MANUAL and DOCUMENTATION

Version 1.1.0
Date: May 12th 2025

Authors:
Andreas Hahn, Murray B Reed, Rupert Lanzenberger
andreas.hahn@meduniwien.ac.at
murray.reed@meduniwien.ac.at
rupert.lanzenberger@meduniwien.ac.at

Neuroimaging Labs
Department of Psychiatry and Psychotherapy
Medical University of Vienna, Austria
http://www.meduniwien.ac.at/neuroimaging/fPET.html

Table of Contents

1. SUPPORT

If you are uncertain how to design your fPET study, please contact us BEFORE running your scans (see below). The Neuroimaging Labs have successfully collaborated with [1-4] and provided support in the field of fPET imaging for numerous international labs e.g., in Sweden, Finland, Norway, Germany, Australia, USA, Denmark and Spain.

2. CONTACT

For problems, errors, bug reports, suggestions of further improvements and functions of the toolbox as well as questions about the design of your fPET study, please contact the developers at the Neuroimaging Labs, Dept. of Psychiatry and Psychotherapy, Medical University of Vienna, Austria:

andreas.hahn@meduniwien.ac.at
murray.reed@meduniwien.ac.at
rupert.lanzenberger@meduniwien.ac.at

3. HOW TO CITE

When using the fPET toolbox please cite the following manuscript:
https://www.biorxiv.org/content/10.1101/2024.11.13.623377v1

4. QUICK START

- Download the fPET toolbox, SPM and FastICA and add them to your Matlab path (including subfolders).
- Have your fPET data ready, most likely after preprocessing (e.g., motion correction, spatial normalization to MNI space and eventually smoothing).
- Have additional information ready, such as task timings, masks for baseline definition and calculation, etc.
- Choose the analyses that should be done and set up the variable `fpetbatch`, either with a Matlab script or by running the GUI with the command `fpet_tlbx_gui`.
- Refer to the chapter “Description of variables” for details on the different settings.
- Save the fpetbatch either as script or *.mat file.
- Run the toolbox, either with the respective button in the GUI or with the Matlab command `fpet_tlbx(fpetbatch)`.
- Check and interpret the results.
- Contact the developers if questions arise regarding the experimental design or analysis.

5. INTRODUCTION to fPET

As the name indicates, the fPET toolbox is designed for the analysis of functional PET data of the brain. That is, data i) which has been acquired with radiotracer application using a (bolus+)constant infusion protocol and ii) which has a rather high temporal resolution, usually between 1 and 60s. (note: the toolbox may also work with pure bolus application, other temporal resolutions and other organs, but it has not yet been tested with such data).

fPET has its origins in the desire to identify stimulation-induced changes in glucose metabolism with a single [^{18}F]FDG PET scan [5, 6]. Thus, the term fPET is used in analogy to fMRI, where a cognitive task is repeatedly carried out during the scan and the data is analyzed with a general linear model (GLM). This has been further expanded to the application of independent component analysis (ICA) as a data-driven alternative [7, 8]. Finally, high-temporal resolution fPET data enables the computation of molecular connectivity at the level of individuals [9, 10]. This represents a substantial advancement compared to group-level covariance, which is based on static image data.

The toolbox offers all of the above (and additional) approaches and thus provides great flexibility and different algorithms to analyze your data.

6. EXPERIMENTAL SETTINGS

The toolbox includes algorithms that have been successfully used to characterize stimulation induced changes of different

- cognitive tasks (visual and motor [6, 11, 12], Tetris [1, 3, 13], working memory [14], reward [15-17], optogenetic stimulation [4], resting-state [9])
- PET scanner systems (GE Advance [6, 11, 15], Siemens Biograph Vision mMR [1, 12, 13], 600 [9] and Quadra, Brain Biosciences CerePET, Bruker small-animal PET insert for 7T MRI ClinScan [4])
- radioligands ([^{18}F]FDG, 6-[^{18}F]FDOPA [15], [^{11}C]AMT [16])
- species (humans, non-human primates, rodents [4]).

Similarly, connectivity and covariance computations have already been used in previous work [9, 18-21].

NOTE: As the majority of fPET studies are performed with [^{18}F]FDG, the default parameter values are provided for this radioligand. If you are uncertain how to adapt this for other radioligands, please contact the developers.

7. INSTALLATION

Download the toolbox from Github at
https://github.com/NeuroimagingLabsMUV/fPET-toolbox.git

Then, unzip the files and add the directory (and subfolders) to your Matlab path.
The toolbox uses a few functions from SPM12 (https://www.fil.ion.ucl.ac.uk/spm/), which implies that this also needs to be installed (see dependencies).
When using ICA, the FastICA package (http://research.ics.aalto.fi/ica/fastica/index.shtml) also needs to be added to the Matlab path.

8. USAGE

To analyze data with the toolbox, a Matlab variable named 'fpetbatch' is set up. In this variable, all required inputs are defined. This includes the analysis to run…

- General linear model (GLM) and its subfunctions
  - Percent signal change (PSC)
  - Absolute quantification
  - Plotting of time activity curves
- Independent component analysis (ICA)
- Molecular connectivity
- Molecular covariance

…the image data and related data

- 4D fPET input
- 3D masks
- brain atlas
- blood data

…and various additional options

- stimulation regressors, motion parameters and additional regressors
- filter settings

The variable `fpetbatch` can be defined in a script or a graphical user interface (GUI). Afterwards, the main function `fpet_tlbx.m` is called with `fpetbatch` as input parameter, i.e., `fpet_tlbx(fpetbatch)`. All further subfunctions are called internally. Each analysis routine sets/calculates default variables based on the provided input and stores them in a *.mat file, such as `fPET_glm.mat`, `fPET_ica.mat`, `fPET_conn.mat`, etc.
It is good practice to save the file fpetbatch either as a *.mat file or as a script, as this will help in troubleshooting and documentation.

Imaging inputs are given as nifti files (*.nii), other parameter files as tab-separated text files (*.txt). All imaging inputs needs to have the same voxel dimensions, orientation, etc.

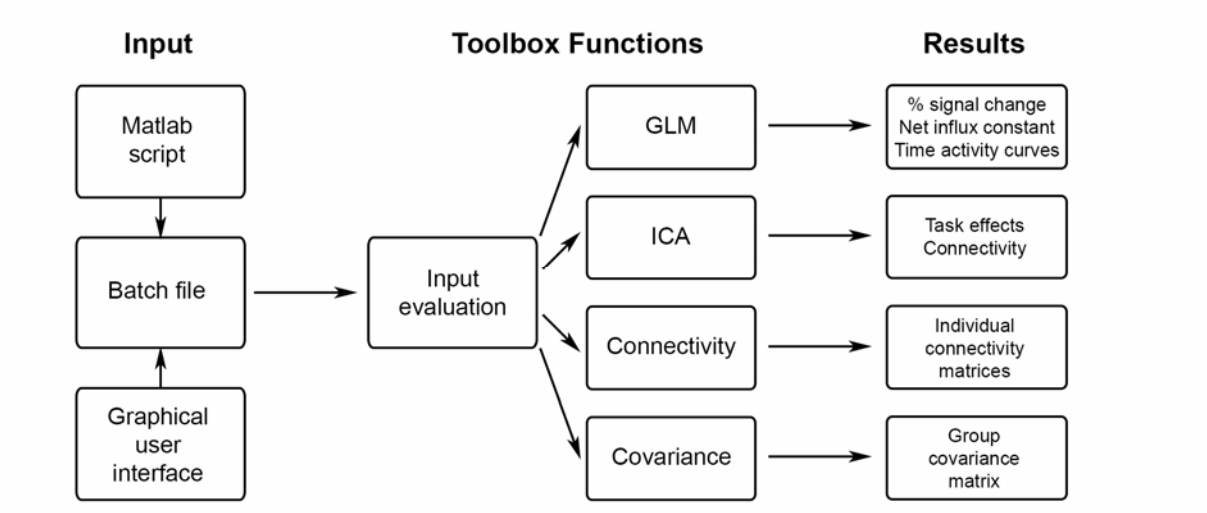

***Figure 1: Schematic overview of the toolbox**. The batchfile is created via a Matlab script or the GUI. Calling the fPET toolbox with fpet_tlbx(fpetbatch) evaluates the input, carries out the specified calculations such as GLM, ICA as well as Molecular Connectivity or Covariance and saves the results in the specified directory.*

9. GRAPHICAL USER INTERFACE (GUI)

The Graphical User Interface (GUI) can be started in Matlab once the toolbox and its dependencies have been added to the Matlab path.
To open the GUI please enter the following command: `fpet_tlbx_gui`. Thereafter, the fPET toolbox GUI will be open and load all default values.

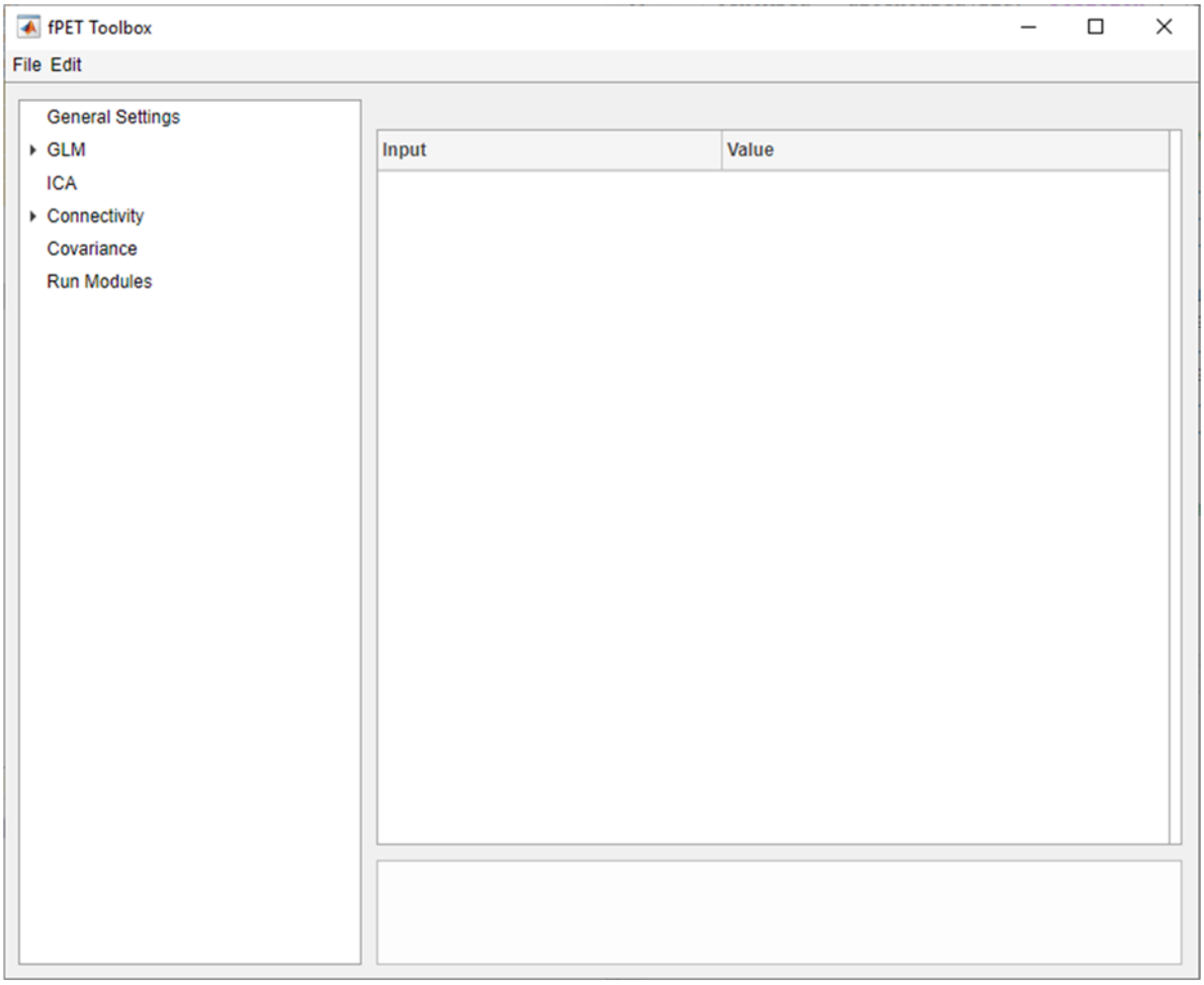

***Figure 2: Start screen of the fPET toolbox Graphical User Interface**. On the left side of the GUI, the general settings and submenu points for each model are listed as a tree and can be selected by clicking on the menu point or module of choice.*

After selecting a module to estimate e.g. the GLM, a list of submenus and corresponding parameters will be listed in the right window of the GUI. After clicking on a parameter, a description of the variable will be displayed in the bottom window. Each parameter includes the variable name, which matches the name and corresponding details in the section “Description of variable” (figure 3, red circle). All fields with the [*] suffix indicate that the field is mandatory and must be filled out for the toolbox to run. This is also displayed in the bottom window the [Mandatory] tag.

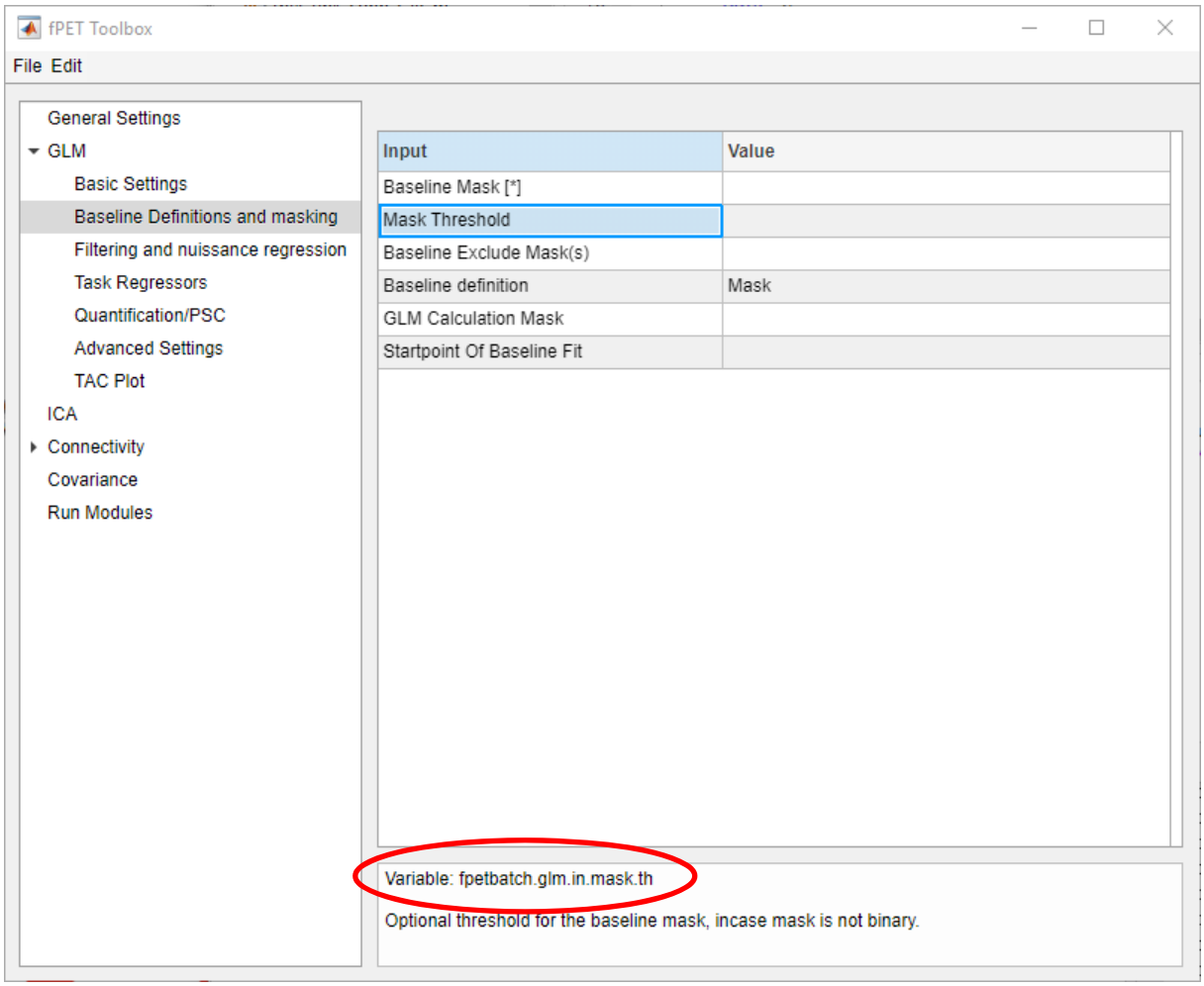

Figure 3: Input parameters of the GLM module.

To fill out each point simply double click on the value field for the variable of choice. Another window will appear, allowing the user to input the data in the correct format.

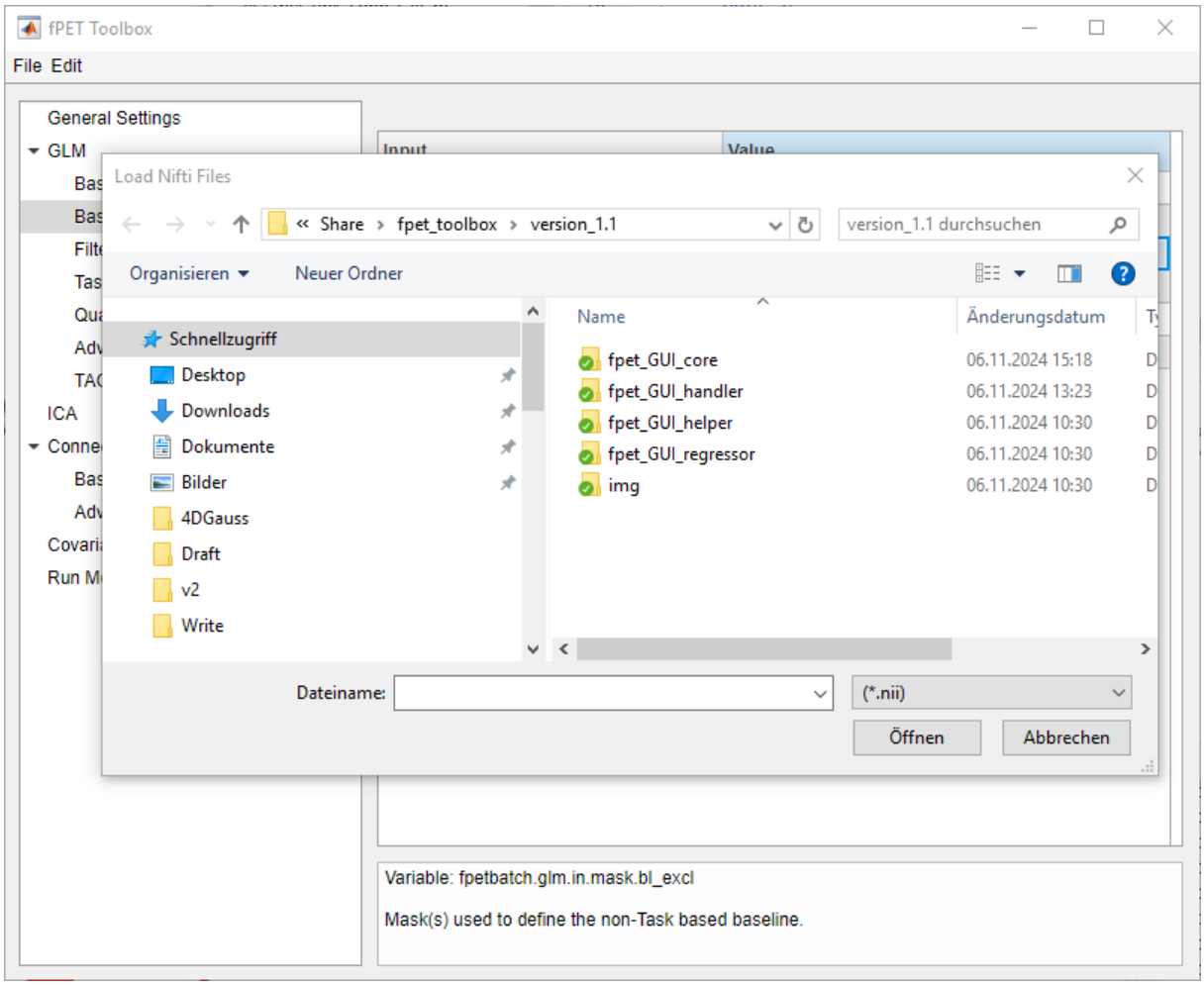

Figure 4: Input window for imaging input data for the GLM module.

After all mandatory options have been defined and the modules to be executed have been selected, the user can navigate to the “Run Modules” menu, where a summary of the status of all modules is listed. Here the user can check if the modules are correctly selected and then click the run button to begin the processing of the data.

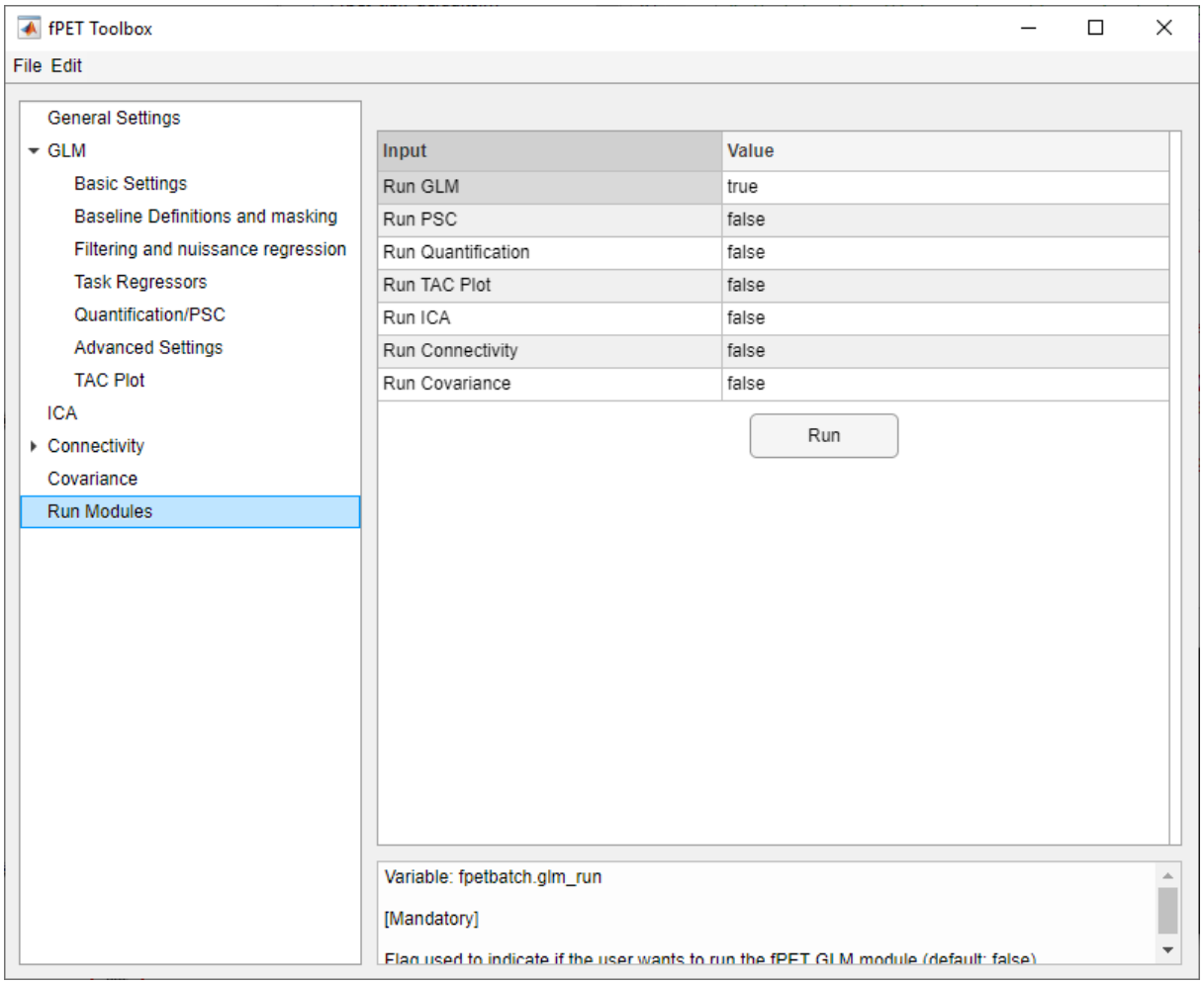

Figure 5: Run menu. *After all options have been entered for a selected module or multiple modules, the user can review the input and which modules should be run. Thereafter, the user can press the run button to process the data.*

All the inputs can be saved, reset, or already created batch files can be loaded utilizing the “File” menu. Finally, the code behind the GUI fpetbatch can be viewed via the “Edit” menu.

10. FUNCTIONALITY

All functions and variables will be explained in the following sections. A complete overview of all variables is also provided in the chapter “description of variables”.

10.1. General

If no results directory (`fpetbatch.dir.result`) is provided, the current working directory will be used. If data already exists in the given results directory the user will be asked if the existing data should be overwritten. This user input can be bypassed by setting the flag `fpetbatch.overwrite = 1`.

To call a specific analysis routine the corresponding flag needs to be set. Each of these functions can be called separately or within a single batch.

```
fpetbatch.run_glm = 1
fpetbatch.run_psc = 1
fpetbatch.run_quant = 1
fpetbatch.run_tacplot = 1
fpetbatch.run_ica = 1
fpetbatch.run_conn = 1
fpetbatch.run_cov = 1
```

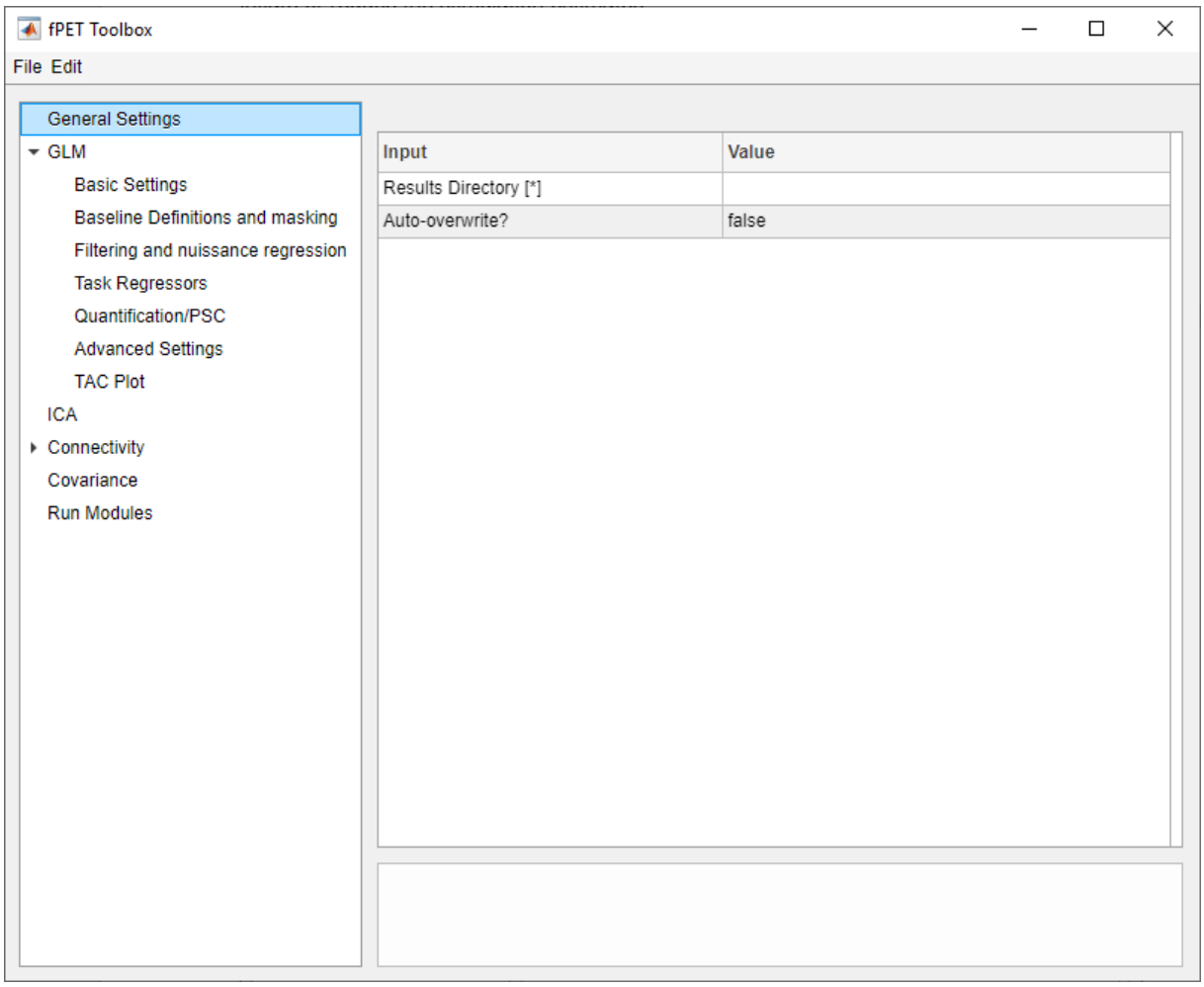

***Figure 6: General Settings**. After starting the fPET toolbox and selecting the General Settings module from the menu, the following screen should appear as above. The options for General Settings can be examined by clicking on them. A single click will display a text in the bottom text area with a short description of the variable selected. Options with the [*] and also listed in the help text as [Mandatory] are needed for the toolbox to run.*

10.2. Generar Linear Model (GLM)

The GLM is the main function to analyze task data based on a pre-defined model. The GLM aims to separate baseline and stimulation effects. The implementation has been described previously [6, 12] and applied in numerous publications [1, 3, 11, 13-16, 22]. The design matrix usually includes a stimulation regressor and potentially further regressors (motion parameters from SPM, non-standard stimulation regressors, nuisance variables, etc.). The regressors of the design matrix are sorted in the following order: constant, baseline, stimulation regressor 1…n, additional regressor 1…m, motion regressor 1…k.

Mandatory inputs for the GLM are 4D input data (`fpetbatch.glm.in.data`), the duration of each frame in seconds (`fpetbatch.glm.in.framelength`) and a definition if timing variables are provided in seconds (`fpetbatch.glm.in.time=1`) or frames (`fpetbatch.glm.in.time=2`). Also, a 3D mask for the baseline calculation needs to be defined (`fpetbatch.glm.in.mask.bl`) to extract an average TAC across all voxels within this mask (usually entire gray matter). This average TAC represents the baseline metabolism, which will be scaled during the GLM for each voxel [12].

Optional inputs for the GLM analysis are also available:
Mask files:
To speed up the processing, it is feasible to restrict calculations to certain voxels (usually gray matter) by providing a 3D mask (`fpetbatch.glm.in.mask.calc`).
To refine the above baseline mask it is possible to exclude voxels from by the baseline TAC definition by providing further 3D masks (`fpetbatch.glm.in.mask.bl_excl{ind_m}`). These usually include areas with expected stimulation-induced activation, e.g., obtained from individual fMRI analysis, meta-analysis, anatomical information, etc. [12]. All non-zero voxels in these masks will be removed from the above baseline mask. The user can also provide a threshold for each mask (`fpetbatch.glm.in.mask.th`) with the following conditions:

- If this threshold is 0, then all voxels < 0 are set to 0.
- If threshold ~= 0, then –threshold < voxel < threshold are set to 0 in the mask (e.g., for individual fMRI statistical results).
- If threshold = NaN, the mask remains unchanged.
- When multiple masks are used, then thresholds are required for all masks or none of them. The order of masks and thresholds must match.

Baseline model:
To define the baseline metabolism, two options are provided. By default, the average TAC is extracted from a mask (`fpetbatch.glm.in.bl_type=1`), usually gray matter with or without refinement by additional masks as described above (see `fpetbatch.glm.in.mask.bl`).
The other option is to model this TAC by a third order polynomial (`fpetbatch.glm.in.bl_type=2`) [6]. This is useful if regions with expected activation are unknown. If this option is chosen, the time point to start the model fit needs to be defined (`fpetbatch.glm.in.bl_start_fit`) in seconds or frames (see `fpetbatch.glm.in.time`). If a bolus+constant infusion protocol is used, a time point after the radiotracer uptake of the initial bolus should be chosen (e.g., after 5min) [12].
Adequate definition of the baseline metabolism is important as errors in this parameter may inflate or reduce the stimulation estimates.

Regressors:
Stimulation regressors are set by defining the start (`fpetbatch.glm.in.regr(ind_r).start`) and end (`fpetbatch.glm.in.regr(ind_r).end`) of each stimulation block/event. This information is provided in seconds or frames (see `fpetbatch.glm.in.time`). Also, a name for each regressor can be provided (`fpetbatch.glm.in.regr(ind_r).name`). By default, stimulus regressors are orthogonalized with respect to the baseline regressor

(fpetbatch.glm.in.regr_orth=1), i.e., the stimulus regressor explains variance on top of the baseline effect [6].

Motion parameters (fpetbatch.glm.in.regr_motion) and additional regressors (fpetbatch.glm.in.regr_add) can be added as text files. By default, motion regressors are subject to principal component analysis and the elbow criterion is used to identify the components that explain a sufficient amount of variance (fpetbatch.glm.in.regr_motion_pca=1) [14]. Additional regressors are required to have the same temporal dimension as the fPET data. For motion regressors see advanced options below.

Further options:
To reduce noise in the data, a FIR low pass filter can be specified (fpetbatch.glm.in.fil.apply, default = 1). The filter order can be set but only half of the desired order should be provided because the function filtfilt filters the data in forward and reverse direction (fpetbatch.glm.in.fil.order, default = 6, i.e., 12th order). The cutoff value for the filter (fpetbatch.glm.in.fil.cutoff) is given in seconds or frames (see fpetbatch.glm.in.time) with the default value being half of the task duration [6].

Advanced options
These usually depend on the data acquisition and modification should be considered carefully.

It is possible to remove initial (fpetbatch.glm.in.rem_start) or final fPET frames (fpetbatch.glm.in.rem_end) from the acquired data (as well as motion/additional regressors). This is useful if radiotracer application started after or ended earlier with respect to data acquisition. This information is provided in seconds or frames (see fpetbatch.glm.in.time).

Often, early fPET frames contain little information (particularly for constant infusion only) and no motion parameters may be available for the beginning of the scan. This can be taken into account by setting a flag (fpetbatch.glm.in.regr_motion_incomplete=1) to add initial zeros to the motion parameters. Similarly, zeros can be added at the end (fpetbatch.glm.in.regr_motion_incomplete=2).

In case the fPET acquisition is incomplete, this can be accounted for with a flag (fpetbatch.glm.in.data_incomplete.flag=1). This may occur if the fPET data acquisition started after radiotracer application and/or if the position was shifted between brain and heart (e.g., to obtain an image-derived input function for quantification) [17]. For this option the start (fpetbatch.glm.in.data_incomplete.start) and end times (fpetbatch.glm.in.data_incomplete.end) of the available data needs to be specified. These two variables are provided in seconds or frames (see fpetbatch.glm.in.time). Furthermore, the timings of the stimulus needs to be adjusted before calling the toolbox by the user.

Furthermore, it is possible to provide predefined weights for the GLM analysis (fpetbatch.glm.in.weight). This might be useful if certain data points are corrupted, etc. The weighting vector has the same length as the image data.

The **outputs** of the GLM are 3D maps of beta estimates and corresponding t-statistics for each regressor, as well as a map of residuals.

10.2.1. Set the timings correctly

Some of the inputs for the GLM require to adequately set the timings in order to match start of radiotracer application, start of data acquisition, stimulus timings, etc.

- **Generally:**
  It is assumed that the start of the radiotracer application coincides with the start of the data acquisition, which both occur at time 0s. In this case all temporal inputs (stim onset/offset, start of GLM and stimulus, etc.) relate to the start of radiotracer application.
- **When removing data:**
  The variable (`fpetbatch.glm.in.rem_start`) removes initial data to match the start of data acquisition with the start of radiotracer application, again making it the general case above. This is useful if data acquisition started earlier than radiotracer application. The variable (`fpetbatch.glm.in.rem_end`) removes final data after all other variables have been set by the toolbox. This is feasible if problems with radiotracer application, substantial movement, etc. occurred towards the end of the scan.
- **For incomplete data:**
  Start (`fpetbatch.glm.in.data_incomplete.start`) and end time (`fpetbatch.glm.in.data_incomplete.end`) of the actually available data must be provided. The timings of these still assume that radiotracer application occurred at time 0s and a full data acquisition would also start at time 0s. Furthermore, the timings of the stimulus needs to be matched with the actual data before calling the toolbox by the user.

10.3. Percent Signal Change (PSC)

The GLM only yields t-statistics and beta values, but these may be difficult to compare between different tasks, analyses, centers, etc. PSC can be calculated without the need for any further input and thus represents a non-invasive alternative to absolute quantification without the need of an (arterial) input function. This metric is basically calculated as the differences in the slopes between task and baseline effects, i.e., the stimulation-induced change as compared to baseline [23].
The calculation of PSC requires to run the GLM beforehand.
The only user input which is required for PSC is the flag to run the calculation (`fpetbatch.run_psc=1`) and the results directory of the GLM (`fpetbatch.dir.result`).

The **output** of this function is a 3D map of PSC for each stimulation regressor defined in the GLM.

10.4. Absolute Quantification

Absolute quantification of the outcome parameters are computed with the Gjedde-Patlak plot [6, 24], yielding the net influx constant Ki, and the cerebral metabolic rate of glucose CMRGlu (if applicable).

Mandatory inputs for absolute quantification are the flag to run the calculation (`fpetbatch.run_quant=1`) and the results directory of the GLM (`fpetbatch.dir.result`). Similar to the GLM, one needs to define if temporal inputs for quantification (not for other calculations!) are provided in seconds (`fpetbatch.quant.in.time=1`) or minutes (`fpetbatch.quant.in.time=2`). Also, at least the plasma input function needs to be specified (`fpetbatch.quant.in.plasma`).
The absolute quantification requires to run the GLM beforehand.

Optional inputs are the whole-blood input function (`fpetbatch.quant.in.wb`) and the plasma/whole-blood ratio (`fpetbatch.quant.in.pwbr`). These three inputs (plasma, wb and pwbr) are provided as text files with two columns. The first column is the time variable, the second the values in the same units as the fPET data (e.g., kBq/ml).
By default the pwbr is modeled as average of the provided data (e.g., for [^{18}F]FDG, `fpetbatch.quant.in.pwbr_fit=1`), but it can also be fitted with a linear function (e.g., 6-[18F]FDOPA and [11C]AMT, `fpetbatch.quant.in.pwbr_fit=2`).
Further inputs can be the starting value for the linear fit of the Patlak plot (`fpetbatch.quant.in.tstar`), provided as a fraction of the total scan duration (`default=1/3`) and the fractional whole-blood volume (`fpetbatch.quant.in.vb, default=0.05`).
If blood glucose levels are provided in mmol/L (`fpetbatch.quant.in.bloodlvl`), Ki is also converted to CMRGlu, using a default lumped constant of 0.89 (`fpetbatch.quant.in.lc`).

The **output** of this function is a 3D map of the net influx constant Ki (and CMRGlu, if applicable) for each stimulation regressor defined in the GLM.

10.5. Plotting Time Activity Curves (TACs)

To visualize stimulation effects, it might be desired to plot the stimulation-specific TAC. This is done by subtracting all non-relevant regressors*betas from the raw TAC [6].

As **mandatory input**, this requires the numeric definition of the regressors to be plotted (`fpetbatch.tacplot.in.regr`, starting with 1=constant, 2=baseline, 3=first task, etc.) and a 3D image mask which defines the region of interest (`fpetbatch.tacplot.in.mask`). Plotting of TACs can be done for a single subject or as an average across multiple subjects, by providing the results file(s) of the GLM (`[fpetbatch.dir.result 'fPET_glm.mat']`) as a cell input. Thus, plotting TACs requires to run the GLM beforehand.

Optional inputs are flags to plot TACs for each individual (`fpetbatch.tacplot.in.indiv = 1`), the average of a group of subjects (`fpetbatch.tacplot.in.average=1`) and also to plot the raw TACs (`fpetbatch.tacplot.in.raw=1`). If only data of subject is to be plotted, then `fpetbatch.tacplot.in.indiv = 1` needs to be set.

The **output** of this function is a TAC for each regressor defined above, plotted for each individual or as average across all subjects.

GUI Example:

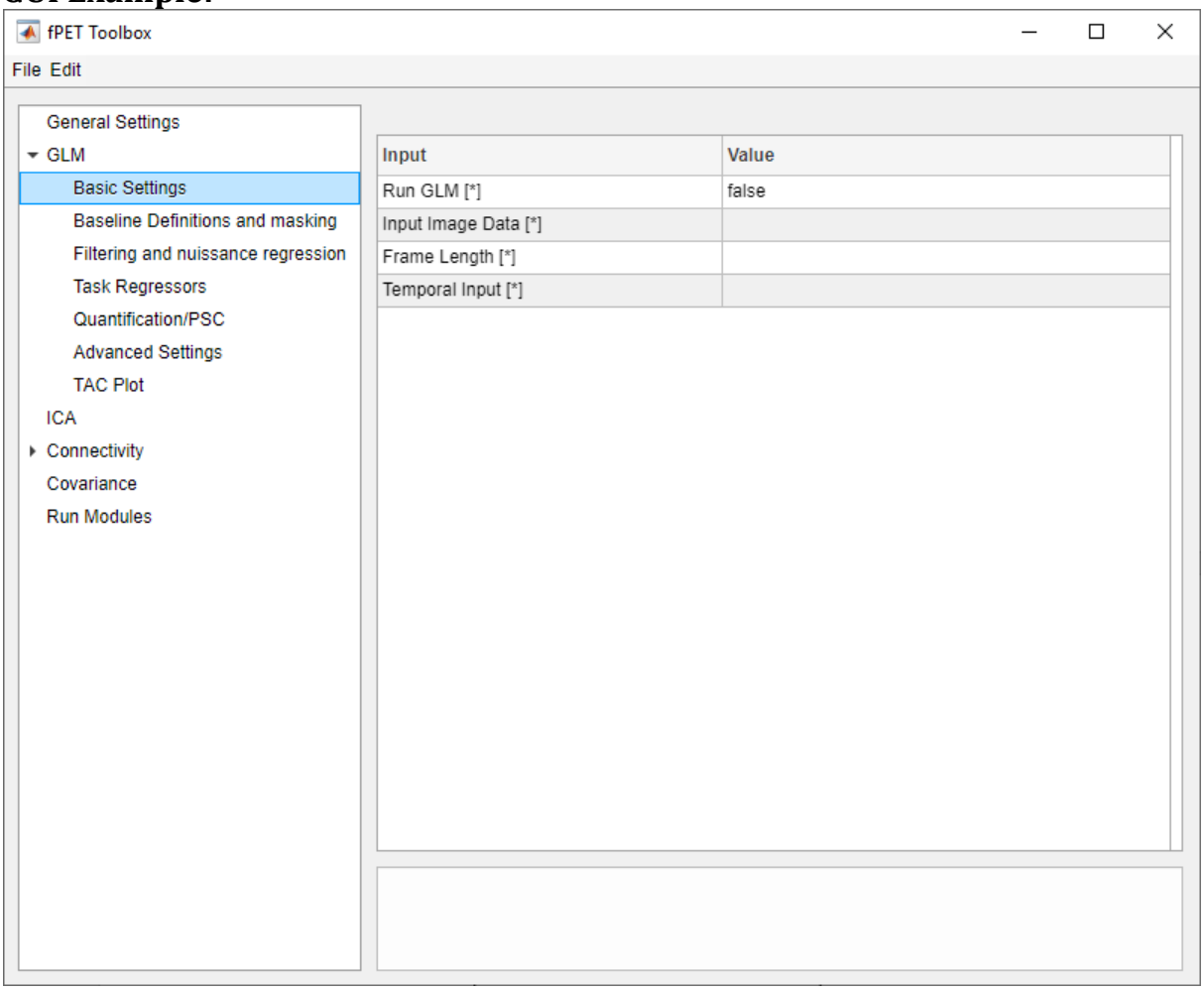

***Figure 7: GLM**. The GLM module is divided into 7 submenus, where only the Basic Settings are displayed above. The options for all variables can be examined by clicking on them. A single click will display a text in the bottom text area with a short description of the variable selected. Options with the [*] and listed in the help text as [Mandatory] are needed for the toolbox to run.*

Script Example:

```
%%
% glm general information
fpetbatch.run_glm = 1;
fpetbatch.glm.in.framelength = 30;            % duration of each frame [sec]
fpetbatch.glm.in.mask.bl = 'gray_matter.nii';     % baseline definition
fpetbatch.glm.in.mask.calc = 'gray_matter.nii';   % mask for calculations
fpetbatch.glm.in.time = 2;                % temporal input for glm in frames
fpetbatch.glm.in.mask.th = 5.02;            % thresholds for exclusion masks
fpetbatch.glm.in.fil.apply = 1;          % apply low pass filter
fpetbatch.glm.in.fil.cutoff = 12/2;      % half of task duration;
fpetbatch.glm.in.bl_type = 1;               % baseline definition: use mask
fpetbatch.glm.in.regr_motion_incomplete = 1;      % zero padding of motion

%%
% repeat the following for each subject
fpetbatch.dir.result = './results/subj001';              % results directory
fpetbatch.glm.in.data = 'fPET_subj001.nii';             % 4D input data
fpetbatch.glm.in.mask.bl_excl{1} = 'fMRI_subj001.nii';   % fMRI activation
% task regressors with 2 conditions (easy and hard) and 2 blocks each
fpetbatch.glm.in.regr(1).start = [condition1_start1 condition1_start2];
fpetbatch.glm.in.regr(1).end = [condition1_end1 condition1_end2];
fpetbatch.glm.in.regr(2).start = [condition2_start1 condition2_start2];
fpetbatch.glm.in.regr(2).end = [condition2_end1 condition2_end2];
fpetbatch.glm.in.regr(1).name = 'easy';
fpetbatch.glm.in.regr(2).name = 'hard';
% motion parameters
fpetbatch.glm.in.regr_motion = 'motion_subj001.txt';

% psc
fpetbatch.run_psc = 1;

% quantification
fpetbatch.run_quant = 1;
fpetbatch.quant.in.time = 2;           % temporal input for quant in minutes
fpetbatch.quant.in.bloodlvl = BloodGlucose_subj0001;
fpetbatch.quant.in.wb = 'WholeBlood_subj001.txt';
fpetbatch.quant.in.pwbr = 'PlasmaWholeBloodRatio_subj0001.txt';

fpet_tlbx(fpetbatch);



%%
% After each subject is completed

fpetbatch.run_tacplot = 1;
fpetbatch.tacplot.in.regr = 2:4;   % plot baseline (2), condition 1-2 (3-4)
% directory(ies) of input data
for ind_s = 1:nr_of_subj
    fpetbatch.tacplot.in.dir{ind_p} =
    sprintf('./results/subj%03i/fPET_glm.mat', ind_s);
end
fpetbatch.tacplot.in.mask = 'region_of_interest.nii';
fpetbatch.tacplot.in.indiv = 0;       % do not plot individual data
fpetbatch.tacplot.in.average = 1;     % plot average data
fpetbatch.tacplot.in.raw = 1;         % 1=plot raw tac

fpet_tlbx(fpetbatch);
```

10.6. Independent Component Analysis (ICA)

A data-driven alternative to identify stimulation effects is given by spatio-temporal ICA. The approach was implemented as described previously [4, 7, 8]. A few preprocessing steps are required before ICA. This includes removal of the baseline uptake by intensity normalization as well as z-scoring before and after data reduction by PCA separately for each individual. Afterwards, the timecourses of all subjects are concatenated, and then PCA and z-scoring are carried out again across individuals. After the ICA, the components are estimated these need to be inspected visually for relevance.

Mandatory input for ICA are 4D input data of one subject or a group of subjects (`fpetbatch.ica.in.data`). Furthermore, a mask needs to be provided to define which voxels are used for ICA (`fpet_defaults.ica.in.mask.calc`).

Optional inputs are the use of PCA (`fpetbatch.ica.in.pca, default=1`), the number of principal components (`fpetbatch.ica.in.pc, default = 40`) and the number of independent components (`fpetbatch.ica.in.ic, default = 20`) to be estimated.
It is possible to remove initial (`fpetbatch.ica.in.rem_start`) or final fPET frames (`fpetbatch.ica.in.rem_end`) from the acquired data. This is useful if radiotracer application started after or ended earlier with respect to data acquisition. This information is provided in seconds (`fpetbatch.ica.in.time=1`) or frames (`fpetbatch.ica.in.time=2`). For the former, the duration of a frame also needs to be defined in seconds (`fpetbatch.ica.in.framelength`).

The **output** of this function is a 3D map for each independent component.

GUI Example:

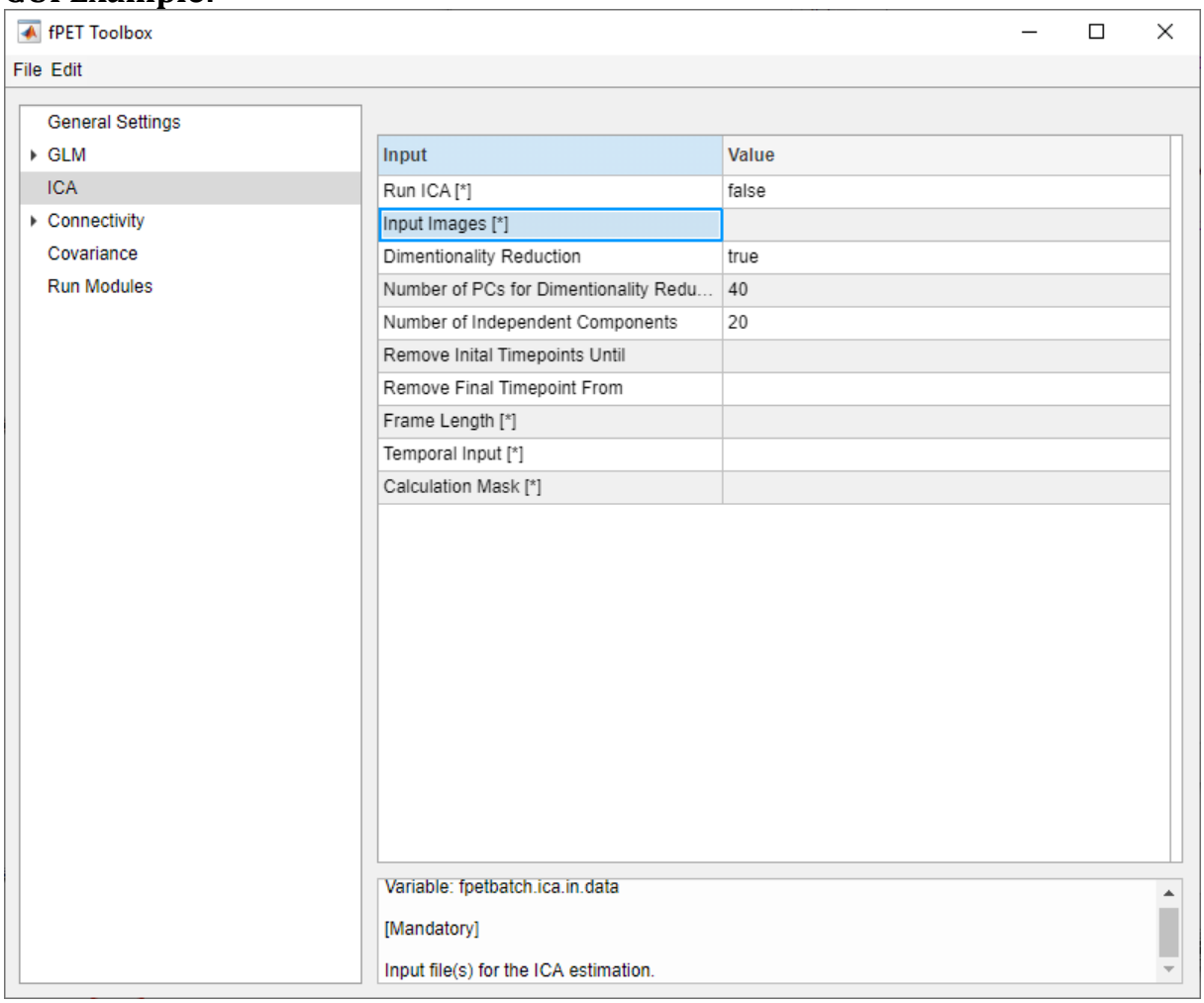

***Figure 8: ICA**. After starting the fPET toolbox and selecting the ICA module from the menu, the following screen should appear as above. The options for "ICA" can be examined by clicking on them. A single click will display a text in the bottom text area with a short description of the variable selected. Options with the [*] and listed in the help text as [Mandatory] are needed for the toolbox to run.*

Script Example:

```
% ica general information
fpetbatch.run_ica = 1;
fpetbatch.dir.result = './results/ica';

% input files
for ind_s = 1:nr_of_subj
    fpetbatch.ica.in.data{ind_s} = sprintf('./fPET_subj%03i', ind_s);
end

fpetbatch.ica.in.mask.calc = 'gray_matter.nii';   % mask for calculations
fpetbatch.ica.in.pc = 20;         % number of principal components
fpetbatch.ica.in.ic = 10;         % number of independent components

fpetbatch.ica.in.time = 2;          % temporal input for ica in frames
fpetbatch.ica.in.rem_start = 10;    % remove initial frames

fpet_tlbx(fpetbatch);
```

10.7. Molecular Connectivity

PET imaging has also been used to estimate interregional associations with various different radioligands [19, 21, 25]. We would like to note that the terminology is used in accordance with a recent consensus work in the field [18], which has also been interpreted in the same way by others [26]. Essentially, this emphasizes that the term 'connectivity' is used to compute linkage between regions across time within an individual. On the other hand, the term 'covariance' refers to the computation of associations across an entire group of subjects, i.e., between subjects.

Thus, molecular connectivity estimates moment-to-moment fluctuations in the fPET signal, which in turn requires to eliminate the baseline radiotracer uptake. This can be achieved with various options such as regression against a representative baseline TAC (defined as average of a mask or modeled by a third order polynomial) or by modeling the baseline with a third order polynomial [9].

Mandatory inputs are 4D input data (`fpetbatch.conn.in.data`), the duration of each frame in seconds (`fpetbatch.conn.in.framelength`) and a definition if timing variables are provided in seconds (`fpetbatch.conn.in.time=1`) or frames (`fpetbatch.conn.in.time=2`). Also, a 3D atlas with regions of interest needs to be defined (`fpetbatch.conn.in.atlas`).

Optional inputs are the definition to remove the baseline radiotracer uptake. One option is the regression against a representative baseline TAC. This baseline TAC can be defined by a mask (`fpetbatch.conn.in.bl_type=1`) [12], which can be further modeled by a third order polynomial (`fpetbatch.conn.in.bl_type=2`) [6]. For any of these options a 3D mask for the baseline definition needs to be provided (`fpetbatch.conn.in.mask_bl`). Another option is to fit each ROI's TAC separately with a third order polynomial (`fpetbatch.conn.in.bl_type=3`) [9]. Using any of the polynomial models above further requires to define the starting point of the polynomial (`fpetbatch.conn.in.bl_start_fit`) and all data before the fit is discarded from further calculations.

Motion parameters (`fpetbatch.conn.in.regr_motion`) and additional regressors (`fpetbatch.conn.in.regr_add`) can be added as text files. By default, motion regressors are subject to principal component analysis and the elbow criterion is used to identify the components that explain a sufficient amount of variance (`fpetbatch.conn.in.regr_motion_pca=1`) [14].

Advanced options

It is possible to remove initial (`fpetbatch.conn.in.rem_start`) or final fPET frames (`fpetbatch.conn.in.rem_end`) from the acquired data (as well as motion/additional regressors). This is useful if radiotracer application started after or ended earlier with respect to data acquisition. This information is provided in seconds or frames (see `fpetbatch.conn.in.time`).

Often, early fPET frames contain little information (particularly for constant infusion only) and no motion parameters may be available for the beginning of the scan. This can be taken into account by setting a flag (`fpetbatch.conn.in.regr_motion_incomplete=1`) to add initial zeros to the motion parameters. Similarly, zeros can be added at the end (`fpetbatch.conn.in.regr_motion_incomplete=2`).

The **output** of this function is a respective individual connectivity matrix, based on pairwise correlations between brain regions.

GUI Example:

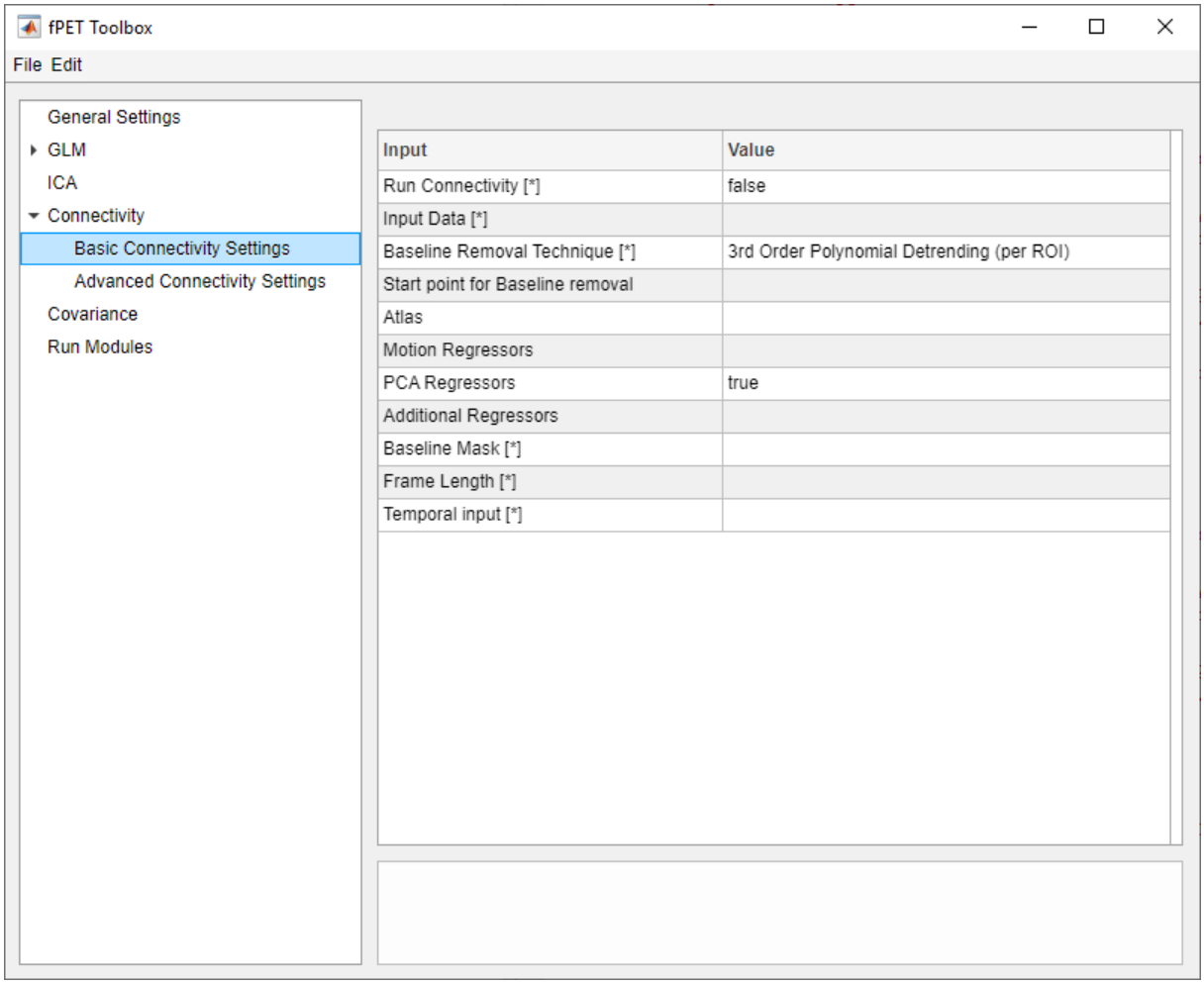

***Figure 9: Molecular Connectivity**. After selecting the Connectivity module from the menu, the following screen should appear. The options for "Connectivity" are separated into two submenus, the 1st being "Basic Connectivity Settings" where the necessary information must be filled out by the user. The 2nd submenu "Advanced Connectivity Settings" contains the advanced options (not displayed here) for a more flexible but complex analysis. A single click will display a text in the bottom text area with a short description of the variable selected. Options with the [*] and also listed in the help text as [Mandatory] are needed for the toolbox to run.*

Script Example:

```
%%
% molecular connectivity general settings
fpetbatch.run_conn = 1;

fpetbatch.conn.in.time = 2;             % temporal input for conn in frames
fpetbatch.conn.in.framelength = 60;     % duration of each frame [sec]
fpetbatch.conn.in.bl_type = 3;         % fit each ROI's TAC with polynomial
fpetbatch.conn.in.bl_start_fit = 1;      % start polynomial from 1st frame
fpetbatch.conn.in.atlas = 'atlas.nii';   % region of interest atlas

%%
% repeat the following for each subject
fpetbatch.dir.result = './results/subj001';     % result directory
fpetbatch.conn.in.data = 'fPET_subj001.nii';    % 4D input data
fpetbatch.conn.in.regr_motion = 'motion_subj001.txt';

fpet_tlbx(fpetbatch);
```

10.8. Molecular Covariance

As mentioned above, the computation of covariance matrices between subjects with PET has already been used 40 years ago [25, 27]. The approach uses static 3D maps (SUV, Ki, volume of distribution, binding potential, etc) and calculates pairwise regional correlations between subjects. As a consequence, it is not possible to draw inference for a single individual but only for the entire group. For more detailed discussions about this and related issued see other recent work [10, 18, 28, 29].

Mandatory inputs for this function are 3D images of several subjects (`fpetbatch.cov.in.data`) and an atlas that delineates the brain regions of interest (`fpetbatch.cov.in.atlas`).

As an **optional input** a mask can be provided to normalize the input data (`fpetbatch.cov.in.mask_norm`). This can for instance be a whole-brain or cerebellum mask, to convert e.g., SUV to SUVR values. Furthermore, it is possible to provide a vector or matrix in text format with covariates, e.g., sex, age, etc (`fpetbatch.cov.in.regr_add`).

The **output** is a correlation matrix calculated between all subjects with pairwise correlations of all brain regions.

GUI Example:

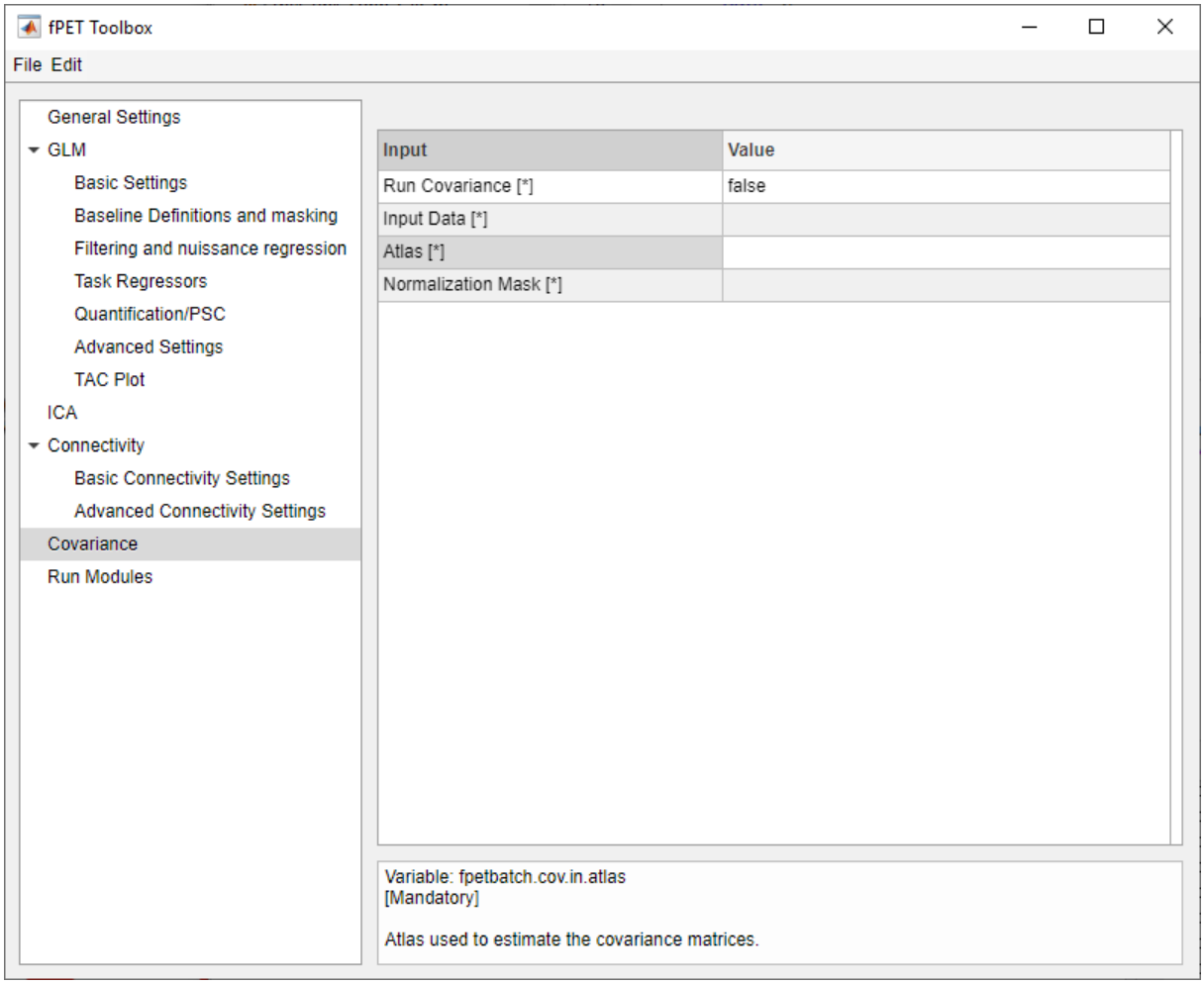

Figure 10: Molecular Covariance. *Once the „Covariance" option has been selected, the above shown screen should appear. A single click will display a text in the bottom text area with a short description of the variable selected. Options with the [*] and also listed in the help text as [Mandatory] are needed for the toolbox to run.*

Script Example:

```
%%
% molecular covariance general settings
fpetbatch.run_cov = 1;
fpetbatch.dir.result = './results/cov';     % result directory;

% input files
for ind_s = 1:nr_of_subj
    fpetbatch.cov.in.data{ind_s} = sprintf('./static_subj%03i', ind_s);
end

% region of interest atlas
fpetbatch.cov.in.atlas = 'atlas.nii';

% mask for intensity normalization
fpetbatch.cov.in.mask_norm = 'gray_matter.nii';

fpet_tlbx(fpetbatch);
```

11. ENVIRONMENT and DEPENDENCIES

The toolbox has been written in Matlab (versions R2021b update 3 and R2018a). The toolbox has a dependency on SPM12 (https://www.fil.ion.ucl.ac.uk/spm/) and uses a few functions such as for data loading and saving. For ICA the FastICA package is used (https://research.ics.aalto.fi/ica/fastica/).

12. LICENSE and DISCLAIMER

The fPET toolbox is open-source software released under the terms of the GNU General Public License version 2 as published by the Free Software Foundation (https://www.gnu.org/licenses/old-licenses/gpl-2.0.html). In short, it may be modified and distributed for non-commercial purposes, provided the original source is cited. The software is provided 'as is,' without any warranty, express or implied, including but not limited to warranties of fitness for a particular purpose. The developers shall not be held liable for any damages or issues arising from its use.

13. ACKNOWLEDGEMENTS

This research was funded in whole or in part by the Austrian Science Fund (FWF) [Grant DOI 10.55776/KLI610 and 10.55776/KLI1151, PI: A. Hahn, Grant DOI 10.55776/KLI1006, PI: R. Lanzenberger] and the Vienna Science and Technology Fund (WWTF) [10.47379/CS18039], Co-PI: R. Lanzenberger. For the purpose of open access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.
The toolbox logo was created with Dall-E-3 (https://openai.com/index/dall-e-3/).

14. DESCRIPTION of VARIABLES

Temporal inputs are highlighted because the flag `fpetbatch.*.in.time` will affect these inputs.

General variables

Variable	Description	Default value
`fpetbatch.dir.result`	Directory where results are saved.	Working directory
`fpetbatch.overwrite`	Flag, overwrite existing results: 1=yes, 0=no.	0

Mandatory variables for GLM

Variable	Description	Default value
`fpetbatch.run_glm`	Flag, 1 = run GLM.	
`fpetbatch.glm.in.data.`	Path/filename of fPET input data, 4D Nifti file. Assumes units = kBq/cm^3	
`fpetbatch.glm.in.framelength`	Duration of each frame [seconds], applies to all frames. Numeric input.	
`fpetbatch.glm.in.mask.bl`	Path/filename of baseline mask (usually gray matter), 3D Nifti file. An average TAC (e.g., across all gray matter voxels) is extracted with this mask to estimate baseline effects. See further options below.	
`fpetbatch.glm.in.time`	Flag, temporal inputs of GLM are given in 1=seconds or 2=frames, applies to all temporal inputs of the GLM.	

Optional variables for GLM

Variable	Description	Default value
Masks		
`fpetbatch.glm.in.mask.calc`	Path/filename of calculation mask (usually gray matter), 3D Nifti file. This mask specifies, where calculations are actually carried out, all other voxels will be zero.	Entire volume
`fpetbatch.glm.in.mask.bl_excl{ind_m}`	Path/filename for baseline exclusion mask(s), 3D Nifti file(s), accepts multiple inputs. Non-zero voxels in these mask(s) will be removed from the baseline mask (`fpetbatch.glm.in.mask.bl`). This may include task activations from individual fMRI data or a meta-analysis, anatomical regions, etc.	
`fpetbatch.glm.in.mask.th[ind_m]`	Thresholds for baseline exclusion masks (`fpetbatch.glm.in.mask.bl_excl{ind_m}`), numeric input. One threshold may be provided for each mask as a real number, order of masks and thresholds must match. When multiple masks are used, then thresholds are required for all masks or none of them. If threshold = 0, then all voxels < 0 are set to 0 in that mask. If threshold ~= 0, then –threshold < voxel < threshold are set to 0 in that mask (can be used e.g., t threshold individual fMRI results).	NaN

	If threshold = NaN, mask remains unchanged.	
Filter		
`fpetbatch.glm.in.fil.apply`	Flag, Do (=1) or do not (=0) apply low pass FIR filter.	1
`fpetbatch.glm.in.fil.order`	Order of FIR filter, numeric input. Specify as half of the desired filter order, since `filtfilt` is used.	6 (i.e., 12 order FIR filter).
`fpetbatch.glm.in.fil.cutoff`	Cutoff frequency of FIR filter. Numeric input, temporal input. Specify in seconds or frames (see `fpetbatch.glm.in.time`), inverse value is calculated internally.	300 sec
Stimulation regressors		
Stimulation regressors are defined in frames or seconds (see `fpetbatch.glm.in.time`). These are constructed as ramp functions with slope = 1 kBq/min. If no stimulation regressor is defined, it is assumed that stimulation is defined as additional regressor or data is resting state. Multiple regressors are accepted, defined by ind_r. For each regressor, multiple starting and end values are accepted which define the different task blocks of the same condition (defined as vector by ind_b).		
`fpetbatch.glm.in.regr(ind_r).start[ind_block]`	Start value(s) of stimulation regressor ind_r. Numeric input, temporal input.	
`fpetbatch.glm.in.regr(ind_r).end[ind_block]`	End value(s) of stimulation regressor ind_r. Numeric input, temporal input.	
`fpetbatch.glm.in.regr(ind_r).name`	Name of regressor ind_r. Text input. Optional.	stim
Baseline model		
`fpetbatch.glm.in.bl_type`	Flag, representative baseline TAC defined by 1=mask only (see `fpetbatch.glm.in.mask.bl` and `fpetbatch.glm.in.mask.bl_excl`) or 2=3rd order polynomial fitting.	1
`fpetbatch.glm.in.bl_start_fit`	If `fpetbatch.glm.in.bl_type`=2, time point to start fit of polynomial given in seconds or frames (see `fpetbatch.glm.in.time`). For continuous infusion only, first time point fits. For bolus+infusion data, chose a time point after the initial bolus uptake in the TAC has passed. Numeric input, temporal input.	
Motion and additional regressors		
`fpetbatch.glm.in.regr_motion`	Path/filename of textfile that includes motion parameters, e.g., realignment from SPM. Assumes that the longer dimension is time.	
`fpetbatch.glm.in.regr_motion_pca`	Flag. Do (=1) or do not (=0) perform PCA on motion parameters.	1
`fpetbatch.glm.in.regr_add`	Path/filename of textfile that includes additional regressors. Assumes that longer dimension is time.	
Advanced options for GLM		
`fpetbatch.glm.in.regr_motion_incomplete`	Flag. Add zeros at the beginning (=1) or end (=2) of motion parameters to match fPET input data.	0
`fpetbatch.glm.in.regr_orth`	Flag. Orthogonalize stimulation regressors to baseline.	1
`fpetbatch.glm.in.rem_start`	Number of initial data points to be removed from the fPET data (and motion/additional regressors). Numeric input, temporal input. Given in seconds or frames (see `fpetbatch.glm.in.time`).	
`fpetbatch.glm.in.rem_end`	Number of final data points to be removed from the fPET data (and motion/additional	

	regressors). Numeric input, temporal input. Given in seconds or frames (see `fpetbatch.glm.in.time`).	
`fpetbatch.glm.in.data_incomplete`	Flag. Indicates that data acquisition was incomplete, partly corrupted, etc.	0
`fpetbatch.glm.in.data_incomplete.start`	Start values of missing data. Numeric input, temporal input. Given in seconds or frames (see `fpetbatch.glm.in.time`).	
`fpetbatch.glm.in.data_incomplete.end`	End values of missing data. Numeric input, temporal input. Given in seconds or frames (see `fpetbatch.glm.in.time`).	

Mandatory variables for PSC

Calculation of PSC requires that a stimulation regressor was defined in the GLM (`fpetbatch.glm.in.regr`).

Variable	Description	Default value
`fpetbatch.run_psc`	Flag, 1 = run PSC.	

Optional variables for PSC

Variable	Description	Default value
`fpetbatch.dir.result`	Directory where GLM results are saved.	Working directory.

Mandatory variables for Quantification

Absolute quantification requires that a stimulation regressor was defined in the GLM (`fpetbatch.glm.in.regr`).
For blood data, either a plasma input function or a whole blood function + plasma/whole blood ratio is provided as text files with two columns: 1=time (seconds or minutes), 2=value (same unit as PET data). If the first row is text input (i.e., a header line), this is ignored for calculations.

Variable	Description	Default value
`fpetbatch.run_quant`	Flag, 1 = run quantification.	
`fpetbatch.quant.in.time`	Flag, temporal inputs for the quantification are given in 1=seconds or 2=minutes, applies to all temporal inputs of the quantification.	
`fpetbatch.quant.in.wb`	Path/filename of textfile with whole blood data. Temporal input.	
`fpetbatch.quant.in.plasma`	Path/filename of textfile with plasma data. Temporal input.	
`fpetbatch.quant.in.pwbr`	Path/filename of textfile with plasma/whole blood ratio data. Temporal input.	

Optional variables for Quantification

Variable	Description	Default value
`fpetbatch.dir.result`	Directory where GLM results are saved.	Working directory
`fpetbatch.quant.in.pwbr_fit`	Flag to define fit of plasma/whole blood ratio. 1=average, 2=linear fit.	1
`fpetbatch.quant.in.lc`	Lumped constant, numeric input.	0.89
`fpetbatch.quant.in.vb`	Fractional whole-blood volume, numeric input.	0.05
`fpetbatch.quant.in.bloodlvl`	Prescan blood glucose level, numeric input given as [mmol/L].	
`fpetbatch.quant.in.parent`	Path/filename of textfile with parent fraction data. Temporal input.	
`fpetbatch.quant.in.tstar`	Time to start fit of Patlak plot, given as fraction of full scan duration, numeric input.	1/3 of scan duration

Mandatory variables for potting TACs

Plotting of TACs requires that a stimulation regressor was defined in the GLM (`fpetbatch.glm.in.regr`).

Variable	Description	Default value
`fpetbatch.run_tacplot`	Flag, 1 = run quantification.	
`fpetbatch.tacplot.in.regr`	Number of regressor(s) to be plotted, starting with 1=constant, 2=baseline, etc. Numeric input.	
`fpetbatch.tacplot.in.dir{ind_f}`	Path/filename(s) of fPET input data (fPET_glm.mat), cell array.	
`fpetbatch.tacplot.in.mask`	Path/filename of mask to extract TAC, 3D Nifti file.	

Optional variables for potting TACs

Variable	Description	Default value
`fpetbatch.tacplot.in.indiv`	Flag to plot individual data.	0
`fpetbatch.tacplot.in.average`	Flag to plot group average data.	1
`fpetbatch.tacplot.in.raw`	Flag to plot raw TAC.	0

Mandatory variables for ICA

Variable	Description	Default value
fpetbatch.run_ica	Flag, 1 = run ICA.	
fpetbatch.ica.in.data {ind_f}	Path/filename of fPET input data, 4D Nifti file. Cell array.	
fpetbatch.ica.in.mask.calc	Path/filename of calculation mask (usually gray matter). i.e., which voxels to include in calculations. 3D Nifti file.	

Optional variables for ICA

Variable	Description	Default value
fpetbatch.ica.in.pca	Flag. Do (=1) or do not (=0) perform PCA for dimensionality reduction.	1
fpetbatch.ica.in.pc	Number of principal components to be estimated. Numeric input.	40
fpetbatch.ica.in.ic	Number of independent components to be estimated. Numeric input.	20
fpetbatch.ica.in.time	Flag, temporal inputs for the ICA are given in 1=seconds or 2=frames, applies to all temporal inputs of the ICA.	
fpetbatch.ica.in.rem_start	Number of initial data points to be removed from the fPET data. Numeric input, temporal input. Given in seconds or frames (see fpetbatch.ica.in.time).	
fpetbatch.ica.in.rem_end	Number of final data points to be removed from the fPET data. Numeric input, temporal input. Given in seconds or frames (see fpetbatch.ica.in.time).	
fpetbatch.ica.in. framelength	Duration of each frame [seconds], applies to all frames. Numeric input.	

Mandatory variables for Molecular Connectivity

Variable	Description	Default value
fpetbatch.run_conn	Flag, 1 = run molecular connectivity.	
fpetbatch.conn.in.data	Path/filename of fPET input data, 4D Nifti file.	
fpetbatch.conn.in.framelength	Duration of each frame [seconds], applies to all frames.	
fpetbatch.conn.in.time	Flag, temporal inputs of connectivity are given in 1=seconds or 2=frames, applies to all temporal inputs of the connectivity calculations.	
fpetbatch.conn.in.atlas	Path of atlas with regions of interest, 3D Nifti file.	

Optional variables for Molecular Connectivity

Variable	Description	Default value
Baseline removal and filter		
fpetbatch.conn.in.bl_type	Flag to define removal of baseline radiotracer uptake. 1=mask to define representative TAC, 2=third order polynomial fitted to representative TAC, 3=third order polynomial fitted to each ROI's TAC separately.	3
fpetbatch.conn.in.mask_bl	If bl_type = 1 or 2: path of the mask for baseline definition (usually gray matter). An average TAC (e.g., across all gray matter voxels) is extracted with this mask to remove baseline effects. 3D Nifti file.	
fpetbatch.conn.in.bl_start_fit	If bl_type = 2 or 3: time point to start fit of third order polynomial. Specified in seconds or frames (see fpetbatch.conn.in.time). Temporal input.	
Motion and additional regressors		
fpetbatch.conn.in.regr_motion	Path/filename of textfile that includes realignment parameters, e.g., from SPM. Assumes that the longer dimension is time.	
fpetbatch.conn.in.regr_motion_pca	Flag. Do (=1) or do not (=0) perform PCA on motion parameters.	1
fpetbatch.conn.in.regr_add	Path/filename of textfile that includes additional regressors. Assumes that longer dimension is time.	
Advanced options		
fpetbatch.conn.in.regr_motion_incomplete	Flag. Add zeros at the beginning (=1) or end (=2) of motion parameters to match fPET input data.	
fpetbatch.conn.in.rem_start	Number of initial data points to be removed from the fPET data (and motion/additional regressors). Numeric input. Temporal input, given in seconds or frames (see fpetbatch.conn.in.time).	
fpetbatch.conn.in.rem_end	Number of final data points to be removed from the fPET data (and motion/additional regressors). Numeric input. Temporal input, given in seconds or frames (see fpetbatch.conn.in.time).	

Mandatory variables for Molecular Covariance

Variable	Description	Default value
fpetbatch.run_cov	Flag, 1 = run molecular covariance.	
fpetbatch.cov.in.data {ind_f}	Path/filenames of PET input data, cell input of several 3D Nifti files.	
fpetbatch.cov.in.atlas	Path/filename of atlas with regions of interest, 3D Nifti file.	
fpetbatch.cov.in.mask_norm	Path/filename of mask to normalize (i.e., scale) data, usually gray matter.	

Optional variables for Molecular Covariance

Variable	Description	Default value
fpetbatch.cov.in.regr_add	Path/filename of textfile that includes additional regressors (e.g., sex, age, etc.). Assumes that longer dimension is subjects.	

15. REFERENCES

1. Hahn A, Breakspear M, Rischka L, Wadsak W, Godbersen GM, Pichler V, et al. Reconfiguration of functional brain networks and metabolic cost converge during task performance. Elife. 2020;9:e52443. doi:10.7554/eLife.52443.
2. Stiernman LJ, Grill F, Hahn A, Rischka L, Lanzenberger R, Panes Lundmark V, et al. Dissociations between glucose metabolism and blood oxygenation in the human default mode network revealed by simultaneous PET-fMRI. Proc Natl Acad Sci U S A. 2021;118. doi:10.1073/pnas.2021913118.
3. Godbersen GM, Klug S, Wadsak W, Pichler V, Raitanen J, Rieckmann A, et al. Task-evoked metabolic demands of the posteromedial default mode network are shaped by dorsal attention and frontoparietal control networks. eLife. 2023;12:e84683.
4. Haas S, Bravo F, Ionescu TM, Gonzalez-Menendez I, Quintanilla-Martinez L, Dunkel G, et al. Active Suppression of the Nigrostriatal Pathway during Optogenetic Stimulation Revealed by Simultaneous fPET/fMRI. bioRxiv. 2023:2023.10.19.556049. doi:10.1101/2023.10.19.556049.
5. Villien M, Wey HY, Mandeville JB, Catana C, Polimeni JR, Sander CY, et al. Dynamic functional imaging of brain glucose utilization using fPET-FDG. Neuroimage. 2014;100:192-9.
6. Hahn A, Gryglewski G, Nics L, Hienert M, Rischka L, Vraka C, et al. Quantification of Task-Specific Glucose Metabolism with Constant Infusion of 18F-FDG. J Nucl Med. 2016;57:1933-40. doi:jnumed.116.176156.
7. Jamadar SD, Ward PG, Li S, Sforazzini F, Baran J, Chen Z, et al. Simultaneous task-based BOLD-fMRI and [18-F] FDG functional PET for measurement of neuronal metabolism in the human visual cortex. Neuroimage. 2019;189:258-66. doi:10.1016/j.neuroimage.2019.01.003.
8. Li S, Jamadar SD, Ward PGD, Premaratne M, Egan GF, Chen Z. Analysis of continuous infusion functional PET (fPET) in the human brain. Neuroimage. 2020;213:116720. doi:10.1016/j.neuroimage.2020.116720.
9. Reed MB, Ponce de Leon M, Vraka C, Rausch I, Godbersen GM, Popper V, et al. Whole-body metabolic connectivity framework with functional PET. Neuroimage. 2023;271:120030. doi:10.1016/j.neuroimage.2023.120030.
10. Jamadar SD, Ward PGD, Liang EX, Orchard ER, Chen Z, Egan GF. Metabolic and Hemodynamic Resting-State Connectivity of the Human Brain: A High-Temporal Resolution Simultaneous BOLD-fMRI and FDG-fPET Multimodality Study. Cereb Cortex. 2021;31:2855-67. doi:10.1093/cercor/bhaa393.
11. Hahn A, Gryglewski G, Nics L, Rischka L, Ganger S, Sigurdardottir H, et al. Task-relevant brain networks identified with simultaneous PET/MR imaging of metabolism and connectivity. Brain Structure & Function. 2018;223:1369-78. doi:10.1007/s00429-017-1558-0.
12. Rischka L, Gryglewski G, Pfaff S, Vanicek T, Hienert M, Klobl M, et al. Reduced task durations in functional PET imaging with [(18)F]FDG approaching that of functional MRI. Neuroimage. 2018;181:323-30. doi:10.1016/j.neuroimage.2018.06.079.
13. Klug S, Godbersen GM, Rischka L, Wadsak W, Pichler V, Klobl M, et al. Learning induces coordinated neuronal plasticity of metabolic demands and functional brain networks. Communications biology. 2022;5:428. doi:10.1038/s42003-022-03362-4.
14. Hahn A, Reed MB, Vraka C, Godbersen GM, Klug S, Komorowski A, et al. High-temporal resolution functional PET/MRI reveals coupling between human metabolic and hemodynamic brain response. Eur J Nucl Med Mol Imaging. 2023. doi:10.1007/s00259-023-06542-4.
15. Hahn A, Reed MB, Pichler V, Michenthaler P, Rischka L, Godbersen MG, et al. Functional dynamics of dopamine synthesis during monetary reward and punishment processing. J Cereb Blood Flow Metab. 2021;41:2973-85.
16. Hahn A, Reed MB, Murgas M, Vraka C, Klug S, Schmidt C, et al. Dynamics of human serotonin synthesis differentially link to reward anticipation and feedback. Mol Psychiatry. 2024. doi:10.1038/s41380-024-02696-1.
17. Reed MB, Handschuh PA, Schmidt C, Murgas M, Gomola D, Milz C, et al. Validation of cardiac image derived input functions for functional PET quantification. https://doiorg/101101/2023092923296343. 2023.

18. Reed M, Cocchi L, Knudsen G, Sander C, Gryglewski G, Chen J, et al. Connecting the Dots: Approaching a Standardized Nomenclature for Molecular Connectivity Combining Data and Literature. bioRxiv. 2024:2024.05.10.593490. doi:10.1101/2024.05.10.593490.
19. Hahn A, Lanzenberger R, Wadsak W, Spindelegger C, Moser U, Mien LK, et al. Escitalopram enhances the association of serotonin-1A autoreceptors to heteroreceptors in anxiety disorders. J Neurosci. 2010;30:14482-9.
20. Hahn A, Haeusler D, Kraus C, Hoflich AS, Kranz GS, Baldinger P, et al. Attenuated serotonin transporter association between dorsal raphe and ventral striatum in major depression. Hum Brain Mapp. 2014;35:3857-66.
21. Vanicek T, Hahn A, Traub-Weidinger T, Hilger E, Spies M, Wadsak W, et al. Insights into Intrinsic Brain Networks based on Graph Theory and PET in right- compared to left-sided Temporal Lobe Epilepsy. Sci Rep. 2016;6:28513.
22. Rischka L, Godbersen GM, Pichler V, Michenthaler P, Klug S, Klobl M, et al. Reliability of task-specific neuronal activation assessed with functional PET, ASL and BOLD imaging. J Cereb Blood Flow Metab. 2021;41:2986-99. doi:10.1177/0271678X211020589.
23. Godbersen GM, Falb P, Klug S, Silberbauer LR, Reed MB, Nics L, et al. Non-invasive assessment of stimulation-specific changes in cerebral glucose metabolism with functional PET. Eur J Nucl Med Mol Imaging. 2024;51:2283-92. doi:10.1007/s00259-024-06675-0.
24. Patlak CS, Blasberg RG. Graphical evaluation of blood-to-brain transfer constants from multiple-time uptake data. Generalizations. J Cereb Blood Flow Metab. 1985;5:584-90. doi:10.1038/jcbfm.1985.87.
25. Horwitz B, Duara R, Rapoport SI. Intercorrelations of glucose metabolic rates between brain regions: application to healthy males in a state of reduced sensory input. J Cereb Blood Flow Metab. 1984;4:484-99. doi:10.1038/jcbfm.1984.73.
26. Deery HA, Liang E, Siddiqui MN, Murray G, Voigt K, Di Paolo R, et al. Metabolic connectivity in ageing. bioRxiv. 2024:2024.06.16.599247. doi:10.1101/2024.06.16.599247.
27. McIntosh AR, Grady CL, Ungerleider LG, Haxby JV, Rapoport SI, Horwitz B. Network analysis of cortical visual pathways mapped with PET. J Neurosci. 1994;14:655-66.
28. Sala A, Lizarraga A, Ripp I, Cumming P, Yakushev I. Static versus Functional PET: Making Sense of Metabolic Connectivity. Cerebral Cortex. 2021;32:1125-9. doi:10.1093/cercor/bhab271.
29. Jamadar SD, Egan GF. Resting-State FDG-PET Connectivity: Covariance, Ergodicity, and Biomarkers. Response to Commentary by Sala et al.; Static versus Functional PET: Making Sense of Metabolic Connectivity. Cereb Cortex. 2022;32:2054-5. doi:10.1093/cercor/bhab316.