FUGUE

<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FUGUE>

# **Research Overview**

FUGUE is, most generally, a set of tools for EPI distortion correction. It also refers to a specific command line tool fugue.

* Functional images acquired using standard **EPI sequences are distorted due to magnetic field inhomogeneities**
* Inhomogeneities are caused by magnetic susceptibility differences in neighbouring tissues within the head   
  + particularly for air/bone or air/tissue interfaces in the sinuses
* Inhomogeneities result in **geometrical distortion and signal loss**, particularly in the inferior frontal and temporal regions
* Field inhomogeneities can be measured with a **fieldmap image**   
  + measured field values can be used to calculate the geometric distortion and signal loss
  + this information can be used to *compensate* for (not completely remove) these artefacts
* Artefacts are compensated by   
  + geometrically unwarping the EPI images, and
  + applying cost-function masking in registrations to ignore areas of signal loss
* Areas where signal loss has occurred *cannot be restored* with any form of post-processing, as the signal has been lost - only different acquisition techniques (which are available) can restore signal in these areas.

Distortions may be corrected for

1. improving registration with non-distorted images (e.g. structurals), or
2. dealing with motion-dependent changes.

FUGUE is designed to deal only with the first case - improving registration. The issue of motion-dependent signal changes (due to motion-dependent changes in field inhomogeneity and distortion directions) is not dealt with in the current version.

# **Fieldmap Acquisition**

Unfortunately, there is no standard sequence for fieldmap acquisitions and different scanners return different images. Normally these images require processing before they represent images with field values in the desired units (of radians/second) in each voxel. For SIEMENS scanners we provide a tool: [fsl\_prepare\_fieldmap](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FUGUE/Guide#SIEMENS_data) to assist with this. Currently this does not support other scanners, but we welcome collaboration from people with other scanners to expand this functionality.

The most common fieldmap sequence acquires two images with different echo times. The change in MR phase from one image to the other is proportional to both the field inhomogeneity in that voxel and the echo time difference. The field value is therefore given by the difference in phase between these two images divided by the echo time difference. This is true for Spin Echo, Gradient Echo or EPI sequences. However, EPI-based fieldmaps suffer from the same distortions (more or less) as the functional images, while Spin Echo or Gradient Echo based fieldmap images do not. Within FSL you cannot use EPI-based fieldmaps with the standard processing, and their use in general is very problematic. We strongly recommend that Spin Echo or Gradient Echo fieldmap sequences are used to acquire the images.

MR phase is the most important quantity in a fieldmap sequence, whereas in normal imaging this phase is not of interest and is normally not saved when reconstructing the images. As a consequence, raw fieldmap scans are somewhat different from most scans, and may contain images of complex values, or separate phase and magnitude images. Furthermore, some scanners/sites may do the full reconstruction of acquired scans to yield a real-valued map of field inhomogeneities (in units of Hz, radians per second, Tesla or ppm). Alternatively no reconstruction may be done, and the raw phase and magnitude (or complex) images may be saved instead, although with modern coil arrays it can be difficult to get correct phase measurements. It is important for each different scanner/site/sequence to know what form your data is in. If they have been converted to NIFTI or ANALYZE format, then you can use the FSL tools (particularly fslinfo and FSLView) to determine the types of images present. To obtain fieldmaps that can be used within FSL using the FSL tools (in particular, PRELUDE and FUGUE), please refer to the page on preparing fieldmaps for FEAT.

# **GUI and command line software**

Two interfaces exist for applying FUGUE to unwarp images: (i) a GUI, that is part of the FEAT preprocessing options, and (ii) a command-line script, epi\_reg. We strongly recommend that new users make use of the pre-stats part of FEAT to do all such processing (note that it is not necessary to have fMRI data for this - even single volumes can be processed in FEAT by selecting pre-stats only). A GUI also exists for getting fieldmaps into the right form for FEAT. See the detailed documentation on FEAT and documentation below on using making fieldmaps for FEAT.

More experienced users, or those wanting to use fieldmap-based distortion-correction in other situations, can use the epi\_reg command. The documentation for this command can be found as part of [FLIRT](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT).

**Making Fieldmap Images for FEAT**

This section outlines some of the most common ways to construct the required fieldmap images for B0 unwarping in FEAT. Also see the documentation on [FEAT](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FEAT) for more information on fieldmapping and its use in FEAT pre-stats processing.

Each scanner/site/sequence can give you different data. As the phase is of great importance in fieldmapping, but is normally not saved in other sequences, these images are often quite different from the standard images.

## **SIEMENS data**

If you have data from a SIEMENS scanner then we strongly recommend that the tool fsl\_prepare\_fieldmap is used to generate the required input data for FEAT or fugue. Fieldmap data from a SIEMENS scanner takes the form of one phase difference image and two magnitude images (one for each echo time). In the following, where a magnitude image is required, pick the "best looking" one. This image is used for registration and masking but the process is not particularly sensitive to the quality and typically either image will work fine.

There are both GUI and command-line versions of the fsl\_prepare\_fieldmap tool. It is recommended that new users use the GUI.

The Fsl\_prepare\_fieldmap GUI requires the following input:

* Phase image
* Magnitude image (brain extracted - see note below)
* Difference of Echo Times - this is a sequence parameter that your scanner operator/radiographer/technician should know, but make sure you record the value

Brain extraction of the magnitude image is very important and must be *tight* - that is, it must exclude all non-brain voxels and any voxels with only a small partial volume contribution. The reason for this is that these areas are normally very noisy in the phase image (look at them in FSLView - if they are not noisy then this is not so important). It is crucial that the mask (derived from this brain extracted image) contains few of these noisy voxels. This is most easily done by making the brain extraction very tight, erring on excluding brain voxels. The exclusion of brain voxels in this instance is actually fine and will have no repercussions, since the fieldmap is extrapolated beyond this mask, and that is the only purpose that the mask plays. Therefore make sure your mask is (if it can't be perfect) too small. As noted above, either magnitude image (from the different echos) can normally be used here - it is not that important.

The output of the Fsl\_prepare\_fieldmap GUI is a correctly calibrated fieldmap in units of rad/s (as the B0 field can be measured in Tesla, Hz or rad/s in MRI as they are all proportional).

The usage of the fsl\_prepare\_fieldmap command-line tool is very similar to the GUI (and all the above considerations apply):

Usage: fsl\_prepare\_fieldmap <scanner> <phase\_image> <magnitude\_image> <out\_image> <deltaTE (in ms)> [--nocheck]

Prepares a fieldmap suitable for FEAT from SIEMENS data - saves output in rad/s format

<scanner> must be SIEMENS

<magnitude image> should be Brain Extracted (with BET or otherwise)

<deltaTE> is the echo time difference of the fieldmap sequence - find this out form the operator (defaults are \*usually\* 2.46ms on SIEMENS )

--nocheck supresses automatic sanity checking of image size/range/dimensions

e.g. fsl\_prepare\_fieldmap SIEMENS images\_3\_gre\_field\_mapping images\_4\_gre\_field\_mapping fmap\_rads 2.65

## **Non-SIEMENS data**

Currently fsl\_prepare\_fieldmap does not work with non-SIEMENS data.

For users with Philips data we highly recommend looking at the following document from Anna van 't Veer, Anne Hafkemeijer, Henk van Steenbergen and Janna Marie Bas-Hoogendam (Leiden University / Leiden Institute for Brain and Cognition):<https://osf.io/hks7x/> This goes through the whole process of preparing and using a fieldmap from a Philips scanner, which is hopefully standardized across scanners and sites, although at the time of writing we do not know how generally true this is.

For other scanners we welcome collaboration from people using such scanners in order to enhance this tool.

In the meantime, when dealing with data from other scanner manufacturers, or with non-standard sequences, the first step in practice is to find out what kind of fieldmap acquisition data you can get from the scanner, operator or reconstruction software. There are three main types:

1. Complex data containing images at two different echo times
2. Separate phase images for the two different echo times
3. A single, real fieldmap image (showing the field inhomogeneity in each voxel)

If the images from the scanner are in ANALYZE or NIFTI form then you can determine their type, use fslinfo. If not, you must convert to either ANALYZE or NIFTI format before progressing. See the FSL FAQ for information about conversion.

When fslinfo is run it will display a datatype string or number (the value 32 represents a complex image). You can convert a complex image into its phase and magnitude components by using fslcomplex or, alternatively, leave it in this format. The number of images contained in each file is shown as the dim4 value.

To further check what sort of images you have, use slices to display the image and then see which of the following images it most closely resembles. Note that for complex images, slices will display two separate images - one phase and one magnitude. If you only have a single image (dim4=1) then it is possible that it has already been reconstructed fully into a fieldmap.

Typical complex images

A phase image (wrapped)

A phase image (unwrapped and masked)

A magnitude image

A real fieldmap image (masked)

Note that the fieldmap image contains structure and shows substantial inhomogeneities mainly in the inferior regions of the brain.

It is most common to have wrapped phase and magnitude images (case 2 below). If the images appear to have unwrapped phase then the the PRELUDE stage (step 3) should be skipped. However, if the range of values within the image is within 2 pi radians (6.28) then it is likely that there are wraps present (even if they are hard to find by eye) and so, in this case, PRELUDE should still be run.

Once you have determined the type, you need to do some or all of the following steps. As a guide (and this may vary in some cases) the steps that are required are:

For complex data you need to do steps 1a, 2a, 3, 4a and 5 For a pair of phase images you need to do steps 1b, 1c, 2b, 3, 4a and 5 For a single, real fieldmap you need to do steps 1b, 1c, 4b and 5

### **Processing Steps**

#### **Step 1 - Getting the magnitude image**

**(a)** If you start with a complex Analyze or Nifti volume that contains the scans at two echo times then you need to do:

fslcomplex -realabs complex\_acq fieldmap\_mag

**(b)** If you have separate phase images or a single, fieldmap image, then you need to also get a magnitude image that is (i) undistorted and (ii) registered with this phase/fieldmap image. Usually the sequence used to acquire the phase or fieldmap image also contains data that can give you this magnitude image. Check with your scanner operator, physicists and/or analysis people as to how to reconstruct this image - often it just requires extraction from the original DICOM or vendor-specific format.

**(c)** Check that the magnitude image and the phase/fieldmap images have the same resolution. You can do this by looking at the dim and pixdim entries (only the first three of each) as reported by fslinfo.

If they are not the same then they must be resampled to be equal. In this case choose the one with the best resolution and use this as a reference image in flirt with the -applyxfm option to resample the other images. For example, if the magnitude image has a better resolution (smaller pixdims) then do the following:

flirt -in original\_phase0 -ref fieldmap\_mag -applyxfm -out orig\_phase0

flirt -in original\_phase1 -ref fieldmap\_mag -applyxfm -out orig\_phase1

Once this is done, check that the output images (e.g. orig\_phase0) have the same dimensions and resolution as the reference (using fslinfo) and **also** check that they are aligned correctly by loading both the output and reference images into fslview and visually inspecting them.

#### **Step 2 - Getting (wrapped) phase in radians**

**(a)** If you have complex volumes then do:

fslcomplex -realphase complex\_acq phase0\_rad 0 1

fslcomplex -realphase complex\_acq phase1\_rad 1 1

These phase volumes will now be in radians. **(b)** If you have seperate phase volumes that are in integer format then do:

fslmaths orig\_phase0 -mul 3.14159 -div 2048 phase0\_rad -odt float

fslmaths orig\_phase1 -mul 3.14159 -div 2048 phase1\_rad -odt float

Note that the value of 2048 needs to be adjusted for each different site/scanner/sequence in order to be correct. The final range of the phase0\_rad image should be approximately 0 to 6.28. If this is not the case then this scaling is wrong. If you have separate phase volumes are not in integer format, you must still check that the units are in radians, and if not scale them appropriately using fslmaths.

#### **Step 3 - Unwrapping the phase images**

Use PRELUDE to do the required phase unwrapping

prelude -a fieldmap\_mag -p phase0\_rad -o phase0\_unwrapped\_rad

prelude -a fieldmap\_mag -p phase1\_rad -o phase1\_unwrapped\_rad

#### **Step 4 - Getting the fieldmap in rad/s**

**(a)** For separate phase images do:

fslmaths phase1\_unwrapped\_rad -sub phase0\_unwrapped\_rad -mul 1000 -div TE fieldmap\_rads -odt float where **TE** must be replaced with the appropriate difference in echo times (in units of milliseconds).

**(b)** If you have a single, real fieldmap then you must determine the units of this fieldmap (ask an operator/physicist) and rescale to radians per second if it is not already in these units. Common other units are (i) Hz (scale these by 6.28 to get rad/s) and (ii) Telsa (scale these by 2.68e8 to get rad/s).

#### **Step 5 - Regularising the fieldmap**

Fieldmaps can often be noisy or be contaminated around the edges of the brain. To correct for this you can regularise the fieldmap using fugue. Note that the "best" regularisation will depend on many factors in the acquisition and must be determined separately for each site/scanner/sequence. Look at the fieldmap (e.g. using fslview) to decide what is the best regularisation to use - which could also be to do no regularisation.

Some regularisation options are - Gaussian smoothing, despiking and median filtering. Examples of these (in order) are:

fugue --loadfmap=fieldmap\_rads -s 1 --savefmap=fieldmap\_rads

fugue --loadfmap=fieldmap\_rads --despike --savefmap=fieldmap\_rads

fugue --loadfmap=fieldmap\_rads -m --savefmap=fieldmap\_rads

Any combination of these regularisations can be performed. See the command-line documentation on fugue below for more information aspects of regularisation.

# **Command-line tools (advanced)**

For all programs the options follow the normal convention that "single minus" options are separated by a space from their arguments (if any) whilst "double minus" options are separated by an equals sign and no space. For example,

prelude -c data --unwrap=result

or

prelude --complex=data -u result

## **FUGUE**

fugue (FMRIB's Utility for Geometrically Unwarping EPIs) performs unwarping of an EPI image based on fieldmap data. The input required consists of the EPI image, the fieldmap (as an unwrapped phase map or a scaled fieldmap in rad/s) and appropriate image sequence parameters for the EPI and fieldmap acquisitions: the dwell time for EPI (also known as the echo spacing); and the echo time difference (called asym time herein). The main forms of usage are:

fugue -i epi -p unwrappedphase --dwell=dwelltime --asym=asymtime -s 0.5 -u result

* fieldmap specified by a 4D file unwrappedphase containing two unwrapped phase images - from different echo times - plus the dwell time and echo time difference (asym time)

fugue -i epi --dwell=dwelltime --loadfmap=fieldmap -u result

* uses a previously calculated fieldmap

Note the option -s 0.5 is an example of how to specify the regularisation to apply to the fieldmap (2D Gaussian smoothing of sigma=0.5 in this case which is a reasonable default). There are many different forms of regularisation available which can be applied separately or together. These are:

-s sigma

* 2D Gaussian smoothing

--smooth3=sigma

* 3D Gaussian smoothing

-m

* 2D median filtering

--poly=n

* 3D Polynomial fitting of degree n

--fourier=n

* 3D Sinusoidal fitting of degree n

Some other uses are:

fugue -i undistortedimage -p unwrappedphase --dwell=dwelltime --asym=asymtime --nokspace -s 0.5 -w warpedimage

applies the fieldmap as a forward warp, turning an undistorted image into a distorted one - useful for creating a registration target for the EPI from the undistorted absolute fieldmap image Additional options that are useful are:

--mask=maskname

* uses a user-defined mask (called maskname) instead of deriving it from the phasemaps or fieldmaps

--unwarpdir=dir

* specifies the direction of the unwarping/warping - i.e. phase-encode direction - with dir being one of x,y,z,x-,y-,z- (default is y).
* Note: the conventions for x, y and z are voxel axes (in the input image), but with the x-axis being swapped if the qform or sform determinant is positive (to match the internal FSL voxel coordinate axes), though due to other possible changes in the pipeline for reconstructing/analysing both the EPI and fieldmap images, it is usually necessary to determine the sign of the direction by trial-and-error once for each scanning/reconstruction/analysis protocol (each dataset using the same protocol will need the same unwarpdir).

--phaseconj

* uses the phase conjugate correction method, rather than pixel shifts

--nokspace

* for forward warping (only) - uses an image-space method for forward warping

--icorr

* applies an intensity correction term when using the pixel shift method - often poorly conditioned for standard fieldmap acquisitions

## **>> PRELUDE (phase unwrapping)**

**prelude** ( Phase Region Expanding Labeller for Unwrapping Discrete Estimates ) performs 3D phase unwrapping of images. As the name implies, it should be run before fugue in order to get a correct fieldmap. This is useful for fieldmap acquisitions, susceptibility weighted imaging (SWI), or other applications involving phase in MR (or non-MR) images. The input can either be a single complex image (NIfTI or Analyze), or a pair of real images giving the phase and absolute values separately. Also see [Fslutils](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Fslutils) for more ways to manipulate complex image formats. If the images are 4D images, then each 3D volume is unwrapped separately, and the result saved as a 4D image of unwrapped phase images. The output in either case is a real, unwrapped phase image (in radians).

For higher resolution images (e.g. for Susceptibility Weighted Imaging) a mask should be used in order to reduce run-times since any areas that include noisy voxels will significantly increase the required run-time. The best way to generate a mask for a whole brain image is to run BET to perform brain extraction on the absolute image.

The three main forms of usage are:

prelude -c data -u result

* uses a single complex input file

prelude -a data\_abs -p data\_phase -u result

* uses separate phase and absolute input files

prelude -a data\_abs -p data\_phase -u result -m mask

* uses separate phase and absolute input files with a mask

Additional options that are useful are:

-m mask

* uses the user-defined mask (e.g. a BET mask - especially for high resolution images)

-n num

* specifies the number of phase partitions the algorithm uses for initial labelling - a larger value is likely to be more robust but slower

-s

* unwrap in 2D, then stick slices together with appropriate offsets - this is less robust but fast - mainly used for very high-resolution images where optimising speed is an issue

--labelslices

* does labelling in 2D, but unwrapping in 3D - the default for high-res images