PRF_estimation

This repository provides scripts for our work presented in:

Kassinopoulos, M., Mitsis, G.D., 2019. Identification of Physiological Response Functions to Correct for Fluctuations in Resting-State fMRI related to Heart Rate and Respiration. bioRxiv 512855. https://doi.org/10.1101/512855

In this work, we present a novel framework for estimating physiological response functions (PRFs) in resting-state fMRI and we demonstrate the feasibility of this framework on data from the Human Connectome Project (HCP). Specifically, we use physiological recordings acquired during the scan as well as the global signal (i.e. average timeseries across all voxels in the brain) to extract the cardiac and respiration response functions. These PRFs are subsequently used in the analysis to correct for BOLD fluctuations induced by changes in heart rate and respiration. In addition, in this work we present **probabilistic maps of brain areas affected by cardiac pulsatility, breathing motion, heart rate and breathing pattern** obtained from 41 healthy young subjects of the HCP. These maps can be downloaded from:

- Neurovault: https://neurovault.org/collections/5654/, or
- SharePoint: https://mcgill-my.sharepoint.com/personal/michalis_kassinopoulos_mail_mcgill_ca/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fmichalis%5Fkassinopoulos%5Fmail%5Fmcgill%5Fca%2FDocuments%2FMy%20project%2FMy%20Papers%2FPaper%201%20%2D%20Effect%20of%20HR%20and%20RVT%20on%20fMRI%20data%2FGithub%20%2D%20PRF%20estimation

Extraction of PRF curves and associated physiological regressors

In our work, we present 5 main PRF models that can be used to extract physiological regressors related to fluctuations in HR and breathing pattern. In each of these models, what is needed for the PRF estimation is the heart rate (HR), the respiratory flow (RF; or respiration volume per time in the case of the standard model) and the global signal (GS) of each scan in the dataset. The scripts in this repository were written for the resting-state fMRI data from the HCP. However, they can be adjusted to any dataset as far as physiological recordings (cardiac and respiratory signal) are recorded. If you have troubles adjusting it to your dataset please free to send me a sample file of your physiological data and I can adjust the script to your dataset.

Specifically, for a particular scan (e.g. scan Rest2_RL from subject 118730) we are using the following files from the HCP database (https://db.humanconnectome.org/app/template/Login.vm):

- Physio_log.txt
  - Path: HCP_1200/118730/MNINonLinear/Results/rfMRI_REST2_RL/rfMRI_REST1_LR_Physio_log.txt
  - It contains the cardiac and respiratory signals as well as the time indices that the fMRI volumes were acquired, at a sampling rate of 400 Hz.
- Movement_Regressors_dt.txt:
  - Path: HCP_1200/118730/MNINonLinear/Results/rfMRI_REST2_RL/Movement_Regressors_dt.txt
  - It contains the six motion realignment parameters as well as their derivatives. All twelve timeseries are linearly detrended.
  - If you are working on a different dataset, the six (3 translational and 3 rotational) motion (volume) realignment parameters are estimated during the volume realignment stage. In FSL fMRI software package (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki), volume realignment is done through MCFLIRT within FEAT. The last 6 parameters (i.e. the time derivatives of the first 6 timeseries) can be estimated in numerical computing environments (e.g. Matlab and Python).
- func.nii.gz:

    - Path: HCP_1200/118730/MNINonLinear/Results/rfMRI_REST2_RL/rfMRI_REST2_RL.nii.gz
    - This is the 4-d fMRI dataset in niftii format. In HCP, this fMRI dataset has been corrected among others for volume misalignment and has been registered to the MNI space (see Methodology – Section 2.1 for more details). If you are working on a different dataset, you can do a similar preprocessing using fMRI software packages such as FSL.
- Brainmask_fs.2.nii.gz:
    - Path: HCP_1200/118730/MNINonLinear/Results/rfMRI_REST2_RL/brainmask_fs.2.nii.gz
    - This is a binary mask in the MNI space with ones at the voxels within the whole brain (WB). You can extract this mask in your dataset using BET from FSL.
- T1w_restore.nii.gz and T1w_restore_brain.nii.gz:
    - Path for first file: HCP_1200/118730/MNINonLinear/T1w_restore.nii.gz &
    - Path for second file: HCP_1200/118730/MNINonLinear/T1w_restore_brain.nii.gz
    - Typically, in an fMRI dataset there is a structural T1 image for each subject. The T1w_restore.nii.gz found in HCP database corresponds to this T1 image after correction for gradient-nonlinearity-induced correction. The T1w_restore_brain.nii.gz differs from the T1w_restore.nii.gz in that voxels outside the brain are set to zero. This file can be obtained using the tool BET in FSL if it's not available in your dataset. These two files are used for tissue segmentation in order to derive the masks for the gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF).

Steps for PRF estimation

1. The script *Preprocess_Phys.m* gets as input the text file *Physio_log.txt* and after preprocessing of the physiological recordings extracts several variables such as the HR timeseries and saves them in the MAT-File *Physio_and_triggers.mat*. *Preprocess_Phys.m* consists of 6 sections. The first section loads the *Physio_log.txt* and should be adjusted to the directory structure your files are stored on your computer. The user should examine the physiological data for each scan one by one as some parameters in the script should be adjusted to account for the quality of the recordings or the differences in the heart rate across scans.

2. The *Extract_GS.m* gets as input the fMRI data, the masks of the whole brain (WB), gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF), the motion parameters as well as the physiological variables extracted with *Preprocess_Phys.m*. The output of this script is the MAT-File *GS.mat* which includes 8 variants of GS. The four variants correspond to the four mean timeseries from the WB, GM, WM and CSF compartment whereas the remaining four variants correspond to the aforementioned timeseries after correcting for motion and high-frequency cardiac and respiratory artifacts (RETROICOR). Note that the scripts for extracting the RETROICOR regressors can be found in the folder RETROICOR along with the script *example_code.m* which shows an example of applying RETROICOR on two voxel timeseries to model the aliased cardiac pulsatility artifacts and the breathing-related motion artifacts. Furthermore,

3. After extracting the GS and the physiological variables one of the following scripts is used to estimate the PRF curves (unless the standard method is used which has predefined curves) and extract the physiological regressors needed to be included later in the analysis (e.g. in the general linear model as confounds):
    a. *M1_PRF_stand.m*
    b. *M2_PRF_pop.m*
    c. *M3_PRF_pop_sc.m*
    d. *M4_PRF_sc.m*
    e. *M5_PRF_sc_vxl.m*

    For details regarding the properties of these models please refer to our article (Kassinopoulos & Mitsis, 2019). Some of the main differences between the models are shown in **Fig. 1**. The physiological regressors extracted from the above scripts can be found in the Matlab variable *xPhys*. If you want to use these variables in a different software than Matlab you will probably have to save them in a text file with the format that the software can recognize.

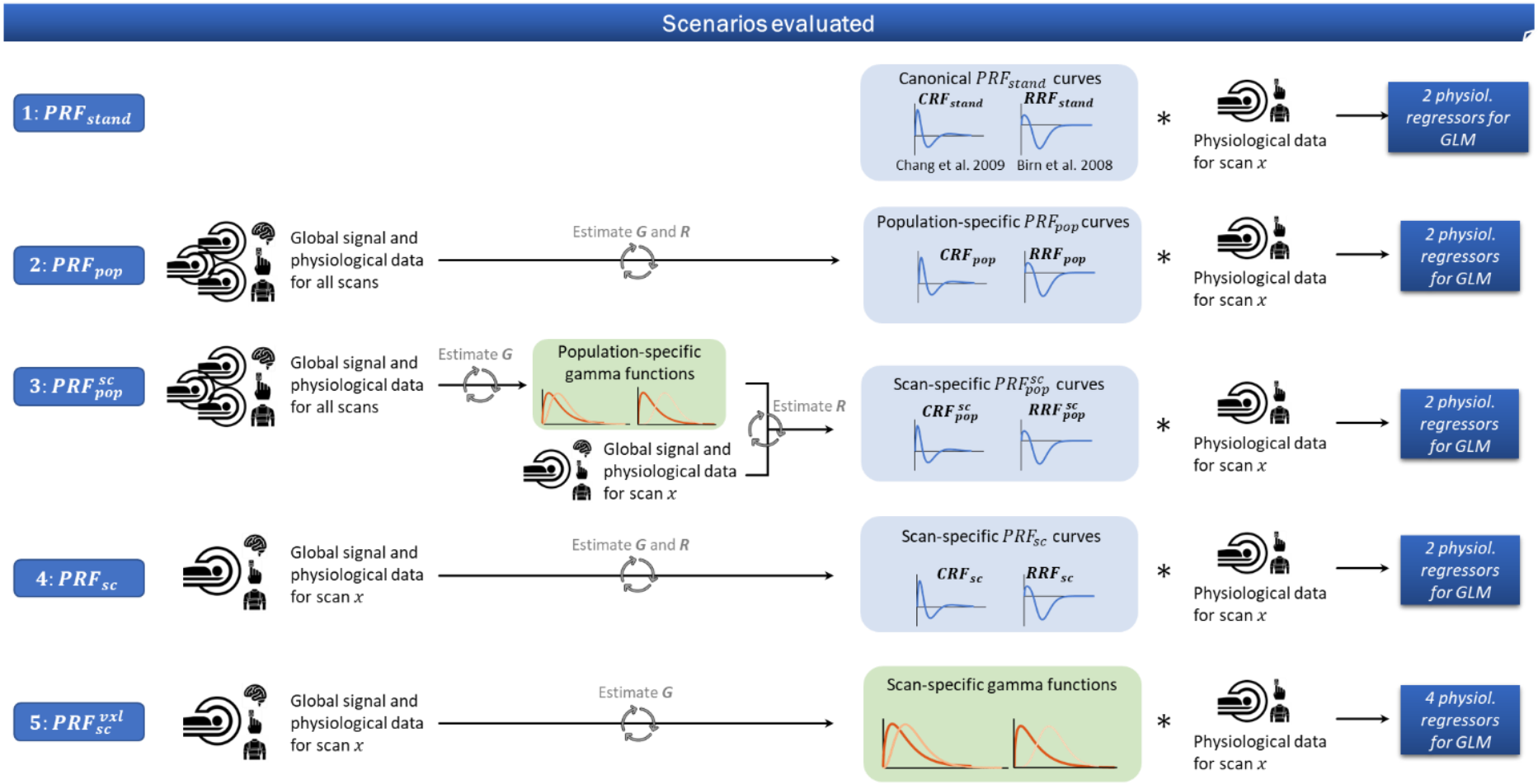

Fig. 1. Illustration of the required steps for deriving the physiological regressors for the voxel-wise GLM analysis in the five main examined models. The cross-validation framework is omitted for simplicity. Also, the models of the form $PRF_{sbj,xxx}$ and $PRF^{sc}_{sbj,xxx}$ from groups II and III in Table 1 respectively, are omitted from this diagram as they are variations of the models PRF_{pop} and PRF^{sc}_{pop} from the groups II and II, respectively (Table 1). Specifically, they only differ in the fact that the data considered in the first step for the estimation of the curves and gamma functions are obtained from the same subject rather than the same population.

How to use the Matlab scripts

All scripts provided in the repository consist of smaller sections. These sections can be easily identified as they start with the two percent signs (%%) followed by a title. As seen in **Fig. 2**, based on these titles using the Matlab editor toolbar you can easily navigate through the sections. The following list provides 3 useful ways to run the scripts:

1. Run the entire code by pressing **Run** on the Editor tab (or by pressing the key F5 in Windows; **Fig. 3**)
2. Run the code in the current section by placing the cursor in the code section and, then, pressing **Run Section** on the Editor tab (or pressing the keys Ctrl+Enter).
3. Run the code in specific lines by selecting the lines and pressing the key F9.

Running only specific sections or lines is useful particularly when you use the script *Preprocess_Phys.m* as you may have to optimize some parameters (e.g. the minimum distance between peaks) manually.

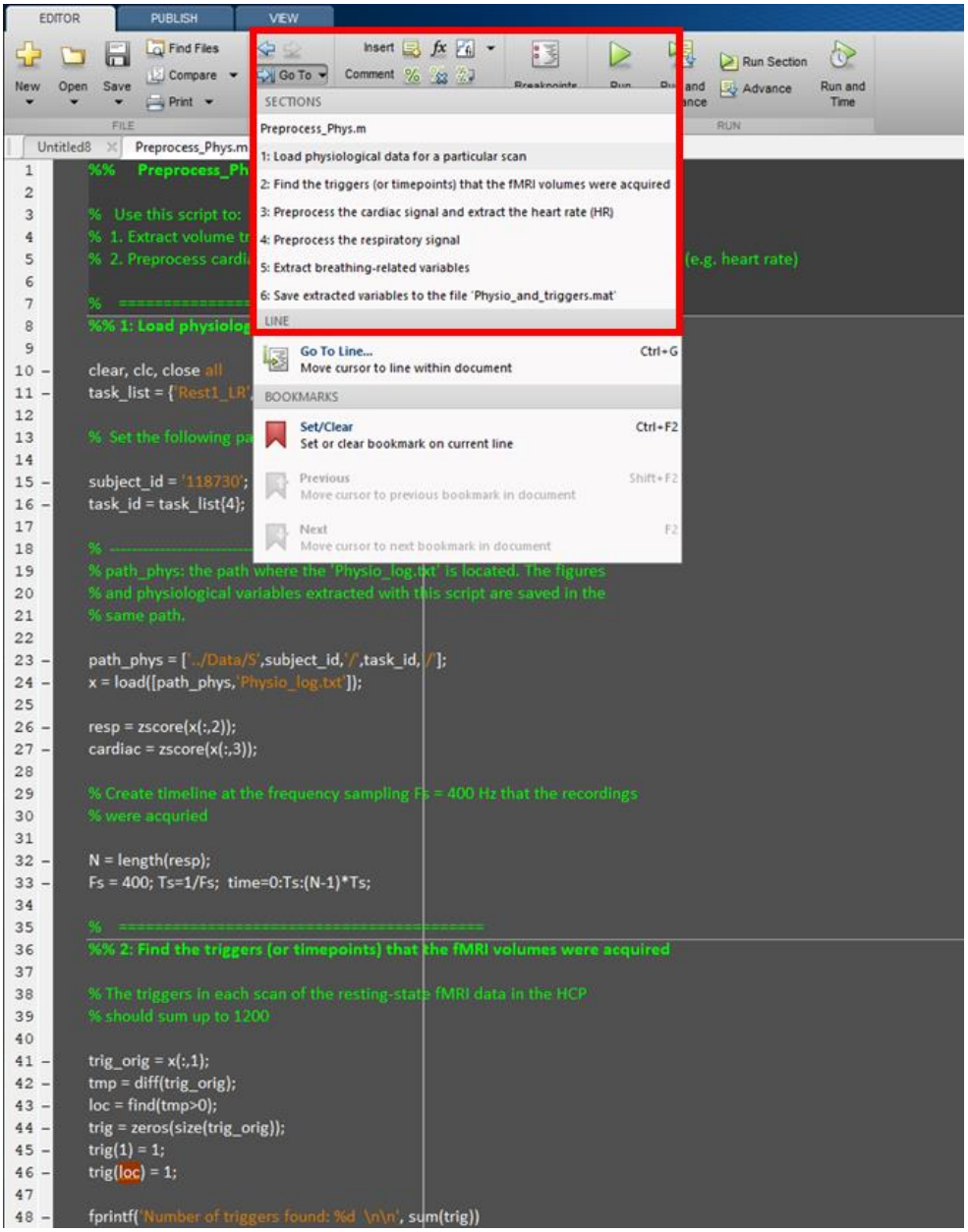

Fig. 2. Screenshot of Matlab Editor showing how the user can easily navigate through the different sections of a script.

Recommended arrangement for working in Matlab Environment

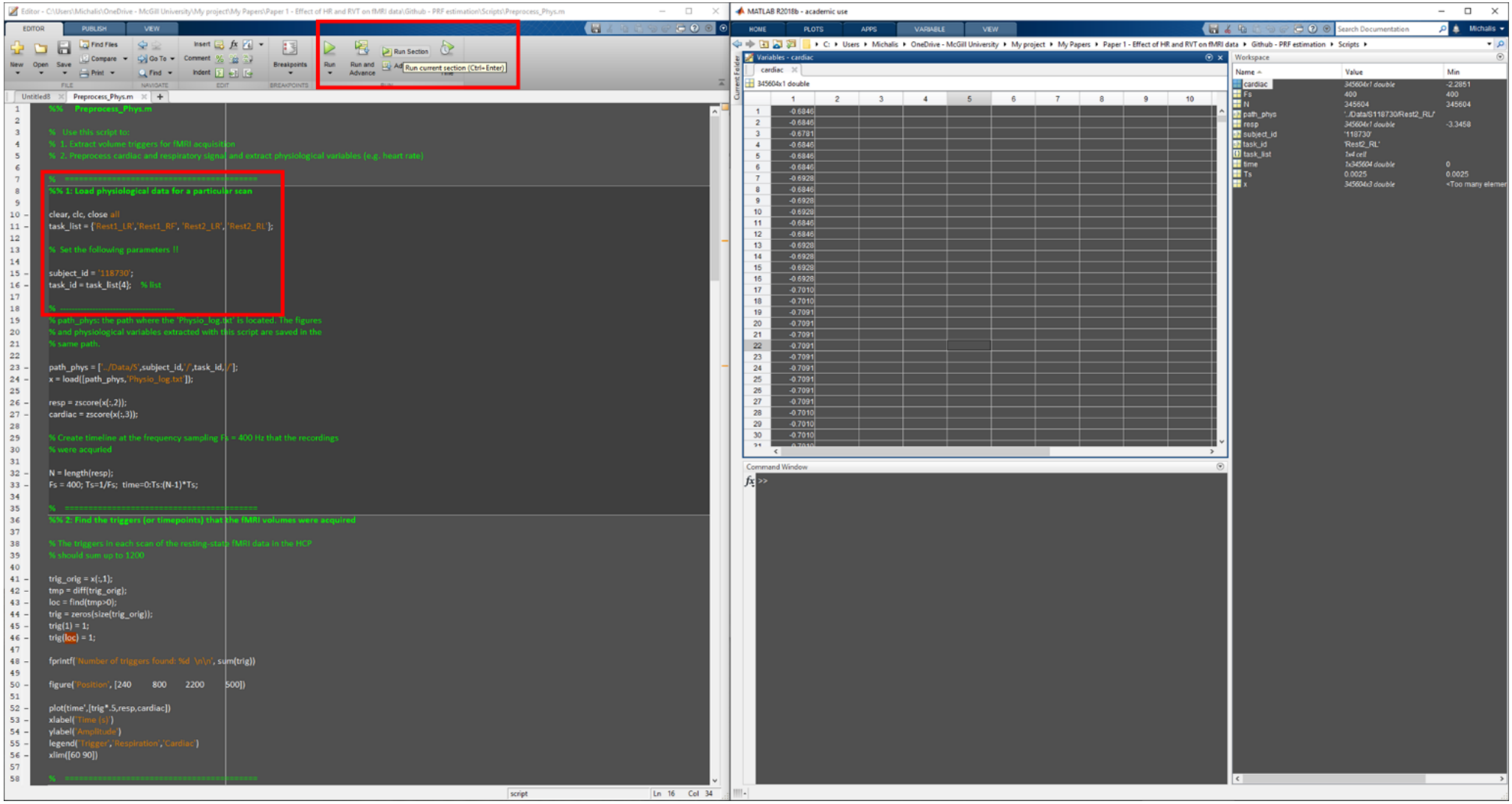

Fig. 3. Recommended arrangement for using the scripts. To run a specific section in the script, select the section by placing the mouse cursor in any of the lines that correspond to the section and, subsequently, press the button **Run Section** on the Editor tab (or press Ctrl+Enter in Windows). To run the entire script press **Run** on the Editor tab (or key F5 in Windows). To run specific lines, select the lines and press F9. All variables are shown in the Workspace area and their content can be viewed by double-clicking on them. Several parameters need to be set for each scan separately and their corresponing lines are found in the beginning of each section (e.g. Variables *subject_id* and *task_id* in Section 1 of *Preprocess_Phys.m*; Fig. 3).

Adjusting the size of the figures

In many parts of the scripts the figures are created with specific dimensions. This is done by using a command like:

```
figure('Position', [240       800       2200       500])
```

Depending on the size and resolution of your monitor these figures may look too small or too large. One way to correct for this is to do the following: First you plot the figure and adjust it to a size that satisfies you, subsequently, you enter *get(gcf,'position')* in the Matlab terminal and, finally, you replace the values in the command shown above with the values returned in the terminal. That way, whenever you run this part of the script, the figure will have a good size.

Author

Michalis Kassinopoulos, PhD candidate
Graduate Program in Biological and Biomedical Engineering, McGill University, Montreal, Canada
E-mail: mkassinopoulos@gmail.com

Date: 31-July-2019

Please do not hesitate to contact me if you have any questions related to the use of these scripts.

Michalis