomed 30, e3601 (2017).

Echo phase combination - MEDI nonlinear fit Liu, T. et al. Nonlinear formulation of the magnetic field to source relationship for robust quantitative susceptibility mapping. Magn Reson Med 69, 467–476 (2012).

ROMEO ROMEO Dymerska, B., Eckstein, K., Bachrata, B., Siow, B., Trattnig, S., Shmueli, K., Robinson, S.D., 2020. Phase Unwrapping with a Rapid Opensource Minimum Spanning TreE AlgOrithm (ROMEO). Magnetic Resonance in Medicine.

MCPC-3D-S Coil Combination: Eckstein, K., Dymerska, B., Bachrata, B., Bogner, W., Poljanc, K., Trattnig, S., Robinson, S.D., 2018. Computationally Efficient Combination of Multi-channel Phase Data From Multi-echo Acquisitions (ASPIRE). Magnetic Resonance in Medicine 79, 2996–3006.

1.24.3 Background field removal

LBV Zhou, D., Liu, T., Spincemaille, P. & Wang, Y. Background field removal by solving the Laplacian boundary value problem. NMR Biomed 27, 312–319 (2014).

PDF Liu, T. et al. A novel background field removal method for MRI using projection onto dipole fields (PDF). NMR Biomed 24, 1129–1136 (2011).

RESHARP Sun, H. & Wilman, A. H. Background field removal using spherical mean value filtering and Tikhonov regularization. Magn Reson Med 71, 1151–1157 (2014).

SHARP Schweser, F., Deistung, A., Lehr, B. W. & Reichenbach, J. R. Quantitative imaging of intrinsic magnetic tissue properties using MRI signal phase: an approach to in vivo brain iron metabolism? Neuroimage 54, 2789–2807 (2011).

VSHARP Li, W., Wu, B. & Liu, C. Quantitative susceptibility mapping of human brain reflects spatial variation in tissue composition. Neuroimage 55, 1645–1656 (2011).

iHARPERELLA Li, W., Avram, A. V., Wu, B., Xiao, X. & Liu, C. Integrated Laplacian-based phase unwrapping and background phase removal for quantitative susceptibility mapping. NMR Biomed 27, 219–227 (2014).

BFRnet Xuanyu Zhu, Yang Gao, Feng Liu, Stuart Crozier, Hongfu Sun, 2022. BFRnet: A deep learning-based MR background field removal method for QSM of the brain containing significant pathological susceptibility sources

1.24.4 QSM

TKD Wharton, S., Schäfer, A. & Bowtell, R. Susceptibility mapping in the human brain using threshold-based k-space division. Magn Reson Med 63, 1292–1304 (2010).

Shmueli, K. et al. Magnetic susceptibility mapping of brain tissue in vivo using MRI phase data. Magn Reson Med 62, 1510–1522 (2009).

Closed-form solution Bilgic, B. et al. Fast image reconstruction with L2-regularization. J Magn Reson Imaging 40, 181–191 (2014).

LSQR Li, W., Wu, B. & Liu, C. Quantitative susceptibility mapping of human brain reflects spatial variation in tissue composition. Neuroimage 55, 1645–1656 (2011).

Star-QSM Wei, H. et al. Streaking artifact reduction for quantitative susceptibility mapping of sources with large dynamic range. NMR Biomed 28, 1294–1303 (2015).

Wei, H. et al. Imaging whole-brain cytoarchitecture of mouse with MRI-based quantitative susceptibility mapping. Neuroimage 137, 107–115 (2016).

Wei, H. et al. Investigating magnetic susceptibility of human knee joint at 7 Tesla. Magn Reson Med 78, 1933–1943 (2017).

FANSI Milovic, C., Bilgic, B., Zhao, B., Acosta-Cabronero, J. & Tejos, C. Fast nonlinear susceptibility inversion with variational regularization. Magn Reson Med 80, 814–821 (2018).

Bilgic, B. et al. Fast quantitative susceptibility mapping with L1-regularization and automatic parameter selection. Magn Reson Med 72, 1444–1459 (2014).

Bilgic, B., Chatnuntawech, I., Langkammer, C. & Setsompop, K. Sparse methods for Quantitative Susceptibility Mapping. in (eds. Papadakis, M., Goyal, V. K. & Van De Ville, D.) 9597, 959711 (SPIE, 2015).

MEDI Liu, T. et al. Morphology enabled dipole inversion (MEDI) from a single-angle acquisition: Comparison with COSMOS in human brain imaging. Magn Reson Med 66, 777–783 (2011).

Liu, J. et al. Morphology enabled dipole inversion for quantitative susceptibility mapping using structural consistency between the magnitude image and the susceptibility map. Neuroimage 59, 2560–2568 (2012).

Liu, Z., Spincemaille, P., Yao, Y., Zhang, Y. & Wang, Y. MEDI+0: Morphology enabled dipole inversion with automatic uniform cerebrospinal fluid zero reference for quantitative susceptibility mapping. Magn Reson Med 79, 2795–2803 (2018).

NDI Polak D., Chatnuntawech I., Yoon J., Srinivasan Iyer S., Lee J., Setsompop K., and Bilgic B. VaNDI: Variational Nonlinear Dipole Inversion enables QSM without free parameters (program number 0319). Proceedings of the International Society for Magnetic Resonance in Medicine 2019, Montreal Canada

MRI Susceptibility Calculation Methods For the TKD software implementation, the following citation shall be included in the acknowledgements: Shmueli, K et al. (2009). Magnetic susceptibility mapping of brain tissue in vivo using MRI phase data, Magnetic Resonance in Medicine vol 62 issue 6, 1510-1522 and Schweser, F et al. (2013). Toward online reconstruction of quantitative susceptibility maps: superfast dipole inversion, Magnetic Resonance in Medicine vol 69 issue 6, 1581-1593.

For the dirTik and iterTik software implementations in the package, the following citation shall be included in the acknowledgements: Karsa, A et al. (2019). High Repeatability of Quantitative Susceptibility Mapping (QSM) in the Head and Neck With a View to Detecting Hypoxic Cancer Sites, In Proceedings of the 27th Annual Meeting of the ISMRM, Montreal, p. 4939 and Schweser, F et al. (2013). Toward online reconstruction of quantitative susceptibility maps: superfast dipole inversion, Magnetic Resonance in Medicine vol 69 issue 6, 1581-1593.

QSMnet+ J. Yoon, E. Gong, I. Chatnuntawech, B. Bilgic, J. Lee, W. Jung, J. Ko, H. Jung, K. Setsompop, G. Zaharchuk, E.Y. Kim, J. Pauly, J. Lee. Quantitative susceptibility mapping using deep neural network: QSMnet. Neuroimage. 2018 Oct;179:199-206.

W. Jung, J. Yoon, S. Ji, J. Choi, J. Kim, Y. Nam, E. Kim, J. Lee. Exploring linearity of deep neural network trained QSM: QSMnet+. Neuroimage. 2020 May; 116619.

W. Jung, S. Bollmann, J. Lee. Overview of quantitative susceptibility mapping using deep learning: Current status, challenges and opportunities. NMR in Biomedicine. 2020 Mar; e4292.

LP-CNN Kuo-Wei Lai, Manisha Aggarwal, Peter van Zijl, Xu Li & Jeremias Sulam, 2020. Learned Proximal Networks for Quantitative Susceptibility Mapping

xQSM Yang Gao, Xuanyu Zhu, Bradford A. Moffat, Rebecca Glarin, Alan H. Wilman, G. Bruce Pike, Stuart Crozier, Feng Liu, Hongfu Sun, 2020. xQSM: quantitative susceptibility mapping with octave convolutional and noise-regularized neural networks.

1.24.5 SWI/SMWI

CLEAR-SWI Eckstein, K., Bachrata, B., Hangel, G., Widhalm, G., Enzinger, C., Barth, M., Trattnig, S., Robinson, S.D., 2021. Improved susceptibility weighted imaging at ultra-high field using bipolar multi-echo acquisition and optimized image processing: CLEAR-SWI. Neuroimage 237, 118175–118175.

SMWI Gho, S.-M., Liu, C., Li, W., Jang, U., Kim, E.Y., Hwang, D., Kim, D.-H., 2014. Susceptibility map-weighted imaging (SMWI) for neuroimaging. Magnetic resonance in medicine 72, 337–346.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search