TMFC_denoise

Release v1.4.1

Ruslan Masharipov

Oct 11, 2025

INTRODUCTION

TMFC_denoise is a MATLAB toolbox for SPM12/SPM25 that performs GLM-based denoising (**noise regression**).

This toolbox allows you to **add noise regressors** to the original general linear model (GLM), calculate **framewise displacement (FD)**, Derivative of root mean square VARiance over voxelS **(DVARS)**, and **FD-DVARS correlation** before and after denoising.

The updated GLMs can be used for **task-based activation analysis** or for **task-modulated functional connectivity (TMFC) analysis**.

CHAPTER

ONE

INSTALLATION

You can install **TMFC_denoise** either:

1. As a **standalone toolbox**, or
2. As part of the **TMFC toolbox** (https://github.com/IHB-IBR-department/TMFC_toolbox).

1.1 Dependencies

Before using **TMFC_denoise**, make sure the following software is installed:

- **MATLAB** R2021b or newer
- **SPM12** or **SPM25**
- (Optional) **Parallel Computing Toolbox** – for parallel computations

1.2 Download

You can obtain the toolbox from any of the following sources:

- **GitHub repository:** https://github.com/IHB-IBR-department/TMFC_denoise
- **Zenodo DOI:** https://doi.org/10.5281/zenodo.17176264
- **Included in the TMFC toolbox**: https://github.com/IHB-IBR-department/TMFC_toolbox

1.3 Installation Steps

1. **Download and unzip** the toolbox archive from GitHub or Zenodo.
2. In MATLAB, open *Home* → *Set Path* → *Add with Subfolders*.
3. Select the `TMFC_denoise` folder (or the `TMFC_toolbox` folder, if using the integrated version).
4. Click **Save** and **Close**.
5. Test the installation:

   ```
   TMFC_denoise
   ```

   This command should open the TMFC_denoise GUI.

Alternatively, if you installed the full TMFC toolbox:

```
TMFC
```

Then, in the main TMFC GUI, choose *Tools* → *Denoise*.

1.4 Command-Line Usage

The **TMFC_denoise** toolbox can also be executed without using the GUI:

```
output_paths = TMFC_denoise(SPM_paths,subject_paths,options)

output_paths = TMFC_denoise(SPM_paths,subject_paths,options,anat_paths,func_paths)

output_paths = TMFC_denoise(SPM_paths,subject_paths,options,anat_paths,func_paths,
↪display_FD,estimate_GLMs,clear_all)
```

For command-line examples, see the `examples` folder in the GitHub repository: examples folder of the TMFC_denoise GitHub repository

CHAPTER

TWO

OVERVIEW

TMFC_denoise provides both a graphical user interface (GUI) and command-line functionality. To open the GUI, run the `TMFC_denoise.m` function in MATLAB.

TMFC_denoise generates nuisance regressors, calculates quality control (QC) measures, and estimates updated GLMs using weighted least-squares (WLS) or robust WLS (rWLS).

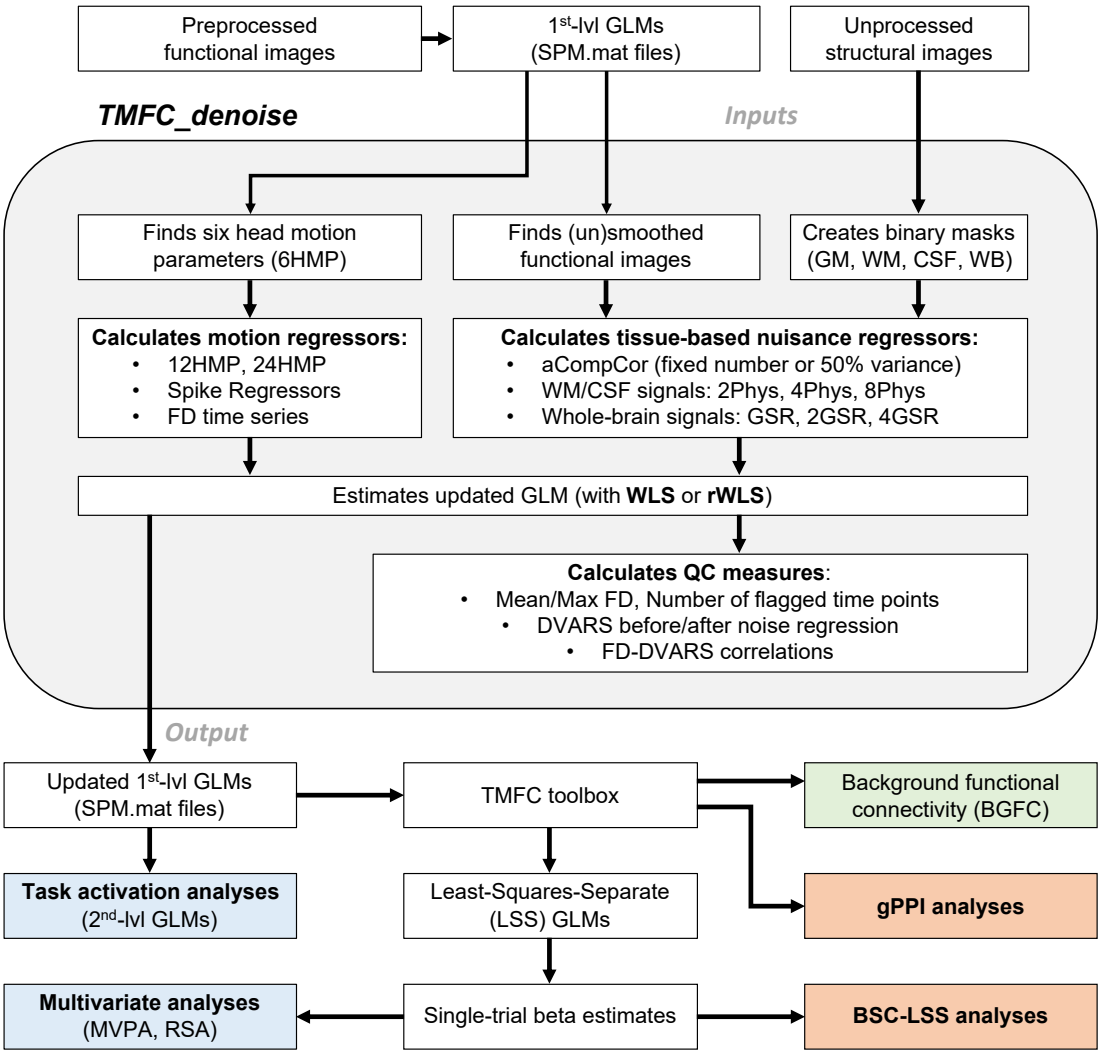

Fig. 1: TMFC_denoise overview.

2.1 Inputs

Inputs include unprocessed structural and preprocessed functional images, together with first-level GLMs specified in SPM.

Users specify paths to first-level GLMs (`SPM.mat` files), select denoising options, and set masking parameters. First-level GLMs must be **specified and estimated** in SPM12 or SPM25 and **must include six head motion regressors**.

Unprocessed structural and preprocessed functional images can be automatically identified through the GUI. Functional images may be preprocessed using an SPM-based pipeline (e.g., see the `preproc_fmri.m` function in `/spm/batches/`; Penny et al., 2011) or with alternative pipelines such as fMRIPrep (Esteban et al., 2019). Preprocessing should include **realignment** and **normalization**, whereas **slice-time correction** and **smoothing** are optional.

2.2 Denoising Options

1) Head motion expansions
2) Framewise displacement (FD)
3) Spike regressors
4) aCompCor regressors
5) WM/CSF regressors
6) Global signal regressors (GSR)
7) DVARS (Derivative of root mean square VARiance over voxelS) and FD-DVARS correlations
8) Robust weighted least squares (rWLS)

2.3 Outputs

All outputs, including noise regressors, updated GLMs, and QC measures, are saved in a `TMFC_denoise` subfolder within each subject's first-level GLM directory. Group-level QC measures can be saved as a single `.mat` file in a user-specified directory.

2.4 How to Use Updated GLMs

1. Updated GLMs can be used for **task-based activation analyses**.
2. Updated GLMs can be used as **input to the TMFC toolbox**, which implements:

- **Background functional connectivity (BGFC)**
- **Least-squares-separate (LSS) GLMs**
- **Beta-series correlation (BSC-LSS)**
- **gPPI with deconvolution**

3. The **TMFC toolbox** can also generate denoised **volume of interest (VOI)** files for **dynamic causal modelling (DCM)**. **Note:** The original model should be prepared for DCM analysis. Both **TMFC_denoise** and the **TMFC toolbox** support `SPM.mat` files with concatenated sessions (i.e. `spm_fmri_concatenate.m`).

4. Denoised **single-trial beta estimates** (outputs of LSS GLMs) can also be used for **multivariate approaches**, including **multivoxel pattern analysis (MVPA)** and **representational similarity analysis (RSA)**.

CHAPTER

THREE

PREPARE YOUR DATA

TMFC_denoise uses information from `SPM.mat` files to obtain paths to functional images and SPM binary masks.

3.1 Original GLMs

The original model must be **estimated** (not only **specified**) in SPM12 or SPM25.

Note: Original models must include **six head motion regressors**, and the specified functinal images must be **realigned** and **normalized** (smoothing is optional).

If you estimated first-level GLMs and later moved the GLM or functinal-image folders to another location, you need to update the paths specified in the `SPM.mat` files (`SPM.xY.VY` and `SPM.swd` fields).

To change paths in `SPM.mat` files using the GUI, enter the following command in the MATLAB window:

```
tmfc_change_paths_GUI
```

If you use the TMFC toolbox GUI, click *Tools* → *Change paths*.

Select the subjects whose `SPM.mat` files you want to update. Enter the old path pattern (see `SPM.swd` field in the SPM.mat file) and the new path pattern:

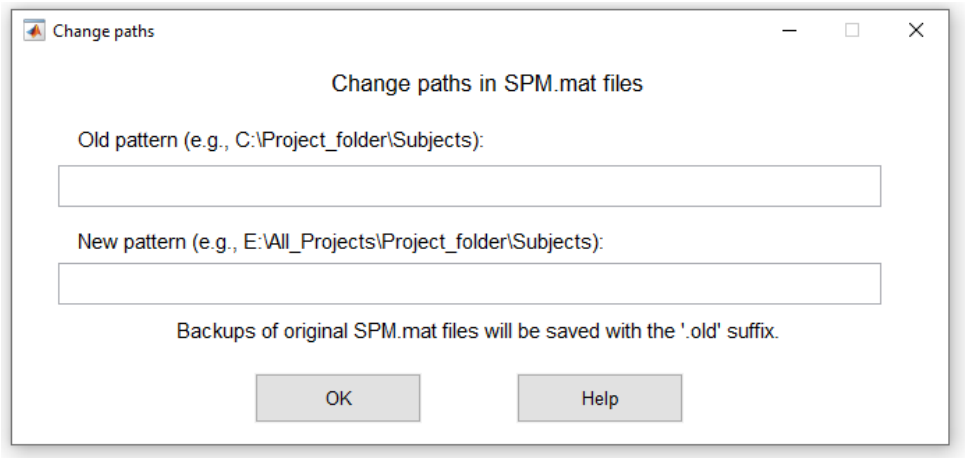

Fig. 1: Change paths GUI.

To change paths in `SPM.mat` files from the command line, enter:

```
tmfc_change_paths_GUI(SPM_paths, old_path, new_path)
```

3.2 Structural Images

To calculate tissue-based regressors and/or DVARS, you need to create tissue-specific binary masks. These masks are generated from unprocessed T1-weighted structural images. Structural images must be in **native space**, since binary masks are created in native space and later normalized to MNI space.

Structural images can be coregistered with functional images (optional). This is not necessary because TMFC_denoise automatically coregisters the skull-stripped structural image and binary masks to the functional images.

Structural images may be in `*.img`/`*.hdr`, `*.nii`, or `*.nii.gz` format.

Note: To automatically select structural images using the GUI, they must: - Have the same root folder and subfolder structure, - Share the same file format, and - Have a consistent name pattern (to be uniquely detected via a text filter, e.g., `*T1*.nii*`).

3.3 Functional Images

To calculate tissue-based regressors or DVARS, you should extract tissue-specific signals from **unsmoothed** functional images. Functional images must be **realigned** and **normalized**, as binary masks are normalized to MNI space.

If your original model was specified for unsmoothed images, simply press *Preserve functional image paths from the SPM.mat files* during functional image selection. If your model was specified for smoothed images, you need to provide paths to unsmoothed images.

In principle, tissue-specific signals can be extracted from smoothed images (e.g., if unsmoothed data are unavailable), but unsmoothed images are preferred for noise-regressor creation.

Images can be either `*.img`/`*.hdr` (3D) or `*.nii` (3D/4D). The `*.nii.gz` format is not supported.

Note: To automatically select functional images using the GUI, they must: - Have the same root folder and subfolder structure, - Share the same file format, and - Follow a consistent name pattern (to be uniquely detected via a text filter, e.g., `*war*.nii`).

CHAPTER

FOUR

SELECT SUBJECTS

At the first step, the `TMFC_denoise` GUI prompts the user to select the first-level GLMs to be updated with nuisance regressors.

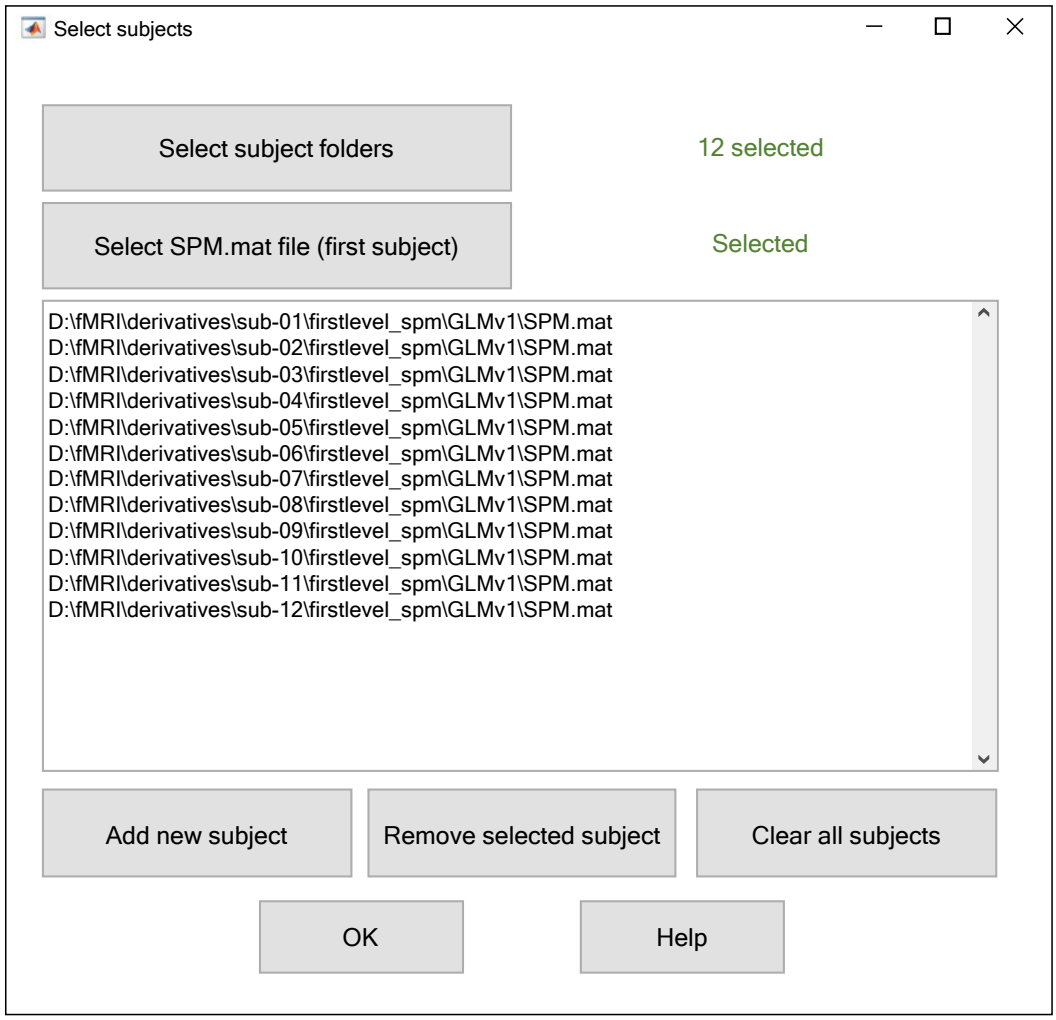

Fig. 1: Select subjects GUI.

First, select the subject folders, each containing a subfolder with first-level GLMs. Then, select the `SPM.mat` file for the GLM of interest for the **first subject**. Paths to GLMs for all other subjects will be constructed automatically.

To open *Select subjects* GUI independently of the main `TMFC_denoise` function, run:

```
[SPM_paths, subject_paths] = tmfc_select_subjects_GUI(0);
```

Outputs:

- `SPM_paths` — Full paths to selected `SPM.mat` files (cell array)
- `subject_paths` — Paths to selected subject folders (cell array)

4.1 Example 1 — SPM-like Folder Structure

1. Select the subject folders (each containing a `STAT` subfolder with an `SPM.mat` file).
2. Select the `SPM.mat` file for the first subject.

```
project/
├─ rawdata/   # DICOM
└─ derivatives/
   ├─ sub-01/   <------------------------------------ [Select subject folder #1] (1)
   │  ├─ anat/
   │  │  ├─ *T1*.nii
   │  │  └─ *T1*.nii derivatives
   │  ├─ func/
   │  │  ├─ sess-01/
   │  │  │  ├─ Unprocessed functional files (*.nii)
   │  │  │  └─ Preprocessed functional files:
   │  │  │       • smoothed + normalized + realigned (e.g., swar*.nii)
   │  │  │       • unsmoothed + normalized + realigned (e.g., war*.nii)
   │  │  └─ sess-02/ ...
   │  └─ stat/             # First-level models (one folder per GLM)
   │     ├─ GLM-01/
   │     │  ├─ SPM.mat   <--------------------- [Select SPM.mat for first subject] (2)
   │     │  └─ TMFC_denoise/  <----------------------------------- [Output folder]
   │     └─ GLM-02/ ...
   └─ sub-02/ ...          <-------------------------- [Select subject folder #2] (1)
```

4.2 Example 2 — BIDS-like Folder Structure

1. Select the subject folders (each containing a `STAT` subfolder with an `SPM.mat` file).
2. Select the `SPM.mat` file for the first subject.

```
project/
├── sub-01/
│   ├── ses-01/
│   │   ├── anat/
│   │   │   └── *T1*.nii
│   │   └── func/         # Unprocessed functional files
│   └── ses-02/ ...
├── sub-02/ ...
└── derivatives/
    ├── fmriprep/
    │   ├── sub-01/
    │   │   ├── ses-01/
    │   │   │   └── func/
    │   │   │       └── Preprocessed functional files:
    │   │   │           • smoothed + normalized + realigned
    │   │   │           • unsmoothed + normalized + realigned
```

(continues on next page)

(continued from previous page)

```
    │   │   └── ses-02/ ...
    │   └── sub-02/ ...
    └── firstlevel-spm/
        ├── sub-01/   <------------------------- [Select subject folder #1] (1)
        │   ├── GLM-01/
        │   │   ├── SPM.mat   <--------- [Select SPM.mat for first subject] (2)
        │   │   └── TMFC_denoise/  <----------------------- [Output folder]
        │   └── GLM-02/ ...
        └── sub-02/ ...   <--------------------- [Select subject folder #2] (1)
```

4.3 Example 3 — Other (Non-BIDS) Folder Structure

1. Select the subject folders (each containing a `STAT` subfolder with an `SPM.mat` file).
2. Select the `SPM.mat` file for the first subject.

```
project/
├─ rawdata/   # DICOM
├─ nifti/
   ├─ sub-01/
      ├─ anat/
         ├─ *T1*.nii
         └─ *T1*.nii derivatives
      └─ func/
         ├─ sess-01/
            ├─ Unprocessed functional files (*.nii)
            └─ Preprocessed functional files (*.nii):
               • smoothed + normalized + realigned
               • unsmoothed + normalized + realigned
         └─ sess-02/ ...
   └─ sub-02/ ...
└─ firstlevel-spm/
   ├─ sub-01/   <--------------------------------- [Select subject folder #1] (1)
      ├─ GLM-01/
         ├─ SPM.mat   <------------------- [Select SPM.mat for first subject] (2)
         └─ TMFC_denoise/   <--------------------------------- [Output folder]
      └─ GLM-02/ ...
   └─ sub-02/ ...   <---------------------------- [Select subject folder #2] (1)
```

CHAPTER

FIVE

DENOISING OPTIONS

At the next step, the `TMFC_denoise` GUI prompts the user to select denoising options:

To open *Denoising options* GUI independently of the main `TMFC_denoise` function, run:

```
options = tmfc_denoise_options_GUI;
```

Outputs:

```
options.motion = '24HMP';           % Alternatives: 6HMP, 12HMP
options.translation_idx = [1 2 3];  % For FSL/AFNI use [4 5 6]
options.rotation_idx = [4 5 6];     % For FSL/AFNI use [1 2 3]
options.rotation_unit = 'rad';      % For HCP/AFNI use 'deg'
options.head_radius = 50;           % Human head radius in [mm]
options.DVARS = 1;                  % DVARS calculation: 1 = enabled (default), 0 =␣
↪disabled
options.aCompCor = [5 5];           % [5 5] for fixed PCs aCompCor (Default: five WM␣
↪PCs, five CSF PCs);
                                    % [0.5 0.5] for aCompcor50;
                                    % [0 0] do not calculate aCompCor
options.aCompCor_ort = 1;           % Pre-orthogonalization: 1 = enabled (default), 0␣
↪= disabled
options.rWLS = 0;                   % rWLS estimation: 1 = enabled, 0 = disabled
options.spikereg = 0;               % Spike regression: 1 = enabled, 0 = disabled
options.spikeregFDthr = 0.5;        % Select FD threshold for spike regression
options.WM_CSF = 'none';            % Alternatives: 2Phys, 4Phys, 8Phys
options.GSR = 'none';               % Alternatives: GSR, 2GSR, 4GSR
options.parallel = 0;               % Parallel computations: 1 = enabled, 0 = disabled
```

5.1 Head Motion Parameters (HMP)

- Use standard six head motion parameters (**6HMP**).
- Add six temporal derivatives (**12HMP**).
- Add six temporal derivatives and 12 quadratic terms (**24HMP**) (*default*) (Satterthwaite et al., 2012).
- Note: Temporal derivatives are computed as backward differences (Van Dijk et al., 2012).

5.2 Framewise Displacement (FD)

- Specify the order of motion regressors in the SPM.Sess.C structure. *By default*, indices for **translational regressors are [1, 2, 3]** and for **rotational regressors are [4, 5, 6]**.
- **Note:** In SPM, the Human Connectome Project (**HCP**), and **fMRIPrep** the **first three** motion regressors are **translations**. In **FSL** and **AFNI**, the **first three are rotations**. Adding confound regressors in the SPM

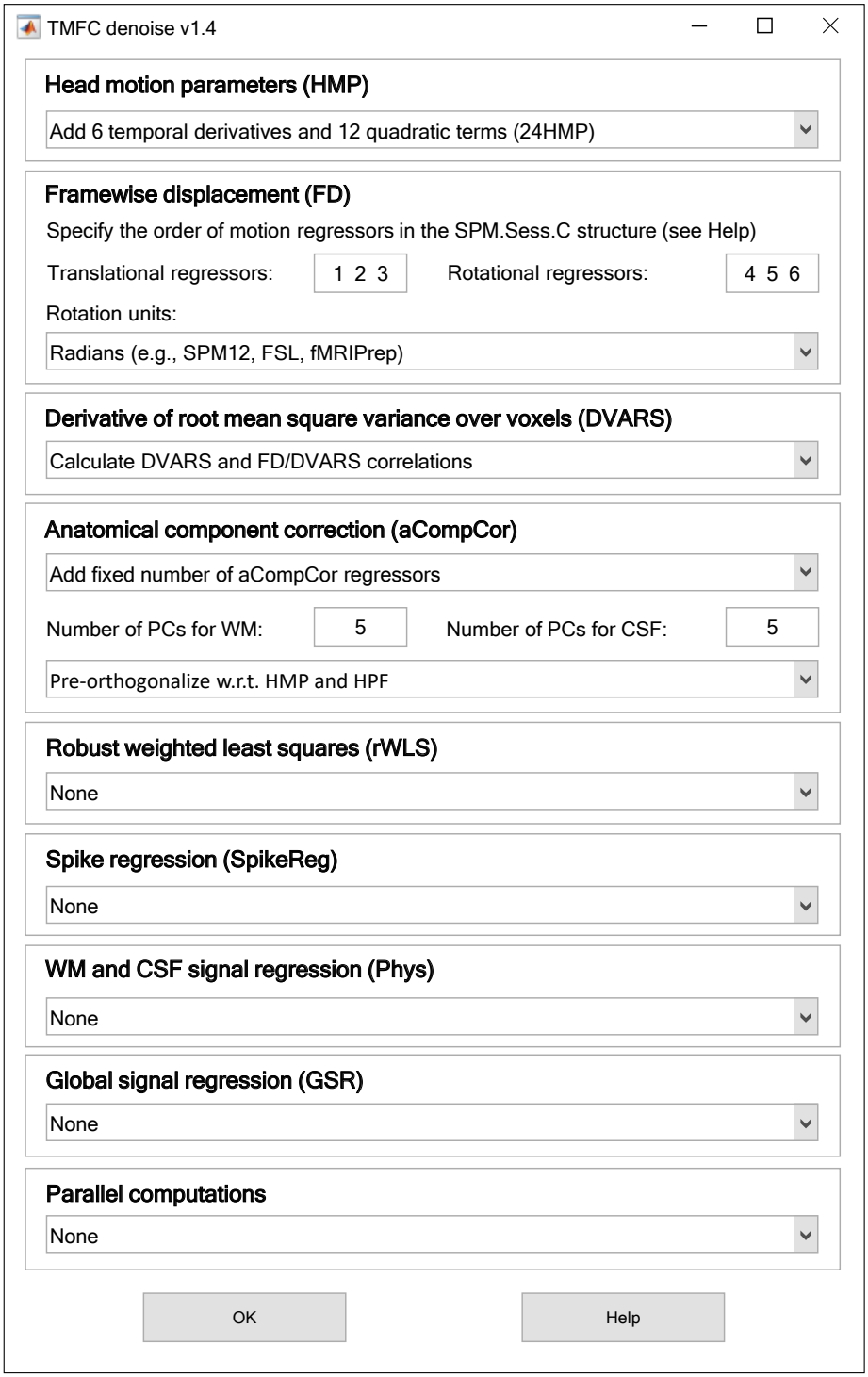

Fig. 1: Denoising options GUI.

batch using the "Regressors" option changes the indices of motion regressors defined with the "Multiple regressors" option (`*.txt/*.mat` files), as they appear last in `SPM.Sess.C`.

- Select rotation units: **radians** (for SPM, FSL, fMRIprep) or **degrees** (for HCP or AFNI).
- **Note:** FD is computed at each time point as the sum of the absolute values of the derivatives of translational and rotational motion parameters (Power et al., 2012).

5.3 Derivative of Root Mean Square Variance Over Voxels (DVARS)

- Calculate **DVARS** and **FD-DVARS correlations** (*default*).
- None (do not calculate DVARS).
- **Note:** DVARS is computed within the GM mask before and after denoising (Muschelli et al., 2014).

5.4 Anatomical Component Correction (aCompCor)

- Add a **fixed number of aCompCor regressors**. Specify the number of principal components (PCs) for WM and CSF (*default*: five for each).
- Add regressors explaining **50% of variance** in WM/CSF.
- **Note:** aCompCor extracts non-neuronal PCs from WM and CSF signals (Behzadi et al., 2007; Muschelli et al., 2014). It performs well in relatively low-motion samples, according to RSFC benchmarking (Parkes et al., 2017). *By default*, WM/CSF signals are **pre-orthogonalized** with respect to high-pass filter (**HPF**) regressors and head motion parameters (**HMP**) to ensure that the extracted PCs are maximally predictive (Mascali et al., 2021).

5.5 Robust Weighted Least Squares (rWLS)

- None (*default*) – the updated model uses the autoregression model specified in the original `SPM.mat` file (`none`, `AR(1)`, or `FAST`).
- Apply rWLS for model estimation.
- **Note:** In the first pass, the rWLS algorithm estimates the noise variance of each image. In the second pass, images are weighted by 1/variance rather than being excluded by an arbitrary threshold (as in spike regression, scrubbing, or despiking). This yields a "soft" continuous down-weighting: the higher an image's variance, the smaller its influence on the results (Diedrichsen and Shadmehr, 2005).

5.6 Spike Regression (SpikeReg)

- None (*default*).
- For each flagged time point, a unit impulse (1 at that time point, 0 elsewhere) is included as a spike regressor (Lemieux et al., 2007; Satterthwaite et al., 2012). Spike regression combined with WM/CSF regression performs well in high-motion samples, according to RSFC benchmarking (Parkes et al., 2017).
- **Note:** The default threshold for spike regression is 0.5 mm FD (Power et al., 2012). The FD threshold can be changed to more liberal values (e.g., 0.9 mm; Glasser et al., 2013) in the FD time-series inspection GUI.

5.7 WM and CSF Signal Regression (Phys)

- None (*default*).
- Add WM and CSF signals (**2Phys**) (Fox et al., 2005).
- Add WM and CSF signals along with their temporal derivatives (**4Phys**).
- Add WM and CSF signals, two temporal derivatives, and four quadratic terms (**8Phys**) (Parkes et al., 2017).

5.8 Global Signal Regression (GSR)

- None (*default*).
- Add whole-brain signal (**GSR**) (Fox et al., 2005, 2009).
- Add whole-brain signal and its temporal derivative (**2GSR**).
- Add whole-brain signal, its temporal derivative, and two quadratic terms (**4GSR**) (Parkes et al., 2017).
- **Note:** In RSFC studies, GSR remains controversial. It accounts well for head motion and physiological fluctuations of non-neuronal origin, but it may also remove BOLD signal fluctuations of neuronal origin (Chen et al., 2012) and can introduce spurious negative correlations (Murphy et al., 2008; Murphy & Fox, 2016).

5.9 Parallel Computations

- None (*default*).
- Enable parallel computations.
- **Note:** On low-end computers, parallelization may decrease overall speed. On mid-range systems, using fewer than the maximum available workers can increase computational efficiency (e.g. 2-4 workers instead of 8-16). To specify the number of workers, enter `parpool(NumWorkers)` in the MATLAB command window before running `TMFC_denoise`.

CHAPTER

SIX

SELECT STRUCTURAL IMAGES

If the user chooses to calculate tissue-based regressors and/or DVARS, the `TMFC_denoise` GUI prompts them to select unprocessed T1-weighted structural images (in native space).

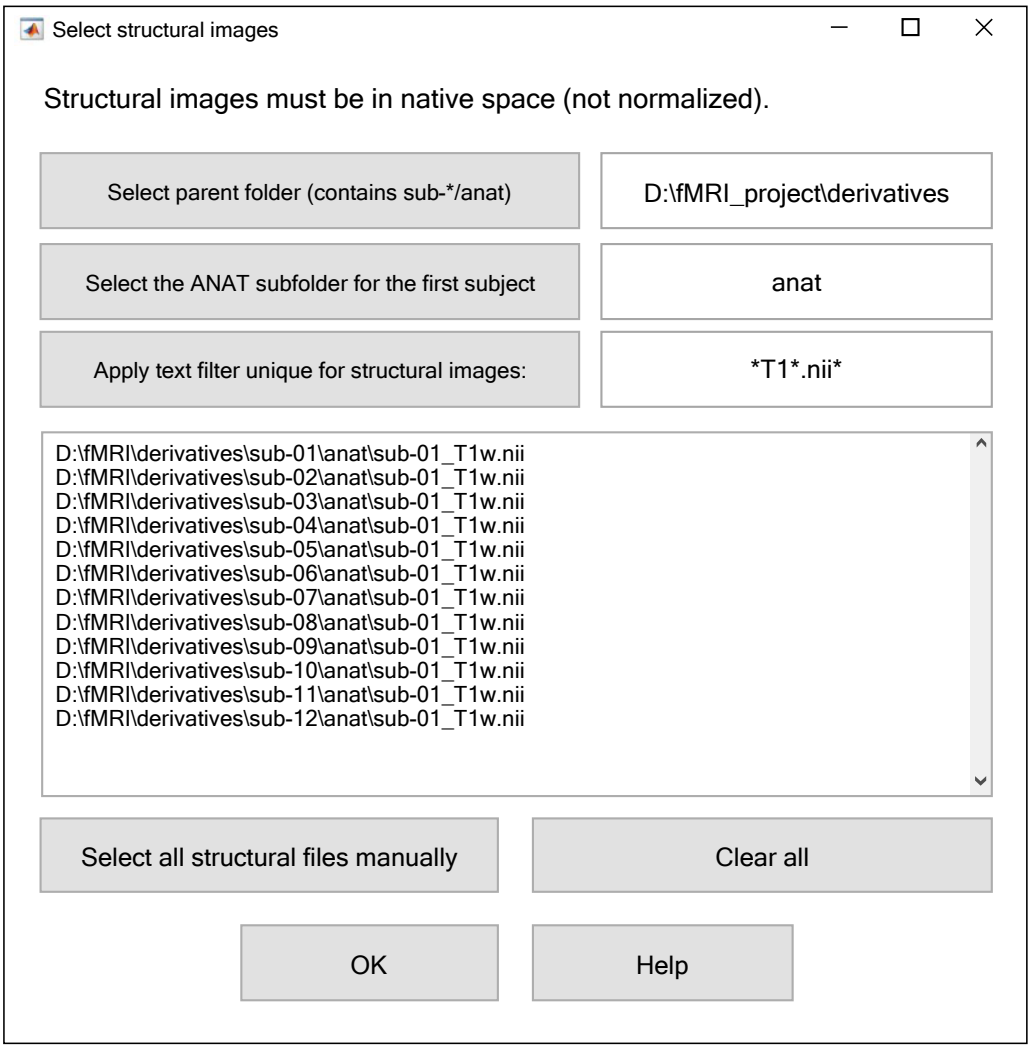

Fig. 1: Select structural images GUI.

First, select the parent folder that contains the subject folders with `ANAT` subfolders for unprocessed T1-weighted structural images (if necessary). By default, the parent folders for first-level GLMs and `ANAT` subfolders are assumed to be the same (in BIDS they differ).

Second, select the `ANAT` subfolder for the first subject. Third, apply a text filter (e.g., `*T1*.nii*`) to match all T1 images. Images may be in `*.img`/`*.hdr`, `*.nii` or `*.nii.gz` format.

Alternatively, all structural images can be selected manually — for example, when they are stored in a single folder. To do this, press: *Select all structural files manually*

To open *Select structural images* GUI independently of the main `TMFC_denoise` function, run:

```
anat_paths = tmfc_select_anat_GUI(subject_paths);
```

Output:

- `anat_paths` — Full paths to unprocessed (native-space) T1-weighted structural images (cell array)

6.1 Example 1 — SPM-like Folder Structure

In this case, there is no need to change the parent folder to select structural images.

```
project/
├─ rawdata/      # DICOM
└─ derivatives/  <-------- [Parent folder with ANAT subfolders (BY DEFAULT)]
   ├─ sub-01/    <---------------------------------- [Selected subject folder]
   │  ├─ anat/   <-------- [Select the ANAT subfolder for the first subject] (1)
   │  │  ├─ *T1*.nii  <---------------------------------- [Apply text filter] (2)
   │  │  └─ *T1*.nii derivatives (tissue seg., bias-corrected T1, etc.)
   │  ├─ func/
   │  │  ├─ sess-01/
   │  │  │  ├─ Unprocessed functional files (*.nii)
   │  │  │  └─ Preprocessed functional files:
   │  │  │       • smoothed + normalized + realigned (e.g., swar*.nii)
   │  │  │       • unsmoothed + normalized + realigned (e.g., war*.nii)
   │  │  └─ sess-02/ ...
   │  └─ stat/       # First-level models (one folder per GLM)
   │     ├─ GLM-01/
   │     │  ├─ SPM.mat    <------------------------- [Selected SPM.mat file]
   │     │  └─ TMFC_denoise/  <------------------------------ [Output folder]
   │     └─ GLM-02/ ...
   └─ sub-02/ ...
```

6.2 Example 2 — BIDS-like Folder Structure

1. Select the parent folder that contains all subject folders with `ANAT` subfolders (if necessary).
2. Select the `ANAT` subfolder for the first subject and apply text filter (e.g., `*T1*.nii`) to match all T1 images.

Here, the default parent folder `project/derivatives/firstlevel-spm` (with `STAT` subfolders) needs to be changed to `project` — the parent folder with `ANAT` subfolders.

```
project/   <------------------- [Select parent folder (contains sub-*/anat)]  (1)
├── sub-01/
│   ├── ses-01/
│   │   ├── anat/   <----- [Select the ANAT subfolder for the first subject]  (2)
│   │   │   └── *T1*.nii   <------------------------------ [Apply text filter]  (3)
│   │   └── func/         # Unprocessed functional files
│   └── ses-02/ ...
├── sub-02/ ...
└── derivatives/
    ├── fmriprep/
```

(continues on next page)

(continued from previous page)

```
    ├── sub-01/
    │   ├── ses-01/
    │   │   └── func/
    │   │       └── Preprocessed functional files:
    │   │           • smoothed + normalized + realigned
    │   │           • unsmoothed + normalized + realigned
    │   └── ses-02/ ...
    └── sub-02/ ...
└── firstlevel-spm/  <-- [Parent folder with ANAT (BY DEFAULT)](Needs to be␣
↪changed!)
    ├── sub-01/   <---------------------------- [Selected subject folder]
    │   ├── GLM-01/
    │   │   ├── SPM.mat   <---------------------- [Selected SPM.mat file]
    │   │   └── TMFC_denoise/  <------------------------ [Output folder]
    │   └── GLM-02/ ...
    └── sub-02/ ...
```

6.3 Example 3 — Other (Non-BIDS) Folder Structure

1. Select the parent folder that contains all subject folders with `ANAT` subfolders (if necessary).
2. Select the `ANAT` subfolder for the first subject and apply text filter (e.g., `*T1*.nii`) to match all T1 images.

Here, the default parent folder `project/firstlevel-spm` (with `STAT` subfolders) needs to be changed to `project/nifti` — the parent folder with `ANAT` subfolders.

```
project/
├── rawdata/   # DICOM
├── nifti/   <------------------- [Select parent folder (contains sub-*/anat)]  (1)
│   ├── sub-01/
│   │   ├── anat/  <---------- [Select the ANAT subfolder for the first subject]  (2)
│   │   │   ├── *T1*.nii  <---------------------------------- [Apply text filter]  (3)
│   │   │   └── *T1*.nii derivatives (tissue seg., bias-corrected T1, etc.)
│   │   └── func/
│   │       ├── sess-01/
│   │       │   ├── Unprocessed functional files (*.nii)
│   │       │   └── Preprocessed functional files (*.nii):
│   │       │       • smoothed + normalized + realigned
│   │       │       • unsmoothed + normalized + realigned
│   │       └── sess-02/ ...
│   └── sub-02/ ...
└── firstlevel-spm/  <-- [Parent folder with ANAT subfolders (BY DEFAULT)](Needs to be␣
↪changed!)
    ├── sub-01/   <---------------------------------- [Selected subject folder]
    │   ├── GLM-01/
    │   │   ├── SPM.mat   <------------------------------ [Selected SPM.mat file]
    │   │   └── TMFC_denoise/   <-------------------------------- [Output folder]
    │   └── GLM-02/ ...
    └── sub-02/ ...
```

CHAPTER

SEVEN

SELECT FUNCTIONAL IMAGES

If the user chooses to calculate tissue-based regressors and/or DVARS, the `TMFC_denoise` GUI prompts them to select unsmoothed preprocessed functional images (in MNI space).

Note: The updated GLMs are specified and estimated using the functional images listed in the original `SPM.mat` files (the `SPM.xY.VY` field). Thus, if smoothed images were used originally, they will be retained in the updated GLMs, while nuisance regressors will be calculated from the unsmoothed images. Although tissue-specific signals *can* be extracted from smoothed images (e.g., if unsmoothed data are unavailable), but unsmoothed images are preferred for noise-regressor creation.

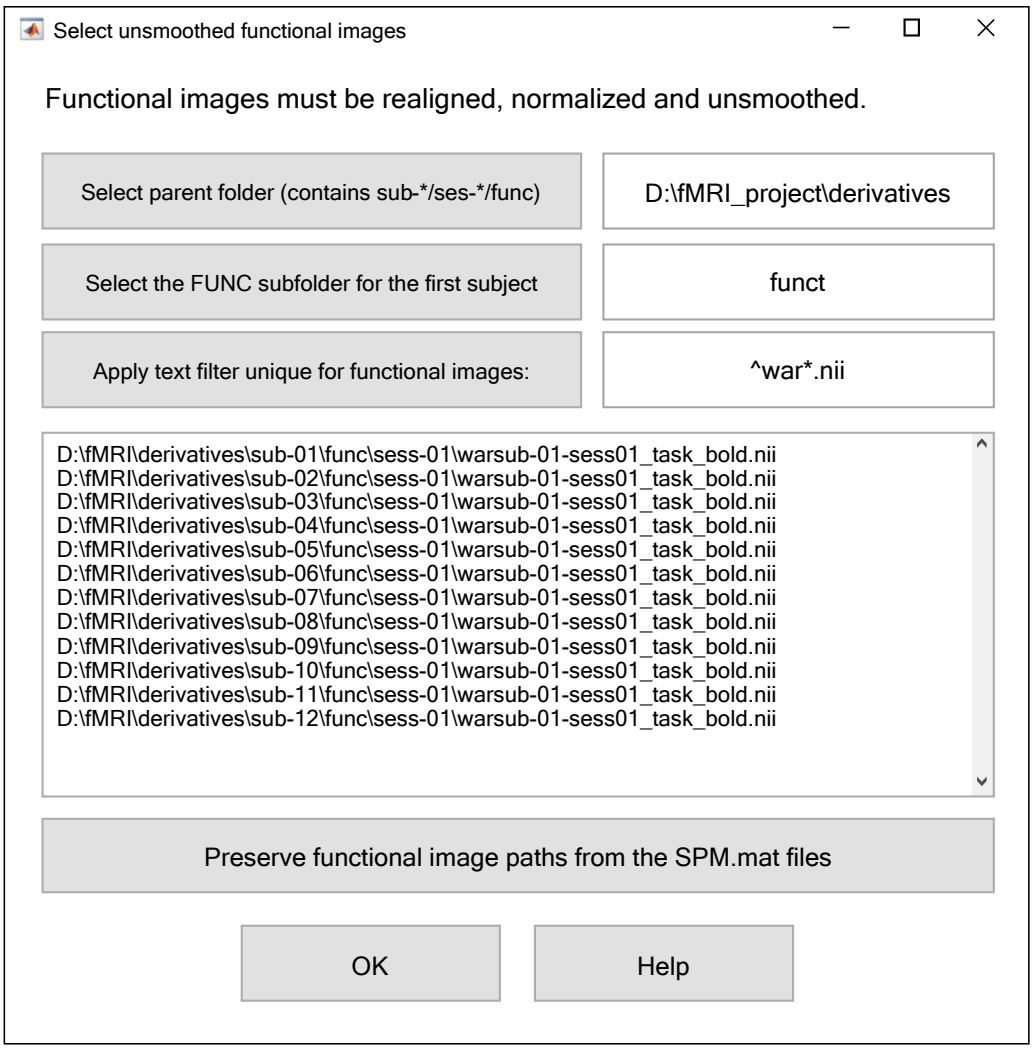

Fig. 1: Select functional images GUI.

First, select the parent folder that contains the subject folders with FUNC subfolders for preprocessed T2*-weighted functional images (if necessary). By default, the parent folders for first-level GLMs and FUNC subfolders are assumed to be the same (in BIDS they differ).

Second, select the FUNC subfolder for the first subject. This subfolder may include multiple session-specific subfolders (e.g., sess-01, sess-02). Third, apply a text filter (e.g., *war*.nii) to match all unsmoothed images. Images can be either *.img/*.hdr (3D) or *.nii (3D/4D). The *.nii.gz format is not supported.

If the GLMs were already specified for unsmoothed images, simply press: *Preserve functional image paths from the SPM.mat files*

To open *Select functional images* GUI independently of the main TMFC_denoise function, run:

```
func_paths = tmfc_select_func_GUI(SPM_paths,subject_paths);
```

Output:

- func_paths — Full paths to unsmoothed preprocessed (MNI-space, realigned) functional images (cell array)

7.1 Example 1 — SPM-like Folder Structure

In this case, there is no need to change the parent folder to select functional images.

```
project/
├─ rawdata/      # DICOM
└─ derivatives/  <------------------ [Parent folder with FUNC subfolders (BY DEFAULT)]
   ├─ sub-01/    <---------------------------------------- [Selected subject folder]
   │  ├─ anat/
   │  │  ├─ *T1*.nii
   │  │  └─ *T1*.nii derivatives
   │  ├─ func/  <------------------- [Select the FUNC subfolder for the first␣
↪subject] (1)
   │  │  ├─ sess-01/
   │  │  │  ├─ Unprocessed functional files (*.nii)
   │  │  │  └─ Preprocessed functional files:
   │  │  │       • smoothed + normalized + realigned (e.g., swar*.nii)
   │  │  │       • unsmoothed + norm. + real. (e.g., war*.nii) <-- [Apply text␣
↪filter] (2)
   │  │  └─ sess-02/ ...
   │  └─ stat/       # First-level models (one folder per GLM)
   │     ├─ GLM-01/
   │     │  ├─ SPM.mat    <----------------------------------- [Selected SPM.mat file]
   │     │  └─ TMFC_denoise/  <---------------------------------------- [Output folder]
   │     └─ GLM-02/ ...
   └─ sub-02/ ...
```

7.2 Example 2 — BIDS-like Folder Structure

1. Select the parent folder that contains all subject folders with FUNC subfolders (if necessary).
2. Select the FUNC subfolder for the first subject and apply text filter (e.g., *war*.nii, *wr*.nii, or *preproc*.nii.gz) to match all fMRI images.

Here, the default parent folder project/derivatives/firstlevel-spm (with STAT subfolders) needs to be changed to project/derivatives/fmriprep — the parent folder with FUNC subfolders.

```
project/
├── sub-01/
│   ├── ses-01/
│   │   ├── anat/
│   │   │   └── *T1*.nii
│   │   └── func/          # Unprocessed functional files
│   └── ses-02/ ...
├── sub-02/ ...
└── derivatives/
    ├── fmriprep/   <-------------- [Select parent folder (contains sub-*/ses-*/
↪func)] (1)
    │   ├── sub-01/
    │   │   ├── ses-01/
    │   │   │   └── func/  <-------- [Select the FUNC subfolder for the first␣
↪subject] (2)
    │   │   │       └── Preprocessed functional files:
    │   │   │           • smoothed + normalized + realigned
    │   │   │           • unsmoothed + normalized + realigned <--- [Apply text␣
↪filter] (3)
    │   │   └── ses-02/ ...
    │   └── sub-02/ ...
    └── firstlevel-spm/  <--- [Parent folder with FUNC (BY DEFAULT)](Needs to be␣
↪changed!)
        ├── sub-01/   <------------------------------------ [Selected subject folder]
        │   ├── GLM-01/
        │   │   ├── SPM.mat   <------------------------------ [Selected SPM.mat file]
        │   │   └── TMFC_denoise/  <---------------------------------- [Output folder]
        │   └── GLM-02/ ...
        └── sub-02/ ...
```

7.3 Example 3 — Other (Non-BIDS) Folder Structure

1. Select the parent folder that contains all subject folders with `FUNC` subfolders (if necessary).
2. Select the `FUNC` subfolder for the first subject and apply text filter (e.g., `*war*.nii`, `*wr*.nii`, or `*preproc*.nii.gz`) to match all fMRI images.

Here, the default parent folder `project/firstlevel-spm` (with `STAT` subfolders) needs to be changed to `project/nifti` — the parent folder with `FUNC` subfolders.

```
project/
├─ rawdata/   # DICOM
├─ nifti/   <---------------- [Select parent folder (contains sub-*/ses-*/func)]  (1)
│   ├─ sub-01/
│   │   ├─ anat/
│   │   │   ├─ *T1*.nii
│   │   │   └─ *T1*.nii derivatives
│   │   └─ func/    <----------- [Select the FUNC subfolder for the first subject]  (2)
│   │       ├─ sess-01/
│   │       │   ├─ Unprocessed functional files (*.nii)
│   │       │   └─ Preprocessed functional files (*.nii):
│   │       │      • smoothed + normalized + realigned
│   │       │      • unsmoothed + normalized + realigned  <----- [Apply text filter] (3)
│   │       └─ sess-02/ ...
│   └─ sub-02/ ...
```

(continues on next page)

(continued from previous page)

```
└─ firstlevel-spm/  <-- [Parent folder with FUNC subfolders (BY DEFAULT)](Needs to be␣
↪changed!)
   ├─ sub-01/   <------------------------------------ [Selected subject folder]
   │  ├─ GLM-01/
   │  │  ├─ SPM.mat   <---------------------------------- [Selected SPM.mat file]
   │  │  └─ TMFC_denoise/   <----------------------------------- [Output folder]
   │  └─ GLM-02/ ...
   └─ sub-02/ ...
```

CHAPTER

EIGHT

HMP EXPANSIONS AND FD PLOTS

The function `tmfc_head_motion` calculates HMP expansions and the FD time series. It is called automatically by the main function `TMFC_denoise` or can be run manually:

```
FD = tmfc_head_motion(SPM_paths,subject_paths,options);
```

The outputs are saved in the `TMFC_denoise` subfolder within each subject's first-level GLM directory: `12HMP.mat`, `24HMP.mat`, and `FD.mat`. The `FD.mat` file contains FD time series for each session, session-wise mean and maximum FD values, and mean/max FD across all sessions:

Table 1: FD.mat file

Field	Description
SPM_path	Full path to original SPM.mat file.
Subject	Subject folder name.
Sess (struct)	Session-wise FD data: • **FD_ts:** FD time-series per session • **mean:** Mean FD per session • **max:** Max FD per session
FD_mean	Mean FD across all sessions.
FD_max	Max FD across all sessions.

8.1 Framewise Displacement Plot

The GUI window for FD time-series inspection is opened with `tmfc_plot_FD`. It is called automatically by the main function `TMFC_denoise`. This interface allows users to change the FD threshold using the *Set FD threshold [mm]* button and to calculate the number of flagged time points exceeding this threshold. The selected threshold is subsequently applied during spike regression, if that denoising option is chosen.

To open FD plot GUI manually run:

```
% Allows saving group FD statistics only:
FDthr = tmfc_plot_FD(FD);

% Allows saving group FD statistics and TMFC denoise settings:
FDthr = tmfc_plot_FD(FD,options,SPM_paths,subject_paths,anat_paths,func_paths);
```

Pressing the *Save* button stores individual subject FD data and group-wise FD statistics in a single `*.mat` file:

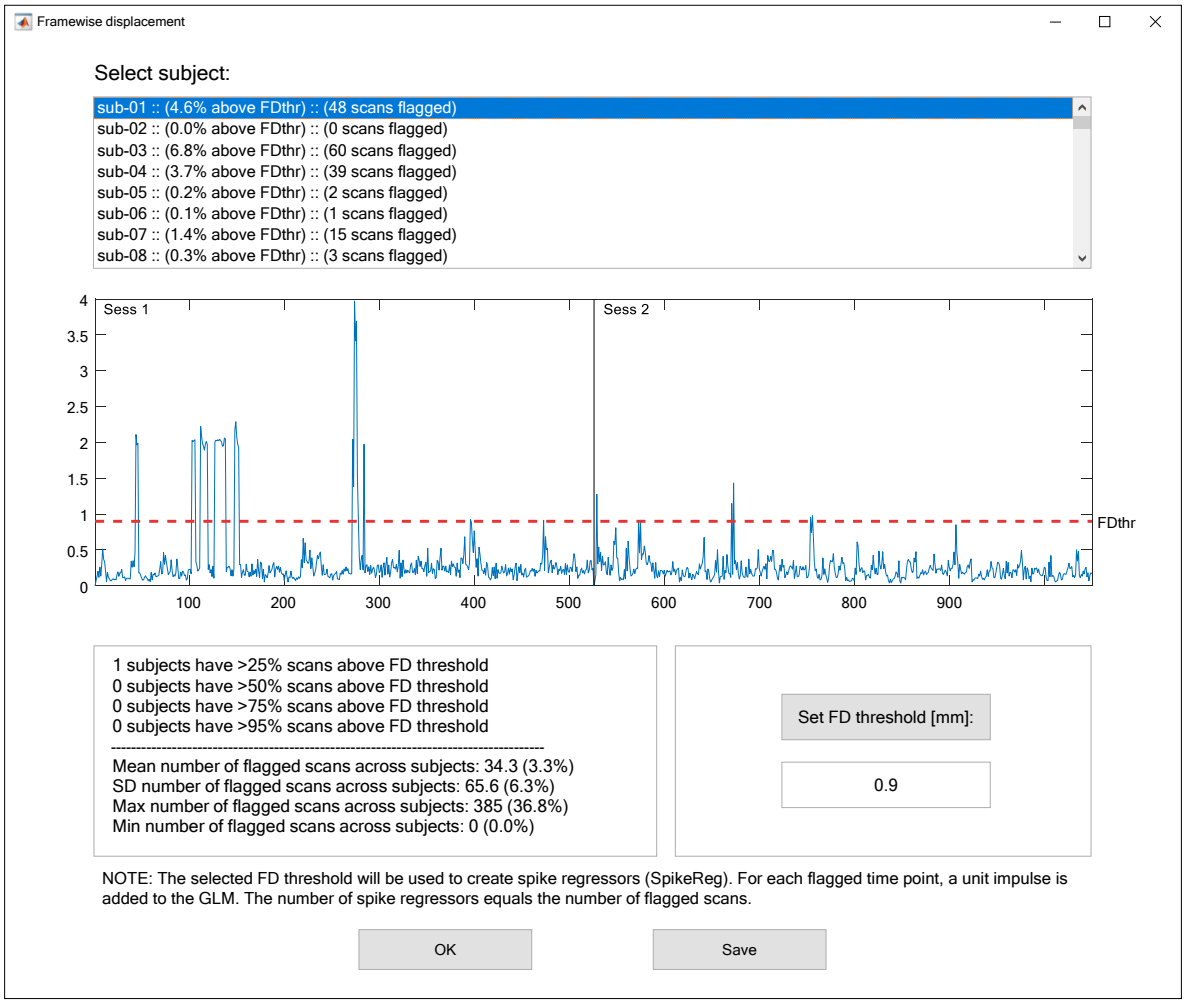

Fig. 1: FD plot GUI.

Table 2: Group_FD.mat file

Field	Description
denoising_settings (struct)	Selected TMFC_denoise settings: • **SPM_paths:** see *Select Subjects* • **subject_paths:** see *Select Subjects* • **options:** see *Denoising Options* • **anat_paths:** see *Select Structural Images* • **func_paths:** see *Select Functional Images*
FD (struct)	Individual FD data for all subjects (see FD.mat table).
FDthr	FD threshold (in millimeters).
flagged (struct)	Flagged time points for each subject. • **Sess:** Number of flagged time points per session • **total:** Total number of flagged time points • **total_prc:** Percentage of flagged time points (total)
max_flagged	Maximum number of flagged time points across subjects.
mean_flagged	Mean number of flagged time points across subjects.
min_flagged	Minimum number of flagged time points across subjects.
N_25prc	Number of subjects with >25% of scans above the selected FD threshold.
N_50prc	Number of subjects with >50% of scans above the selected FD threshold.
N_75prc	Number of subjects with >75% of scans above the selected FD threshold.
N_95prc	Number of subjects with >95% of scans above the selected FD threshold.
sd_flagged	SD number of flagged time points across subjects.

CHAPTER

NINE

SPIKE REGRESSION

The function `tmfc_spikereg` calculates spike regressors. The number of spike regressors equals the number of flagged time points. It is called automatically by the main function TMFC_denoise if the user has selected the SpikeReg option, or it can be run manually:

```
tmfc_spikereg(SPM_paths,options);
```

Outputs are saved in the `TMFC_denoise` subfolder within each subject's first-level GLM directory, with filenames of the form `SpikeReg_[FDthr_0.50mm].mat`, where `FDthr` denotes the selected FD threshold.

Note: Adding spike regressors can introduce negative FD-DVARS correlations. Future work should be devoted to understanding this cautionary behavior.

CHAPTER

TEN

MASK GENERATION

If the user chooses to calculate tissue-based regressors and/or DVARS, the `TMFC_denoise` GUI prompts them to define parameters for GM, WM, CSF, and whole-brain (WB) masks.

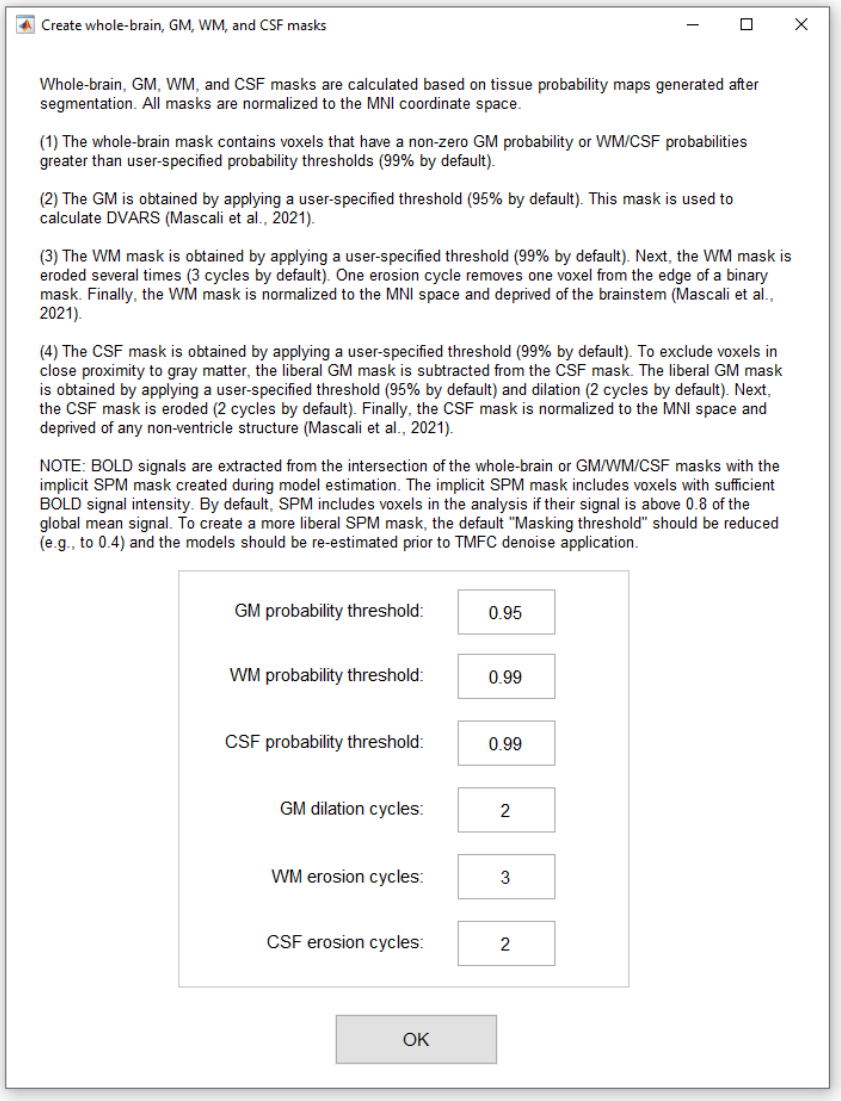

Fig. 1: Mask generation GUI.

To open *Mask parameters* GUI independently of the main `TMFC_denoise` function, run:

```
[options.GMmask.prob,   options.WMmask.prob,  options.CSFmask.prob, ...
 options.GMmask.dilate, options.WMmask.erode, options.CSFmask.erode] = tmfc_masks_
↪GUI();
```

Output:

```
options.GMmask.prob = 0.95;   % Probability threshold for GM mask
options.WMmask.prob = 0.99;   % Probability threshold for WM mask
options.CSFmask.prob = 0.99;  % Probability threshold for CSF mask
options.GMmask.dilate = 2;    % Number of dilation cycles for GM mask
options.WMmask.erode = 3;     % Number of erosion cycles for WM mask
options.CSFmask.erode = 2;    % Number of erosion cycles for CSF mask
```

The function `tmfc_create_masks` generates binary masks for GM (for DVARS), WM/CSF (for aCompCor and Phys regressors), and the whole brain (for GSR). The final masks and intermediate outputs are stored *by default* in the `TMFC_denoise/[WM99e3]_[CSF99e2]_[GM95d2]/Masks` subfolders, where `99` and `95` indicate the selected **probability thresholds**, `e2` refers to the number of **erosion cycles**, and `d2` to the number of **dilation cycles**.

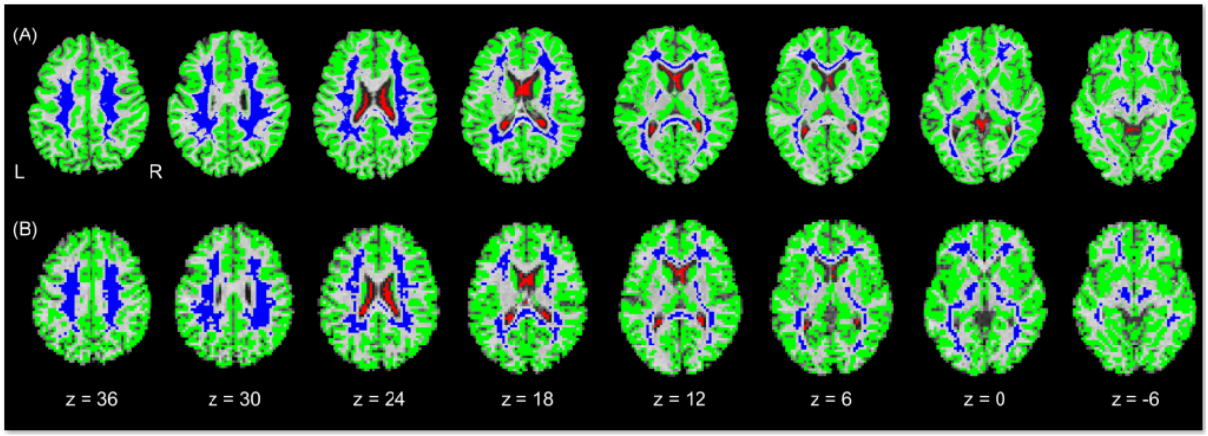

Fig. 2: Example of binary masks superimposed on the skull-stripped structural image for a single subject. (A) Masks in native space. (B) Masks in MNI space.

The function `tmfc_create_masks` is called automatically by the main function TMFC_denoise if the user has selected the corresponding options, or it can be run manually:

```
masks = tmfc_create_masks(SPM_paths,anat_paths,func_paths,options);
```

Output:

Table 1: Masks variable

Field	Description
glm_paths (cell array)	Full paths to mask-specific subfolders containing updated GLMs.
mask_paths (cell array)	Full paths to mask subfolders.
WM (cell array)	Full paths to the final WM masks.
CSF (cell array)	Full paths to the final CSF masks.
GM (cell array)	Full paths to the final GM masks.
WB (cell array)	Full paths to the final whole-brain (WB) masks.

10.1 Step-by-Step Mask Creation

Masks are computed as follows:

1) Segmentation of the unprocessed T1-weighted structural image. This step produces tissue-probability maps used to create individual binary masks (Ashburner & Friston, 2005). The outputs are stored in the TMFC_denoise/Segmentation subfolders and can be reused to generate binary masks with different parameters, without repeating the segmentation step.

2) Calculation of the eroded WM mask in native space. The WM mask is obtained by applying a user-specified probability threshold (default: WMprob = 99%). The WM mask is then eroded several times (default: three cycles). One erosion cycle removes a one-voxel layer from the edge of the binary mask.

3) Calculation of the eroded CSF mask in native space. The CSF mask is obtained by applying a user-specified probability threshold (default: CSFprob = 99%). To exclude voxels adjacent to GM, a liberal GM mask (default: GMprob = 95%, two dilation cycles) is subtracted from the CSF mask. The CSF mask is then eroded (default: two cycles).

4) Calculation of the GM mask in native space. The GM is obtained by applying a user-specified probability threshold (default: GMprob = 95%). Note: This mask is used for DVARS calculation (Mascali et al., 2021) and is not dilated.

5) Calculation of the whole-brain binary mask in native space. This mask contains voxels with nonzero GM probability or with WM/CSF probabilities above the specified thresholds. The equation for the whole-brain binary mask:

$$((GM > 0) + (WM > WM_{prob}) + (CSF > CSF_{prob})) > 0$$

6) Calculation of the skull-stripped structural image. This image is obtained by multiplying the bias-corrected structural image (mT1w) by a binary mask containing voxels with nonzero GM probability or WM/CSF probabilities above the thresholds. Equation for the skull-stripped structural image:

$$mT1w \cdot (((GM > 0) + (WM > WM_{prob}) + (CSF > CSF_{prob})) > 0)$$

7) Calculation of the final masks in MNI space. The eroded WM and CSF masks, the GM mask, and the skull-stripped structural image are normalized to MNI space, then resampled and coregistered to the first functional image. In the next step, the WM mask is deprived of the brainstem (Mascali et al., 2021) by subtracting the binary brainstem mask from the Harvard-Oxford atlas (Desikan et al., 2006). The CSF mask is restricted to the ventricles (Muschelli et al., 2014; Mascali et al., 2021) by multiplying with the binary ventricular mask from the Automatic Lateral Ventricle delIneatioN (ALVIN) atlas (Kempton et al., 2011). Finally, all binary masks are multiplied by the implicit SPM mask generated during model estimation (mask.nii), which includes voxels with sufficient BOLD signal intensity.

Note: By default, SPM retains voxels in the analysis if their signal is above **0.8** of the global mean. To create a more **liberal implicit mask**, the *Masking threshold* in the SPM first-level model batch can be reduced (e.g., to **0.4**), In this case, the models must be re-estimated before applying `TMFC_denoise`.

CHAPTER

ELEVEN

TISSUE-BASED NUISANCE REGRESSORS

The `tmfc_physioreg` function calculates tissue-based nuisance regressors. It is called automatically by the main function TMFC_denoise if the user has selected the corresponding options, or it can be run manually:

```
tmfc_physioreg(SPM_paths,subject_paths,func_paths,masks,options);
```

The outputs are saved in the `TMFC_denoise/[WM*e*]_[CSF*e*]_[GM*d*]` subfolders, where folder names encode the selected mask parameters (see *Mask Generation*):

- `WM*e*` — probability threshold and number of erosion cycles.
- `CSF*e*` — threshold and number of erosion cycles.
- `GM*d*` — threshold and number of dilation cycles.

Generated files include (depending on user-selected options, see *Denoising Options*):

- `2Phys.mat`, `4Phys.mat`, `8Phys.mat` — WM/CSF signals.
- `GSR.mat`, `2GSR.mat`, `4GSR.mat` — whole-brain signals.
- `[aCompCor_*WM_*CSF_Ort].mat` — a fixed number of WM and CSF principal components (PCs). These files also include information on the variance explained by WM/CSF PCs per session and the mean variance explained across sessions.
- `[aCompCor50_Ort].mat` — a variable number of PCs explaining 50% of WM/CSF signals variance. This file also reports the mean and total number of PCs for WM and CSF across sessions.

Note: When pre-orthogonalization of WM and CSF signals is enabled before PC extraction, the suffix `_Ort` is appended to all aCompCor output files.

CHAPTER

TWELVE

MODEL ESTIMATION

The `tmfc_estimate_updated_GLMs` function estimates updated GLMs with nuisance regressors. It is called automatically by the main function TMFC_denoise if the user has selected the corresponding options, or it can be run manually:

```
output_paths = tmfc_estimate_updated_GLMs(SPM_paths,masks,options);
```

The outputs are saved in the `TMFC_denoise/[WM*e*]_[CSF*e*]_[GM*d*]/GLM_*` subfolders, where folder names encode the selected denoising and masking parameters (see *Mask Generation*).

- `WM*e*` — probability threshold and number of erosion cycles.
- `CSF*e*` — threshold and number of erosion cycles.
- `GM*d*` — threshold and number of dilation cycles.

Each selected denoising option appends a corresponding suffix to the updated GLM subfolder (see *Denoising Options*). For example, `GLM_[24HMP]_[aCompCor50]_[rWLS]` indicates that the updated GLM includes 24 head-motion regressors, a variable number of aCompCor regressors explaining 50% of WM and CSF variance, and was estimated using rWLS.

The updated GLM subfolders contain the standard outputs from SPM model estimation, as well as `GLM_batch.m` files, which store `matlabbatch` structures that can be reopened in the SPM batch system.

The `SPM.mat` files in these subfolders can be used as input to the `TMFC toolbox`, which implements **gPPI and BSC-LSS methods** with or without FIR task regression (Masharipov et al., 2024). gPPI and LSS models automatically include nuisance regressors and, optionally, FIR regressors, along with high-pass filter regressors. Therefore, noise regression, FIR task co-activation regression (optional), and high-pass filtering are performed in a single step, which avoids reintroducing signal related to nuisance covariates (Lindquist et al., 2019).

CHAPTER

THIRTEEN

QUALITY CONTROL: DVARS AND FD-DVARS CORRELATIONS

The `tmfc_calculate_DVARS` function calculates DVARS within the GM mask before and after noise regression. It is called automatically by the main function `TMFC_denoise` if the user has selected the corresponding option, or it can be run manually:

```
[preDVARS,postDVARS] = tmfc_calculate_DVARS(FD,SPM_paths,options,masks,output_paths);
```

The outputs are saved in `TMFC_denoise/[WM*e*]_[CSF*e*]_[GM*d*]` subfolders:

- `DVARS_before_denoising.mat` – contains DVARS time series for each session before noise regression, session-wise FD-DVARS correlations, and mean/max FD-DVARS correlation across sessions.
- `DVARS_*.mat` – contains the same information as the previous file, but after noise regression. Filenames encode the selected denoising options (e.g., `DVARS_[24HMP]_[aCompCor50]_[rWLS].mat`).

The GUI window for DVARS time-series inspection is opened with `tmfc_plot_DVARS`. It is called automatically by the main function `TMFC_denoise`.

To open DVARS plot GUI manually run:

```
% Allows saving group FD-DVARS statistics only:
tmfc_plot_DVARS(preDVARS,postDVARS,FD);

% Allows saving group FD-DVARS statistics and TMFC denoise settings:
tmfc_plot_DVARS(preDVARS,postDVARS,FD,options,SPM_paths,subject_paths,anat_paths,func_
↪paths,masks);
```

Pressing the *Save* button stores individual subject FD and DVARS data, as well as group-wise DVARS statistics, in a single `*.mat` file:

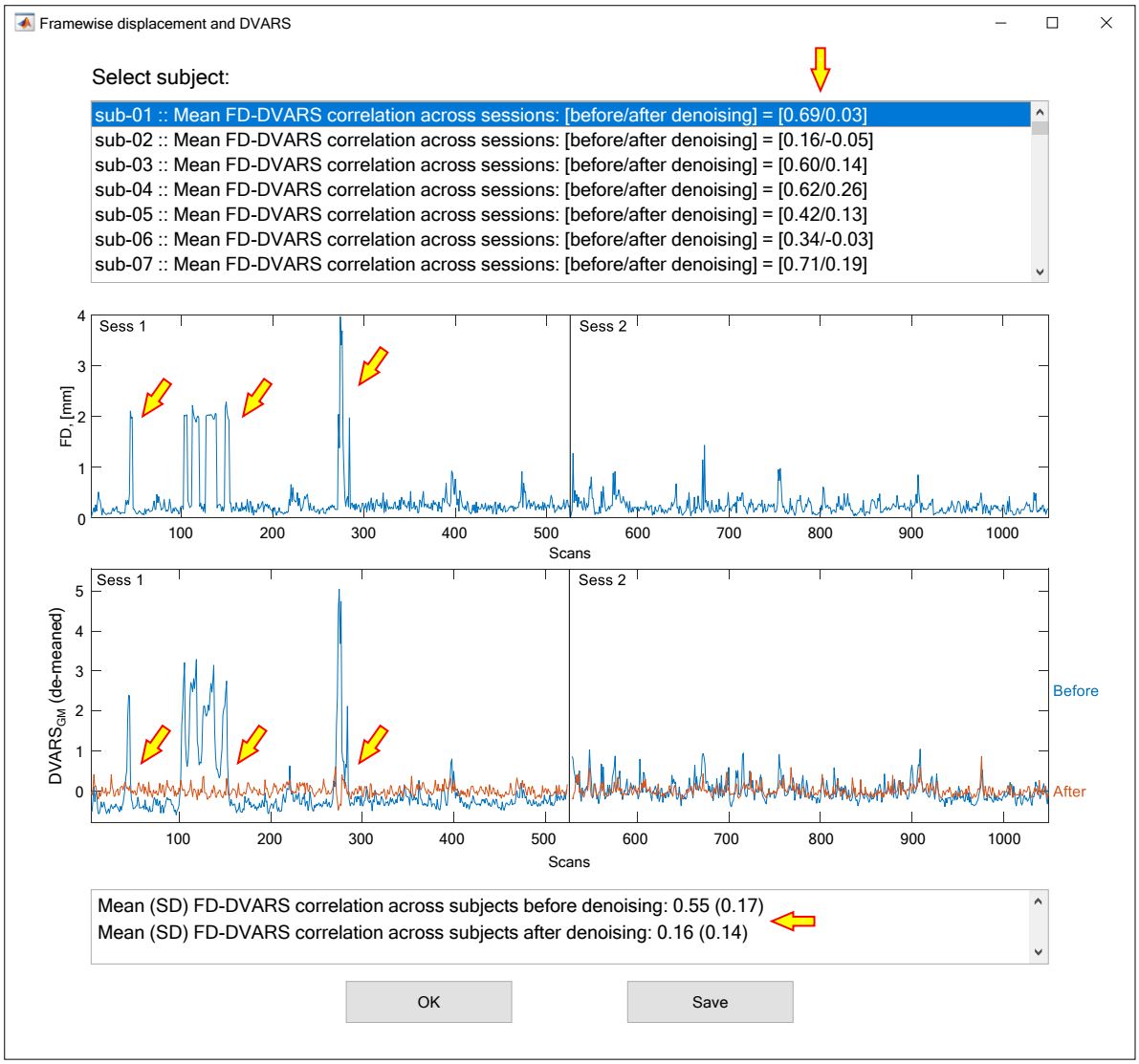

Fig. 1: Graphical interface for DVARS time-series inspection. Example DVARS plot for a single subject. The FD-DVARS correlation was reduced from 0.69 to 0.03 after noise regression. Spikes in the DVARS time series during the first session, associated with high-motion events, were visibly diminished. At the group level, the mean FD-DVARS correlation was decreased toward zero.

Table 1: Group_FD_DVARS.mat file

Field	Description
denoising_settings (struct)	Selected TMFC_denoise settings: • **SPM_paths:** see *Select Subjects* • **subject_paths:** see *Select Subjects* • **options:** see *Denoising Options* • **anat_paths:** see *Select Structural Images* • **func_paths:** see *Select Functional Images* • **masks:** see *Mask Generation*
FD (struct)	Individual FD data for all subjects (see *HMP expansions and FD plots*).
group_mean_post_FI	Group mean FD-DVARS correlation after denoising.
group_mean_pre_FD	Group mean FD-DVARS correlation before denoising.
group_SD_post_FD_	Group SD of FD-DVARS correlation after denoising.
group_SD_pre_FD_I	Group SD of FD-DVARS correlation before denoising.
postDVARS (struct)	DVARS data for each subject (after denoising). • **Sess:** Incluse DVARS time series and FD-DVARS correlation for each session. • **Mean_FD_DVARS_corr:** Mean FD-DVARS correlation across sessions. • **Max_FD_DVARS_corr:** Maximum FD-DVARS correlation across sessions.
preDVARS (struct)	DVARS data for each subject (before denoising). • **Sess:** Incluse DVARS time series and FD-DVARS correlation for each session. • **Mean_FD_DVARS_corr:** Mean FD-DVARS correlation across sessions. • **Max_FD_DVARS_corr:** Maximum FD-DVARS correlation across sessions.

These values can be reported to demonstrate the effectiveness of noise regression. If denoising is successful, spikes in the DVARS time series at high-motion time points should be reduced, and the FD-DVARS correlation should approach zero.

CHAPTER

FOURTEEN

COMMAND LINE USAGE

All functions in **TMFC_denoise** can be executed directly from the MATLAB command line or used in custom scripts without the GUI.

Example scripts demonstrating typical command-line usage are available in the examples folder of the TMFC_denoise GitHub repository:

- `prepare_auditory_dataset.m` — prepares the **auditory fMRI dataset** from the SPM website: http://www.fil.ion.ucl.ac.uk/spm/data/auditory/
- `prepare_facerep_dataset.m` — prepares the **face repetition fMRI dataset** from the SPM website: http://www.fil.ion.ucl.ac.uk/spm/data/face_rep/
- `TMFC_denoise_Example_01_Auditory_dataset.m` — example of TMFC_denoise usage (**block design**, auditory dataset).
- `TMFC_denoise_Example_02_Facerep_dataset.m` — example of TMFC_denoise usage (**event-related design**, face repetition dataset).

These scripts illustrate the full processing workflow, including dataset preparation, mask generation, denoising, and quality-control steps. They can be used as templates for adapting **TMFC_denoise** to other fMRI studies.

CHAPTER

FIFTEEN

FAQ

If your question is not covered here, please refer to the GitHub Issues page or contact the developer directly at: masharipov@ihb.spb.ru

—

Q1. I only want to calculate motion metrics (FD and DVARS). Is that possible? Yes. In *Denoising options*, set all denoising options to `none` and select `6HMP` for head motion. This will compute FD and FD-DVARS correlations (if DVARS calculation is enabled) without adding regressors to the model.

—

Q2. Can TMFC_denoise be used for resting-state fMRI? Yes. While optimized for task-based analyses, TMFC_denoise can also denoise resting-state data organized in SPM format. Simply skip task regressors in your GLM.

—

Q3. What if I have data preprocessed with fMRIPrep, HCP, or FSL? TMFC_denoise is compatible with GLMs created in SPM using externally preprocessed data. Adjust the order of motion regressors (`translation_idx` and `rotation_idx`) and rotation units (`deg` or `rad`) in the *Denoising options* window accordingly (see *Denoising Options*).

—

Q4. Does TMFC_denoise work only with the TMFC toolbox? No. TMFC_denoise can also be used for task-activation analyses or other purposes independently of the TMFC toolbox (see *Overview*).

—

Q5. How can I resume a partially finished run? TMFC_denoise saves intermediate outputs (e.g., masks, FD, aCompCor) in subject subfolders. If processing was interrupted, simply re-run the same command — previously completed subjects will be skipped automatically.

—

Q6. Do I need to recompute masks or regressors each time I change options? No. Segmentation and already created masks with the same parameters are not recomputed. Similarly, previously generated HMP expansions and tissue-based regressors for a given mask configuration are reused automatically.

—

Q7. My GLM already contains nuisance regressors (except motion parameters). That's fine. Check the indices of motion regressors in `SPM.Sess.C` and specify them in *Denoising options*. All existing regressors from the original model will be preserved in the updated GLM.

—

Q8. My model includes time/dispersion derivatives and parametric modulators. No problem — these regressors will remain in the updated model after denoising.

—

Q9. I don't have unsmoothed preprocessed functional images. Can I still run TMFC_denoise? Yes, although using unsmoothed images for tissue-based regressors is recommended. Smoothed data can still be used, but gray

matter contamination may slightly reduce the specificity of WM/CSF regressors. This effect is minimized by erosion of WM and CSF masks.

—

Q10. My sessions were concatenated using ``spm_fmri_concatenate``. Can I use TMFC_denoise? Yes. TMFC_denoise fully supports concatenated multi-session models created with `spm_fmri_concatenate`.

—

Q11. My model was already estimated with rWLS. Should I select the rWLS option again? You should use models estimated without rWLS (`none`, `AR(1)`, or `FAST`). If your model was previously estimated with rWLS but you do *not* select the rWLS option, the updated model will default to AR(1). To apply rWLS again, enable the rWLS option explicitly.

—

Q12. My eroded WM/CSF mask contains only a few voxels. Is that acceptable? Consider using a more liberal probability threshold and reducing the number of erosion cycles. You may also decrease the number of GM dilation cycles, since the GM mask is subtracted from the CSF mask.

—

Q13. Can I use functional images in native space? For tissue-based regressors and DVARS, no — they must be in MNI space. However, you can still perform HMP expansions, rWLS estimation, and FD calculation on native-space data.

—

Q14. What are the best denoising options for gPPI and BSC analyses? There is currently no universally optimal denoising strategy for gPPI or BSC. Large-scale benchmarking is still needed. Select options empirically for your dataset and inspect FD–DVARS correlations — successful denoising should reduce the group mean correlation toward zero.

—

Q15. Can I use noise regressors from TMFC_denoise in other software? Yes. All generated nuisance regressors are saved in the subject's `TMFC_denoise` subfolder and can be used in other analysis frameworks.

—

CHAPTER

SIXTEEN

REFERENCES

Please cite TMFC_denoise as:

- Ruslan Masharipov. (2025). TMFC denoise (v1.4.1) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.17176264
- <Toolbox paper in review…>

If you use the TMFC toolbox, please cite:

- Masharipov, R., Knyazeva, I., Korotkov, A., Cherednichenko, D., & Kireev, M. (2024). Comparison of whole-brain task-modulated functional connectivity methods for fMRI task connectomics. Communications Biology, 7(1). https://doi.org/10.1038/s42003-024-07088-3

Other sources

Statistical Parametric Mapping (SPM):

- Penny, W., Friston, K., Ashburner, J., Kiebel, S., & Nichols, T. (2011). Statistical Parametric Mapping: The Analysis of Functional Brain Images. Academic Press.

Robust weighted least squares (rWLS):

- Diedrichsen, J., & Shadmehr, R. (2005). Detecting and adjusting for artifacts in fMRI time series data. NeuroImage, 27(3), 624–634. https://doi.org/10.1016/j.neuroimage.2005.04.039

Head motion parameter (HMP) expansions:

- Friston, K. J., Williams, S., Howard, R., Frackowiak, R. S. J., & Turner, R. (1996). Movement-Related effects in fMRI time-series. Magnetic Resonance in Medicine, 35(3), 346–355. https://doi.org/10.1002/mrm.1910350312
- Satterthwaite, T. D., Elliott, M. A., Gerraty, R. T., Ruparel, K., Loughead, J., Calkins, M. E., Eickhoff, S. B., Hakonarson, H., Gur, R. C., Gur, R. E., & Wolf, D. H. (2012). An improved framework for confound regression and filtering for control of motion artifact in the preprocessing of resting-state functional connectivity data. NeuroImage, 64, 240–256. https://doi.org/10.1016/j.neuroimage.2012.08.052
- Van Dijk, K. R., Sabuncu, M. R., & Buckner, R. L. (2012). The influence of head motion on intrinsic functional connectivity MRI. NeuroImage, 59(1), 431–438. https://doi.org/10.1016/j.neuroimage.2011.07.044

Framewise displacement (FD):

- Power, J. D., Barnes, K. A., Snyder, A. Z., Schlaggar, B. L., & Petersen, S. E. (2012). Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. NeuroImage, 59(3), 2142–2154. https://doi.org/10.1016/j.neuroimage.2011.10.018

Spike regression (SpikeReg):

- Lemieux, L., Salek-Haddadi, A., Lund, T. E., Laufs, H., & Carmichael, D. (2007). Modelling large motion events in fMRI studies of patients with epilepsy. Magnetic Resonance Imaging, 25(6), 894–901. https://doi.org/10.1016/j.mri.2007.03.009

- Satterthwaite, T. D., Elliott, M. A., Gerraty, R. T., Ruparel, K., Loughead, J., Calkins, M. E., Eickhoff, S. B., Hakonarson, H., Gur, R. C., Gur, R. E., & Wolf, D. H. (2012). An improved framework for confound regression and filtering for control of motion artifact in the preprocessing of resting-state functional connectivity data. NeuroImage, 64, 240–256. https://doi.org/10.1016/j.neuroimage.2012.08.052

Anatomical component correction (aCompCor):

- Behzadi, Y., Restom, K., Liau, J., & Liu, T. T. (2007). A component based noise correction method (CompCor) for BOLD and perfusion based fMRI. NeuroImage, 37(1), 90–101. https://doi.org/10.1016/j.neuroimage.2007.04.042
- Muschelli, J., Nebel, M. B., Caffo, B. S., Barber, A. D., Pekar, J. J., & Mostofsky, S. H. (2014). Reduction of motion-related artifacts in resting state fMRI using aCompCor. NeuroImage, 96, 22–35. https://doi.org/10.1016/j.neuroimage.2014.03.028

Pre-orthogonalization of tissue-specific signals before aCompCor:

- Mascali, D., Moraschi, M., DiNuzzo, M., Tommasin, S., Fratini, M., Gili, T., Wise, R. G., Mangia, S., Macaluso, E., & Giove, F. (2021). Evaluation of denoising strategies for task-based functional connectivity: Equalizing residual motion artifacts between rest and cognitively demanding tasks. Human Brain Mapping, 42(6), 1805–1828. https://doi.org/10.1002/hbm.25332

Derivative of Root Mean Square Variance Over Voxels (DVARS):

- Power, J. D., Barnes, K. A., Snyder, A. Z., Schlaggar, B. L., & Petersen, S. E. (2012). Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. NeuroImage, 59(3), 2142–2154. https://doi.org/10.1016/j.neuroimage.2011.10.018
- Muschelli, J., Nebel, M. B., Caffo, B. S., Barber, A. D., Pekar, J. J., & Mostofsky, S. H. (2014). Reduction of motion-related artifacts in resting state fMRI using aCompCor. NeuroImage, 96, 22–35. https://doi.org/10.1016/j.neuroimage.2014.03.028

WM and CSF Signal Regression (Phys):

- Fox, M. D., Snyder, A. Z., Vincent, J. L., Corbetta, M., Van Essen, D. C., & Raichle, M. E. (2005). The human brain is intrinsically organized into dynamic, anticorrelated functional networks. Proceedings of the National Academy of Sciences, 102(27), 9673–9678. https://doi.org/10.1073/pnas.0504136102
- Parkes, L., Fulcher, B., Yücel, M., & Fornito, A. (2017). An evaluation of the efficacy, reliability, and sensitivity of motion correction strategies for resting-state functional MRI. NeuroImage, 171, 415–436. https://doi.org/10.1016/j.neuroimage.2017.12.073

Global Signal Regression (GSR):

- Fox, M. D., Snyder, A. Z., Vincent, J. L., Corbetta, M., Van Essen, D. C., & Raichle, M. E. (2005). The human brain is intrinsically organized into dynamic, anticorrelated functional networks. Proceedings of the National Academy of Sciences, 102(27), 9673–9678. https://doi.org/10.1073/pnas.0504136102
- Murphy, K., Birn, R. M., Handwerker, D. A., Jones, T. B., & Bandettini, P. A. (2008). The impact of global signal regression on resting state correlations: Are anti-correlated networks introduced? NeuroImage, 44(3), 893–905. https://doi.org/10.1016/j.neuroimage.2008.09.036
- Fox, M. D., Zhang, D., Snyder, A. Z., & Raichle, M. E. (2009). The global signal and observed anticorrelated resting state brain networks. Journal of Neurophysiology, 101(6), 3270–3283. https://doi.org/10.1152/jn.90777.2008
- Chen, G., Chen, G., Xie, C., Ward, B. D., Li, W., Antuono, P., & Li, S.-J. (2012). A method to determine the necessity for global signal regression in resting-state fMRI studies. Magnetic Resonance in Medicine, 68(6), 1828–1835. https://doi.org/10.1002/mrm.24201
- Murphy, K., & Fox, M. D. (2016). Towards a consensus regarding global signal regression for resting state functional connectivity MRI. NeuroImage, 154, 169–173. https://doi.org/10.1016/j.neuroimage.2016.11.052