# fMRI data preprocessing using a flexible but automated pipeline

## What it required beforehand

* SPM & Matlab
* Bias correct toolbox (Sandrine, based on Antoine’s code) (included in the package)
* Converted fMRI data to NifTi file format
* Anatomical scan (also converted to NifTi, or already preprocessed)
* VBQ toolbox if starting from unpreprocessed MPM data
* RETROICOR (RETROspective Image-based CORrection of physiological noise in fMRI data) toolbox necessary if physiological data are to be used during the first level analysis
* A certain directory structure : two coherent hierarchies are supported

-------------------------------------------------------

Folder hierarchy type 1

(SPM DICOM import output with setting "patid\_date"):

-------------------------------------------------------

-> root

-> subject ID

-> sequence (MR protocol) \_ protocol run number \*\*

-> image files

-------------------------------------------------------

-------------------------------------------------------

Folder hierarchy type 2

(LREN's automated pipeline):

-------------------------------------------------------

-> subject ID

-> session (experimental run number)

-> sequence (MR protocol)

-> repetition (protocol run number)

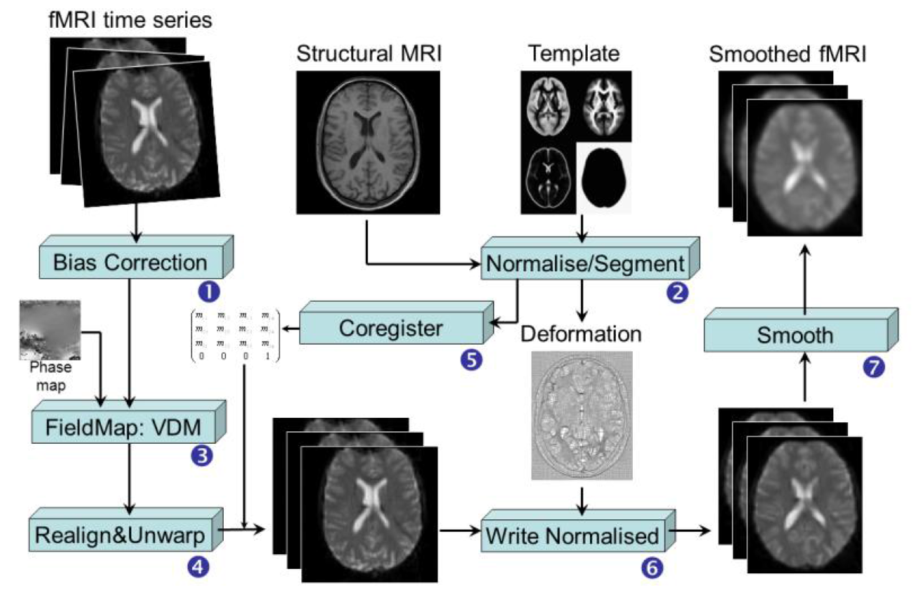
-> image files

-------------------------------------------------------

\*\* (The “\_” between “sequence (MR protocol)” and “protocol run number” designates the fact that the two are concatenated, e.g. “gre\_field\_mapping\_1acq\_rl\_0008” indicates that the protocol is “gre\_field\_mapping\_1acq\_rl” and that it was the 8th MR protocol applied during the acquisition.)

* Main outputs :
  + A MAT file containing SPM jobs (matlabbatch), state of configuration file at the time of running (Opts), files detected (Session), final prefix of preprocessed files (prefixNIIf). The files starts with “Log\_auto\_preproc\_” and adds a date-time-... tag so that preprocessing can be traced back if necessary.
  + Realignment parameters (txt) file (if preprocessing is validated)
  + Preprocessed files (at each stage of preprocessing step) (if preprocessing is validated)

## What it does



Realignment of volumes to the mean image (2-pass procedure) using the job “Realign: estimate and write”. Default values are left, except precision which is increased to 1 (instead of 0.9) and only the mean is written to the disk to save space: functional volumes are therefore realigned by changing the header of the file. This holds as soon as normalization is applied before the first level analysis (if not “fMRI model specification” will later send an error indicating that volumes are not aligned). An option is however provided to reslice all functional volumes if desired (Opts.Reslice). If multiple sessions are available for a given subject, volumes will be realigned first within session and then between session, and a single mean will be outputted for later coregistration.

If B0 maps are found, EPI image distortion is applied using Fieldmap toolbox (“Presubtracted phase and magnitude data” and then “Realign & unwarp”, instead of “Realign: estimate and write”) and pre-defined default files for correction (this needs to be updated each time a new sequence is added). More precisely the first magnitude image is chosen among the two that are available. The favoured GRE field mapping is “gre\_field\_mapping\_1acq\_rl\_64ch”. If the latter is found, sequences such as “gre\_field\_mapping\_1acq\_rl” or “gre\_field\_mapping\_1acq\_rl\_short” are ignored (left for VBQ preprocessing). If “gre\_field\_mapping\_1acq\_rl\_64ch” is not found but other sequences beginning with “gre\_field\_mapping...” are found, those are used instead. If nothing is found, EPI image distortion correction is not applied. The Fieldmap toolbox has been found to exhibit a strange behaviour in some situations, sending back the so-called “schur” error in Matlab’s command line during the job “Presubtracted phase and magnitude data”. This has been eschewed by segmenting the magnitude image, summing tissue classes corresponding to WM, GM and CSF and mask the original magnitude image with the resulting brain mask.

Receive coil bias is particularly strong on machines with high field strength. This is why a bias correction procedure is applied to eschew issues that may arise further during the first level analysis (typically voxels in central brain structures are masked out and inferences are not drawn in these regions!). Basically this procedure simply segment the first functional scans using “Segment” and outputs the bias. This bias is then used to correct all functional volumes.

Coregistration of anatomical scan to the mean functional image. Default values are left.

Segmentation of the coregistered anatomical scan (SPM12 / New Segment in SPM8) : output of tissue classes c1, c2, c3, the bias corrected anatomical scan, the inverse and forward deformation fields.

If DARTEL is called (assuming the pipeline has been run on a root directory containing several subjects), DARTEL is performed using default parameters and the job “DARTEL: create template”. In this case, Segment outputted additional files : rc1, rc2 and rc3. These files are then used to create a DARTEL template specific to the pool of subjects in the root folder. Afterwards flow fields are used in the normalisation step instead of forward deformation fields.

Otherwise (if DARTEL is not called (default behaviour)) the job “Deformations” does the normalisation by using the forward deformation fields. Parameters of this job are left with default values (no smoothing applied), except that normalized volumes are outputted in the source directory (instead of current directory).

Smoothing is applied using “Smooth” and a kernel of 6 mm FWHM.

## Why do I need the VBQ toolbox for the fMRI pipeline?

Normally this part of data preprocessing is supported by the VBQ pipeline but the fMRI pipeline can be called locally by a user to preprocess fMRI data coming from outside of the lab.

## How to use it

The main function to call is “fMRI\_automated\_preproc.m”. This function accepts up to 4 inputs.

The first argument (RootPath) is necessary and indicates the root folder in the form of a string, i.e the folder right above the subject folder, whether this folder contain several subjects or not (e.g. ‘C:\DATA\my\_data\’).

The second argument (SubjectID) is necessary and indicates the subjects to be processed in the root folder (e.g. ‘PR00001’ or ‘sub\_01’). If all subjects in the root folder have to be processed, then this argument can take the following form: ‘’. However it has to be mentioned that if the root folder contains subfolders that do NOT contain fMRI data, it will be necessary to specify the folders to be considered by the pipeline otherwise the pipeline will fail to find data in those subfolders.

The third argument is optional and is mainly intended for testing purposes. It accepts a structure containing information about the way preprocessing should be done.

IMPORTANT : if this third argument is not present, “fMRI\_automated\_preproc.m” reads the configuration file (“fmri\_pipeline\_config.txt”). This optional argument basically bypass the information contained in the configuration file and use the structure inputted to process the fMRI data. It is thus the choice of the final user to adapt the configuration file (useful for long-term changes) or to inject information through this optional argument (in the latter case only the present call of the function will be affected by the parameters specified in the structure and each time such a preprocessing will be needed this optional argument will be required and, beforehand, constructed manually).

The function “fMRI\_automated\_preproc.m” contains a rather detailed help section on how to construct the structure for the optional structure. It corresponds to the way the configuration file is written. Therefore although the pipeline runs perfectly with default parameters one may want to modify its behaviour in some circumstances. Because constructing a structure for the optional argument is time-consuming, it is recommended to modify the configuration file instead, except if the behaviour of the pipeline has to be changed for a very short period of time. Reading the help section of the main function is also recommended to modify the configuration file correctly. In summary, the way each option is named (e.g. “MinVolNum”) shall NOT be changed, neither the semicolon or space right after removed or changed. Only the values at the end of each line should be modified if necessary, and this should be done by choosing among proposed alternative values only. Unless the code of the function is modified to inform the pipeline that a new parameter is present (and how to deal with it), additional parameters will be ignored.

Now let’s take a look at the configuration file:

|  |  |
| --- | --- |
| MinVolNum: 10 | Integer, indicates the minimal number of functional volumes that defines a true fMRI run. This option has been implemented to automatically discard runs that are aborted for any reason during acquisition (e.g. task not starting). fMRI runs with less than the specified number will be automatically ignored. |
| RunDARTEL: 0 | Numerical, can be 1 (create DARTEL template from sample and estimate flow fields to normalize to MNI space) or 0 (uses deformation fields with default SPM template instead). (only relevant if processing several subjects at once (SubjectID = ‘’)). |
| Mode: 'interactive' | String, can be set to 'interactive' (open the resulting jobs in SPM Batch Editor interface before running it) or 'run' (simply run the jobs). ‘run’ is more appropriate for true automated preprocessing, but ‘interactive’ is the default mode as it allows a visual check before running the jobs. The latter is very useful for debugging. |
| TokenEPI: 'al\_mepi' | String, keyword used to detect the folder(s) containing the functional scans. |
| RegisterToMean: 1 | Numerical, can be set to 1 (during realignement, register to the first and then to the mean using a two-pass procedure (default)) or 0 (register to the first volume only). It has to be beared in mind that coregistration will later use either the mean or the first functional volume as a reference and that the reference has to be representative. |
| FWHM: [6;6;6] | 1 x 3 numerical, defines the size of the Gaussian smoothing kernel in x-, y- and z-directions (FWHM). |
| DetectResolution: 1 | Numerical, can be set to 1 (use the token "mm" to detect potential presence of multiple EPI datasets at different resolutions (and process them separately by duplicating folders containing B0 maps and structural scan)) or 0 (assumes all datasets (if several in each subject's folder) have the same resolution). Although multi-resolution acquisitions are relatively rare, preprocessing can dramatically fail in such cases. The Realign module will fail because the mean image cannot be computed with different voxel and / or image dimensions. This aims at resolving this issue but relies only on the assumption that the resolution of fMRI acquisition can be detected in the sequence name. For example, “al\_mepi2d\_v2f\_3mm” will be considered as a 3mm resolution sequence, and “al\_mepi3d\_v3c\_1pt5mm” will be considered as a 1.5mm resolution sequence. |
| StructMT: 1 | Integer indicating the rank of the MT image for normalisation when multiple anatomical data are present. Either StructMT or StructMPRAGE has to be set to 1, and the other to 2. |
| StructMPRAGE: 2 | Integer indicating the rank of the MPRAGE image for normalisation when multiple anatomical data are present. Either StructMT or StructMPRAGE has to be set to 1, and the other to 2. |
| DirStructure: 'LRENpipeline' | string, can be either 'LRENpipeline' or 'DICOMimport'. SPM DICOM import is commonly used and encountered in data acquired from outside of the lab, whereas the other option is more precise, aims at dealing with more complex situations (protocols are repeated, longitudinal studies) and is intended for the automated preprocessing of LREN Prisma's data. This is meant to deal with different types of organization of data. See the remark on directory structure above, at the beginning of the document, and bear in mind that the structure has to be similar and coherent across the different subjects. |
| MaskMag: 1 | Scalar, can be 1 (mask the magnitude image of B0 maps for EPI image distortion correction to prevent "schur" and related errors) or 0 (don't mask the magnitude image). It is recommended to leave it to default (1) as the errors are difficult to predict and can make a whole preprocessing fail.  This also applies to masking of magnitude image of B0 maps for creation of multiparameter maps using VBQ. |
| Threshold\_masking\_MT\_with\_PDw: 100 | Scalar, threshold for masking MT map created using VBQ using PDw. Usually it is recommended to set it to 100 or so. This step is removing noisy values in voxels close to the edges of the brain to obtain a better segmentation of the structural scan, and subsequently a better normalization of brain images. Obviously if MPM data have been previously preprocessed the fMRI pipeline will ignore this parameter and use the masked MT map already computed. |
| Reslice: 0 | When realigning images, reslicing can be performed. This is usually not necessary as normalization to MNI space will anyhow reslice images (and one interpolation is better than two), but in case GLM analysis has to be done in native space, this option can be specified.  Reslice is then an integer that can be equal to 0 (to reslice only the mean image) or 1 (to reslice the mean and all the other volumes). |
| SpecialTokenStruct: '' | Common anatomical data are MPRAGE or MPMs. These protocols will be linked with a sequence name containing the keyword “mprage” or respectively “mt\_al\_mtflash”, “pd\_al\_mtflash”, and “t1\_al\_mtflash”. For special cases a special token can be informed via this parameter. When set to ‘’, the pipeline looks for anatomical data such as MPRAGE or MPMs according to the default tokens defined above. However any token can be specified in the form of a string, such as e.g. “structural”. It has to be noted that in the latter case the anatomical data will be assumed to be ready for coregistration (i.e. if some particular MPM data have a different sequence name, the masked MT map should be available before running the fMRI pipeline). |
| FilterAlreadyPreprocessedSMRI: 1 | Data that are already preprocessed will be often encountered, especially with anatomical scan. This option simply avoids confusion for the pipeline when detecting files that are the output of a previous preprocessing (such as files beginning with c1, c2, etc.) and prevents to consider them as a multitude of unpreprocessed files. Although it is recommended to leave this option to 1 (filtering), it can be disabled (0). |
| FilterAlreadyPreprocessedFMRI: 0 | Sometimes the pipeline will also encounter fMRI data that have already (partially) preprocessed. Again, this option simply avoids confusion for the pipeline when detecting files that are the output of a previous preprocessing and prevents to consider them as a multitude of unpreprocessed files. Contrarily to the other option, this one is unnecessary and is left to 0 (no filtering is applied) because although the majority of jobs will overwrite the previously processed files, the realignment of functional volumes shouldn’t be done multiple times. Even though it could be easily checked if realignment has been already performed with a relatively high level of confidence (the headers of the files are usually at least slightly different for each file), the original files are in this case not available anymore and it is necessary to start again from a backup of the converted data (or recomputed them from the DICOM!) because otherwise : 1) the data will be realigned twice ; 2) the realignment parameters file will be incorrect, preventing any additional motion correction at the first level analysis. |

## Why is there a second function named “fMRI\_automated\_preproc\_multires.m”? I thought “fMRI\_automated\_preproc.m” was dealing with multi-resolution datasets!

“fMRI\_automated\_preproc.m” indeed deals with multi-resolution datasets. However it does it in a particularly inefficient way. The issue is the following: as a single mean image is impossible to compute during realignment, several means are computed to maintain high-quality preprocessing. Besides it has to be beared in mind that a “realign to the first” strategy wouldn’t solve the problem because it would still produce 3 references (the first scan). Furthermore it would potentially compromise the high quality of data preprocessing achieved through the two-pass procedure. Therefore the concern arises: a single anatomical scan has to be coregistered to several mean images. Although a suboptimal solution could consist of reslicing each mean and produce a mean of the mean images, a more clever solution can be adopted: move the functional images so that they match the anatomical scan instead of the reverse. Usually this solution is not chosen because it is more time-consuming to coregister several hundred of scans to one instead of the reverse. Nevertheless the procedure doesn’t have to be repeated for each volume but can be rather substantially improved by estimating the optimal registration of the mean of the functional volume to the anatomical scan once, and simply apply the estimated rigid-body transformation matrix to the time-serie. In summary “fMRI\_automated\_preproc\_multires.m” solves the problem of multi-resolution datasets this way, whereas “fMRI\_automated\_preproc.m” solves the problem by making making copies of the anatomical scan and coregister each of these to the each mean image of each functional time-serie. The additional drawback of the latter procedure is that each time-serie with a particular resolution will not necessarily be in the same space as the others. This prevents further analyses comparing the results for the different resolutions.

## I thought the fMRI pipeline aimed at doing preprocessing only, but I see some functions related to first level analysis... Why?

“fMRI\_automated\_first\_level.m” is the main function that can automatically perform first-level analyses. Everyone who had to deal with fMRI data knows that once the boring preprocessing is done, there is another “boring” job to do before coming to the interesting group-level analysis : the first level analysis, for each subject. This is not a *boring* task per se, because it is a crucial step in the modelling of the response of interest. The latter statement can be understood in two senses: 1) an inappropriate modelling at the subject level can circumvent results at the group level, especially because the power of fMRI is the relatively high number of measure per subject (hundreds of time points per run) and this advantage shouldn’t be underestimated to get the most of the data in the subsequent group level analysis; 2) one can say that the researcher’s intervention definitely starts here, as this part of preprocessing is tightly linked with the experimental paradigm and is therefore highly study-specific. Nevertheless once the experimental design is informed, estimation is very often a surprisingly straightforward procedure yet relatively time-consuming and making contrasts between regressors of interest still take some time for the contrast images and SPMs to be computed. Moreover, this procedure has to be repeated for each subject. This is where it becomes really boring. Hopefully this procedure can be automatized in SPM. Nonetheless preprocessed functional volumes need to be located for each subject with their corresponding realignment parameters file in order to improve the validity of the inferences (about 90% of the signal in fMRI is due to head motion). Sometimes physiological data are available to further improve the statistical analyses. In the latter situation a complex interplay arises between image acquisition and raw physiological signals. For all these aforementioned purposes, and because a substantial part of those information had to be gathered previously during the preprocessing, “fMRI\_automated\_first\_level.m” facilitates and optimizes this part of the scientific work with functional neuroimaging data by incorporating in a single function “fMRI model specification”, “Model estimation” and “Contrast manager”. It is fuelled with the outputs of “fMRI\_automated\_preproc.m” (or equivalently “fMRI\_automated\_preproc\_multires.m”) and just requires from the end-user some additional information detailed in the help section of the function.

## I am seeing warning messages in Matlab’s command line, should I be worried?

The warning messages the pipeline delivers have been designed to be the clearest possible. For example there is inevitably a warning message that reminds end-users what is done for EPI image distortion correction (whether B0 maps were found or not, whether default set of files (LREN) for Fieldmap toolbox are used, ... and so on). Read them and it should be easier to make an informed decision: aborting the preprocessing immediately or pressing quietly the green arrow... Do not forget also that the interactive mode allows checking the jobs before running them, and remember that it is always safer to have somewhere a copy of the original data...

## Can the fMRI pipeline fail?

Although a considerable work has been done to prevent errors, this can unfortunately happen. The pipeline is still in beta testing. Furthermore there are probably some situations we hadn’t thought about. The question should rather be: what happens when it fails? Error messages in Matlab’s command line can sometimes occur although it has been avoided as much as possible. This situation has the merit of being clear but can lead to the abortion of a daemon running under Matlab, which is a more problematic. If the pipeline fails to retrieve files for a subject (for any reason, the most common one being having forgot that there is an additional folder that has nothing to do with the normal folder hierarchy, e.g. having a folder called “Matlab scripts” at the same level as the folders containing the data!). In such situations the pipeline will deliver a warning message that “one subject could not be processed correctly” and the problematic case is skipped. Furthermore a MAT-file beginning with “WARNING\_auto\_preproc\_” will be saved in the directory of each problematic subject, whose content will be a message indicating where the problem was coming from (e.g. ‘Folder with structural scan not found’). Finally, SPM can also occasionally deliver warning messages. Part of those can be ignored (e.g. “QFORM0 representation has been rounded” with Deformations) but some others should alert end-users (e.g. “Item ‘...', field 'val': Value must be either empty, a cellstr or a cfg\_dep object.”). This type of issue can occasionally occur and can either result in an inappropriate processing, either make further processing steps fail and be skipped or generate error messages that stop processing. Most of the time they don’t abort Matlab commands, which is a preferable condition with truly automatized platforms, but they are silly in the sense that nothing will warn the end-user that something went wrong, except the final output which will be hopefully questioned.

NOTA BENE : default files for EPI image distortion have to be updated ! 🡺 see “fmri\_protocols\_2014\_08\_07.txt” !