RaBIDS

Rapid analysis pipeline using the Brain Imaging Data Structure

Tutorial with example data

For RaBIDS v0.2.1

Christian Paret – 2020

RG Psychobiologie of Selfregulation

Department of Psychosomatic and Psychotherapeutic Medicine

Central Institute of Mental Health

christian.paret@zi-mannheim.de

Tel: -4462

1   Table of content

2   Data availability

The current RaBIDS version, the manual, tutorial and complimentary data is available via

https://github.com/christianparet/RaBIDS

To receive the mytrainingdata practice-dataset, write an email request to

christian.paret@zi-mannheim.de

3   Introduction to this practice session

In this practice session, you will import a training fMRI-dataset into a BIDS-compatible structure. This introduction explains how the raw neuroimaging data (i.e., dicom files and logfiles) need to be provided. It is shown in the subsequent sections how to define study and subject information to use the fully automated import routines.

RaBIDS version 0.2 is developed for data received from Siemens PrismaFit machines. It currently works with logfiles received from Presentation software (neurobehavioral systems, version 18-20).

Matlab (compatibility tested with version 2020a), SPM12 and dicm2nii need to be installed on your computer. Consult the manual for detailed information.

3.1   MRI images

3.1.1   Naming of MRI series

An MRI measurement protocol can consist of several MRI series. For example, you may administer two functional tasks in one session, and the technical assistant (MTA) will start an echoplanar imaging (EPI) sequence for each task-run by selecting the corresponding tab at the MRI computer. It is recommended to assign unique names to each of these MRI series by changing the name of this tab. Ask the MTA or your local support team to do this when you set up the sequence. You should also have information ready on the number of images needed per sequence. The name and number of images will be used for automatic identification of each series for each task.

In the provided training dataset, two tasks were administered: a task called "scenes" and a task called "faces". The names of the EPI-sequences were "ep2d_TR2000_64_Scenes" and "ep2d_TR2000_64_Faces", respectively. Number of images for the scenes task is between 426 and 430, and number of images in the faces task is between 366 and 370. In addition, an anatomical MPRAGE scan was acquired (name: "t1_mpr_ns_sag_pat2_iso_asy", 192 images).

For more information on the training task see section 6.

3.1.2   Exporting MRI series

A training dataset of one participant was recorded (see section 6 for task information). The MTA who operated the MR machine exported the images to the data exchange server (at Central Institute also known as "hobbes") according to the structure below. Siemens machines export MR images typically into the dicom-mosaic data format with the file ending ".IMA" (sometimes ".dcm").

Generic structure for saving MRI images from the scanner site, suggested for your future projects (names in upper case are placeholders and need to be replaced according to your individual study. Ignore explanatory text in brackets). Each line corresponds to a directory. Indents indicate hierarchical structure: indented lines relate to directories that are nested within the superordinate directory:

- DATA EXCHANGE SERVER (i.e., hobbes)
    - STUDYNAME
        - dicomdir
            - sub-SUBJECTNUMBER
                - ses-SESSIONNUMBER
                    - *here are the MRI images*

For this tutorial you will be provided with this architecture:

- data exchange server
    - RABIDS-example
        - dicomdir
            - sub-RABIDS01
                - ses-01
                    - *RABIDS01.MR.PSM_BI-STUDIE…..IMA*
                    - *…*

To anticipate: If you adhere to the suggestions above in your own future projects, it will not be necessary to copy your MR images from the data exchange server to your local project directory anymore. Instead, you will define the path beforehand to point to your study directory on the server. RaBIDS will recognize and import your data automatically from there. This saves space on the hard drive, it saves your valuable time of tedious copy-pasting of MRI data and it protects for possible errors.

3.2   Presentation logfiles

Presentation (provided by the company neurobehavioral systems) is used for stimulus presentation during functional MRI. Future versions of RaBIDS may handle logfiles from other software, but Presentation is used in nearly all of our studies at the Central Institute and therefore was used to develop this beta RaBIDS version (v0.X).

Presentation logfiles contain task information that is required to receive stimulus functions from your expermiment, and the latter are necessary for modeling your fMRI data. The stimulus function lists the events of your experiment (e.g. onset of IAPS picture, duration of picture presentation, onset of fixation cross, etc.) and links them to the time when the fMRI volumes were acquired.

The data structure for saving logfiles from the scanner site is similar to the one used above, with the difference that data is saved to a directory called "sourcedata" instead of "dicomdir".

In this tutorial you can navigate through the data structure to the logfiles as shown below:

- data exchange directory
  - RaBIDS-example
    - sourcedata
      - sub-RABIDS01
        - ses-01
          - *sub-RABIDS01_ses-01_task-faces.log*
          - *sub-RABIDS01_ses-01_task-scenes.log*

Again, I recommend to adopt this structure for your own project. RaBIDS can then automatically find the logfiles corresponding to subjects, sessions and tasks. You don't need to copy the sourcedata directory from the exchange server to your local project directory.

3.3   Get training data

To get started with this example, download and unzip the directory called "mytrainingdata.zip" to your local directory.

The structure of the unzipped training dataset should look like this:

- mytrainingdata
    - data exchange server *(see 3.1.2 for information)*
        - RABIDS-example
            - sourcedata (*contains the logfiles with stimulus event information*)
            - dicomdir (*contains the imaging data*)
    - your project directory
        - dataset *(this directory is empty)*

When you access the mytrainingdata directory on Windows it should look like this:
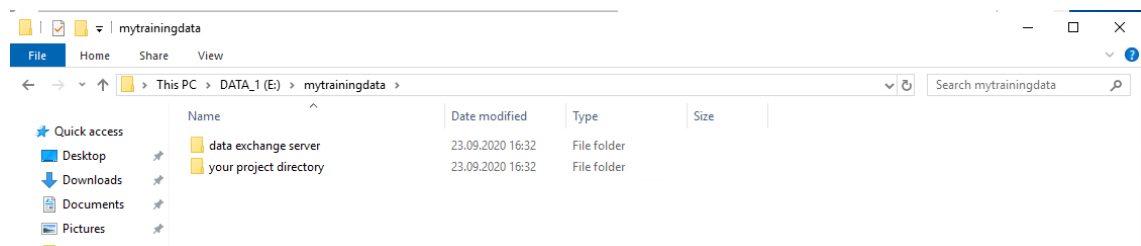
Download the code-directory to the dataset directory (\mytrainingdata\your project directory\dataset\code):
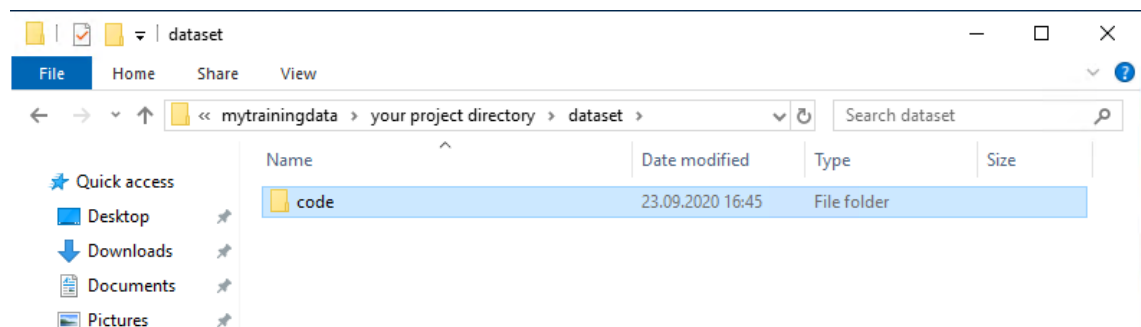
4 Import data

Go to the templates directory (\\your project directory\dataset\code\RaBIDS templates), select the file "datasheet.xlsx" and copy it to the code directory (your project directory\dataset\code\).

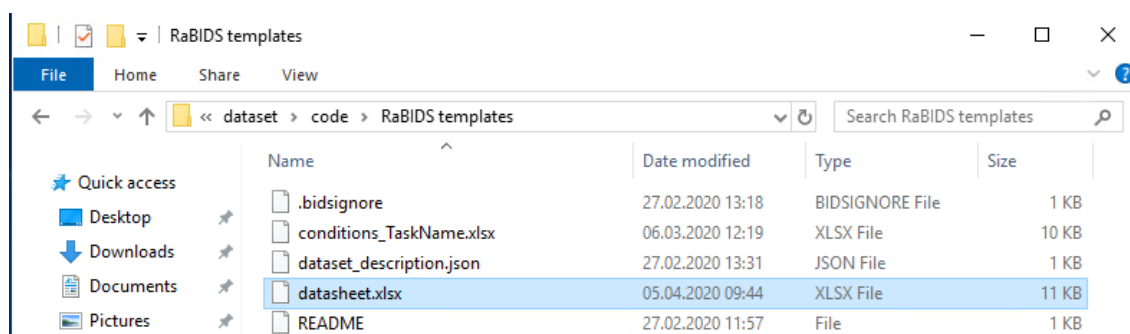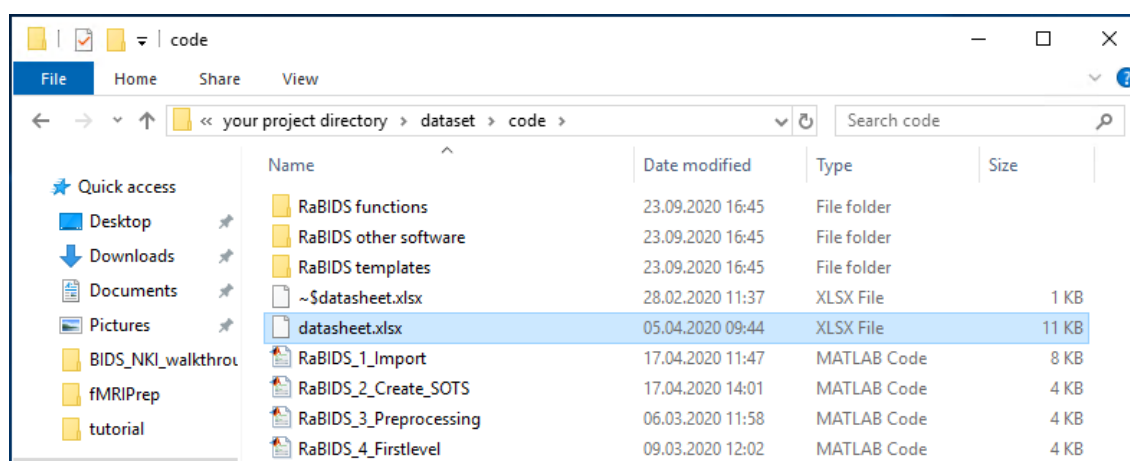Image showing where to find the datasheet.xlsx:

Image showing where to copy-paste it:

To use the automated import program we first need to populate the "datasheet.xlsx" with information about the training data. **Table 1** (see Appendix) displays the required user input to import the data with the program "RaBIDS_1_Import.m". Fill in the user input as shown in **Table 1**. Alternatively, you can skip this step and copy-paste the table datasheet.xlsx from the directory "tutorial supportive data". This table is ready to run for this tutorial. Consult the manual for more information on the different ObjectTypes.

Important when filling in the datasheet: Go to UserInput field of ObjectType "data analysis path" (field C6) and write here the path name to your project directory. Go to UserInput field in ObjectType "data exchange path" (field C7) and write path name to the directory containing the MR images (i.e., the directory \data exchange server\RABIDS-example\dicomdir). To to UserInput field of ObjectType "sourcedata path" (field C8) and write down the path name to the sourcedata directory (i.e., the directory \data exchange server\RABIDS-example\sourcedata).

When you are finished, you need to save the excel file (press Ctrl+S on your keyboard). Now you sould be able to run the program RaBIDS_1_Import.m from Matlab:

Click on RaBIDS_1_Import with the right mouse button:
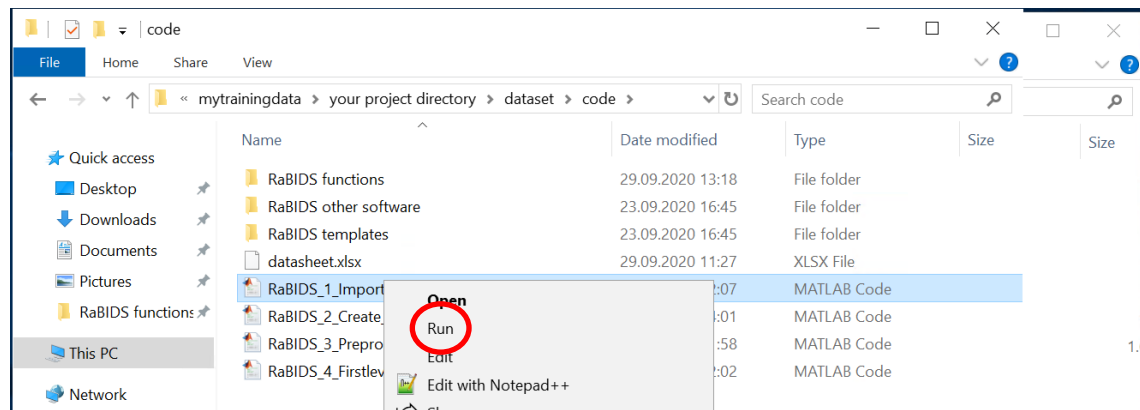
Alternatively, open Matlab and press Run:
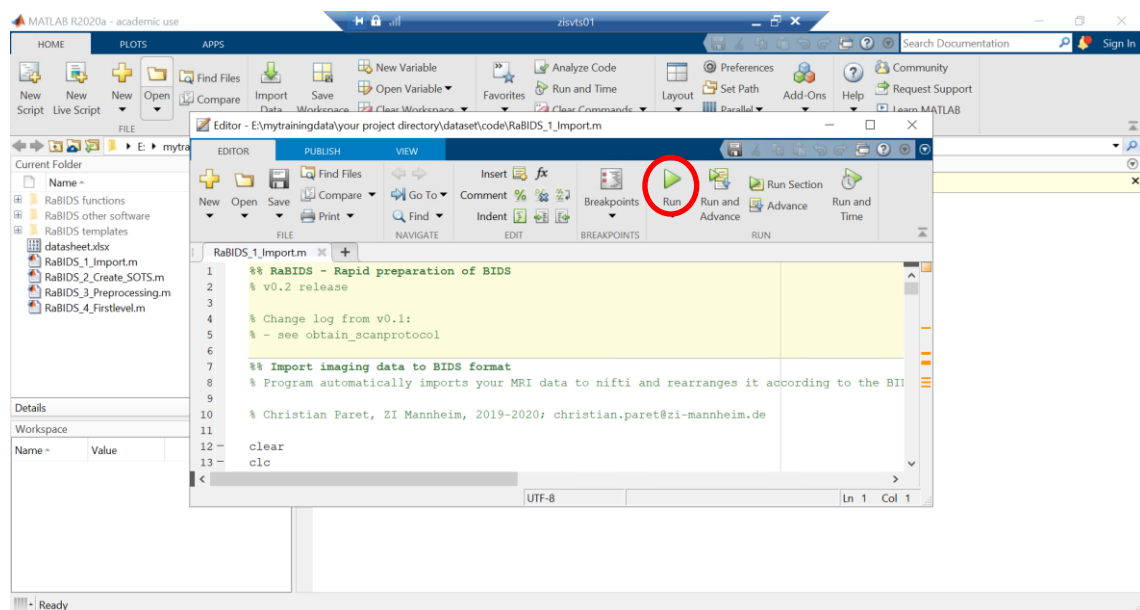
Matlab will write a warning to the command window that variable names were modified (Error #1)[1]. We can ignore this message.

Check the command window output after import. For each of the three Tasks (yes, RaBIDS knows the anatomical image as a "task"!) the program should return that dicom import was successful.

4.1   Scan protocol

Go to the subject directory of session 1 (\mytrainingdata\your project directory\dataset\sub-RABIDS01\ses-01) and open the text file called "sub-

___
[1] You find a document namned "RaBIDS Error Reference" in the manuals directory. This doc serves as reference for known errors and explains solutions.

RABIDS01_ses-01_scanprotocol.txt" with Excel. It lists the series number, name and number of scans of each MRI series. This protocol is generated automatically when you run import step and is used to assign the scans to the tasks.
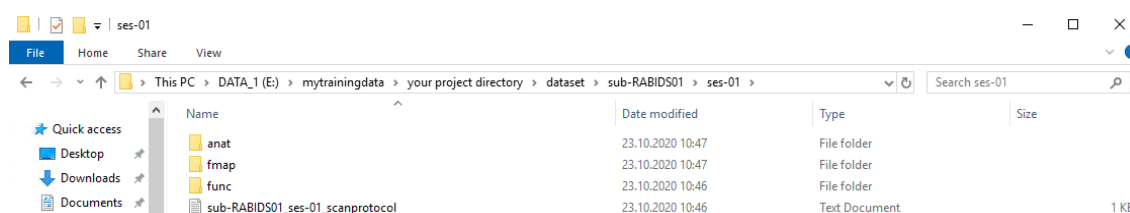
4.2   RaBIDS logs

RaBIDS writes the Matlab command window output to a logfile. This logfile is saved in the logfile directory (\mytrainingdata\ RaBIDS-logs) and is called "sub-RABIDS01_ses-01_RaBIDS-log.txt". The file can be helpful to locate errors.

4.3   Define task conditions and SOT

In this section, you define the conditions and stimulus onset times (SOT). The program will use this information to define the stimulation protocol of the subject, based on the information from the Presentation logfile. Output is saved to your project directory\dataset\sub-RaBIDS01\ses-01\func. Two files are generated per task:

1. sub-RaBIDS_ses-01_func_task-TaskName_multicond.mat: SOTS-file   for analysis with SPM

2. sub-RaBIDS01_ses-01_func_task-TaskName_events.tsv: SOTS-file in BIDS-format.

Go to the templates directory (your project directory\dataset\code\RaBIDS templates), select the file "conditions_TaskName.xlsx" and copy it to the code directory (your project directory\dataset\code\). Rename the file to "conditions_scenes.xlsx" and open it.[2]

---

[2] **Excursus:** To understand what information is needed in the table, go to the directory containing the presentation logfiles (\your project directory\sourcedata\sub-RaBIDS01\ses-01). You can use Excel to open the file "sub-RaBIDS01_ses-01_task-faces.log".
Colomn C "Code" shows the relevant event information. For us, the relevant entries are:
-    "onset_negative_ XXX": onset of an 18-sec picture trial of condition "negative"

Enter the information as shown in Table 2 to the conditions_scenes.xlsx file.

In this example we use the Duration feature. You can use the OffsetID feature to define trial duration, too (refer to manual for more information)[3]. Alternatively, you can skip the step above and download the ready-to-use conditions files from the directory "tutorial supportive data".

Now you the conditions_scenes.xlsx file should appear in your code directory as shown below:
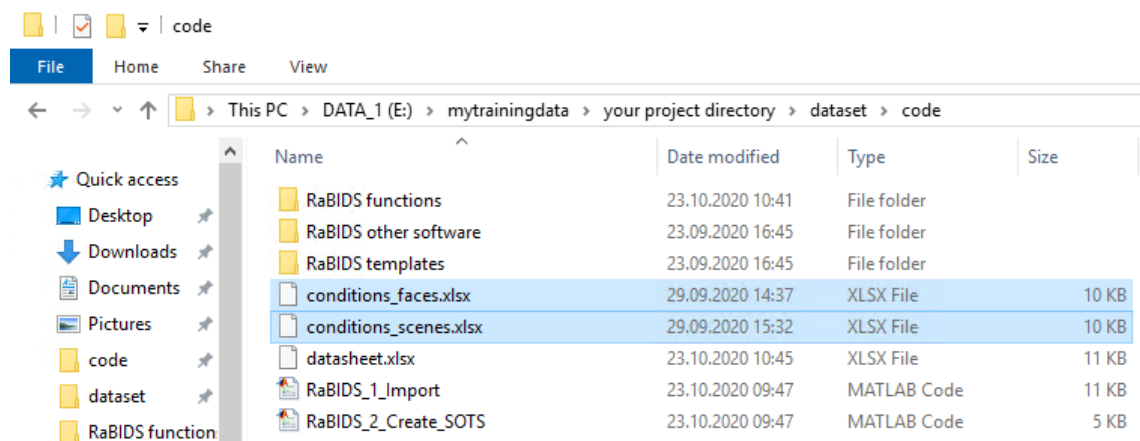
Run RaBIDS_2_Create_SOTS.m in Matlab. Matlab requests to input the TaskName via the Command Window. Use inverted commas around TaskName (e.g. for TaskName=scenes, type 'scenes').

You need to repeat these steps for the faces task (see Table 3).

- "onset_scrambled_XXX": onset of an 18-sec scrambled-picture trial of condition "control"
- "rest": onset of fixation cross

The events are listed with their onset time. When running the program RaBIDS_2_Create_SOTs, this information is looked up automatically for each trial and is written to the ouput files.

[3] In order to add a new Condition to the conditions_scenes.xlsx file, you need to add a new line to the table. To do this, you can mark the third line by clicking on the row number '3' on the very left of the excel-table and press Ctrl and + on the keyboard.

5   BIDS validation

For fun, go to website <u>https://bids-standard.github.io/bids-validator/</u> and validate your dataset. The ouput shows that the dataset was recognized in good order:

Congratulations, you produced your first BIDS-compatible data set!

6   Data description

6.1   Tasks

6.1.1   Faces

Participants were shown faces with emotional expression from the Warsaw Set of Emotional Facial Expression Pictures (WSEFEP, http://www.emotional-face.org/). A block design of 12 blocks with 6 negative emotional faces and 12 blocks with scrambled faces (control condition) was used. In sum, 72 negative faces of 24 actors (12 female, 12 male) were shown for 3 seconds each. The inter-trial interval was jittered between 10±1 seconds. To ensure attention, participants were asked to press a button to indicate for every picture whether the person was male or female, or whether color of the bounding box around scrambled faces was blue or green.

6.1.2   Scenes

During this task, pictures from the OASIS picture set (Kurdi et al., 2017) are presented to induce negative affect. We used pictures with negative affective valence and high arousal (aversive condition, AC) in a block-design. During each of 14 blocks, lasting 18 seconds, three picture stimuli were presented consecutively for 6 seconds each, resulting in a set of 42 negative pictures in total. Scrambled pictures were used in a non-affective control condition (neutral condition, NC) with same number of trials. During the intertrial interval (10±1 seconds), participants viewed a white fixation cross on a black background. To ensure attention, participants were asked to press a button to indicate for every picture whether a person can be seen or not, or whether color of the bounding box around scrambled pictures is blue or green.

6.1.3   MRI data acquisition

FMRI data were acquired on a 3 Tesla MRI Scanner (Magnetom Trio with TIM technology, Siemens Medical Service, Erlangen, Germany) equipped with a 32 channel head coil. Functional images of the BOLD contrast were acquired with gradient echo T2*

weighted echo-planar-imaging sequence (TE=30 ms, TR=2 s, FOV=220 mm x 220 mm, Matrix size=64 x 64, flip angle=80°). A volume comprised 36 slices in AC-PC orientation with a thickness of 3 mm and slice gap of 1 mm. Participants' heads were lightly restrained using soft pads to prevent head movement. A T1-weighted anatomical image was also recorded (TE=3.03 ms, TR=2.3 s, 192 slices and FOV=256 mm x 256 mm, Matrix size 256 x 256, Slice thickness = 1 mm).

Scans were acquired in this order: localizer – anatomical – fieldmap – task-faces – task-scenes

7   Appendix

Table 1. datasheet.xlsx

Note: paths to spm12, dicm2nii may be different on your computer. Data analysis path, data exchange path and sourcedata path need to be costumized, too. Consult the manual for more information on ObjectTypes. Note: Matlab 2020a requires fields MinImages and MaxImages to be filled for the first add-path object, this is the reason why MinImages and MaxImages are set to "0".

ObjectType	Description	UserInput	MinImages	MaxImages
add path	SPM	C:\Program Files\spm12	0	0
add path	dicm2nii	C:\Program Files\dicm2nii	0	0
add path	saveJSONfile	.\RaBIDS other software\saveJSONfile	0	0
add path	functions	.\RaBIDS functions	0	0
data analysis path	e.g. your project directory	E:\mytrainingdata\your project directory	0	0
data exchange path	path to your project's dicom directory on hobbes	E:\mytrainingdata\data exchange server\RABIDS-example\dicomdir	0	0
sourcedata path	path to logfiles	E:\mytrainingdata\data exchange server\RABIDS-example\sourcedata		
dicoms	how to expect dicom files?	BIDS		
subject info	subject ID	RABIDS01		
add prefix	add 'sub-' to subject ID	yes		
MRI series	TaskName and define min-max	scenes	426	430
series info	MRI series ID	ep2d_TR2000_64_Scenes		
MRI series	TaskName and define min-max	faces	366	370
series info	MRI series ID	ep2d_TR2000_64_Faces		
MRI series	TaskName and define min-max	anat	192	
series info	MRI series ID	t1_mpr_ns_sag_pat2_iso_asy		
MRI series	TaskName and define min-max	fieldmap magnitude	72	
series info	MRI series ID	gre_field_mapping		
MRI series	TaskName and define min-max	fieldmap phasediff [scenes] [faces]	36	
series info	MRI series ID	gre_field_mapping	0,00492	0,00738
general suffix	general suffix to find MR images	PSM_BI-STUDIE		
session ID	first session ID	ses-01		
session info	first session suffix			
first image	first image (delete initial n-1 images)	yes	5	
write scan protocol	saves scan protocol to dicom directory	yes		
overwrite import	overwrite existing session directory	yes		
overwrite sots	overwrite existing sots files (...multicond.mat)	yes		

Table 2. conditions_scenes.xlsx

	Name	OnsetID	Duration	OffsetID	ContrastType	ContrastPlus1	ContrastMinus1
Condition1	negative	onset_negative	18	none			
Condition2	control	onset_scrambled	18	none			
Condition3	rest	rest	0				
Logfile ID format	free						
Log ID	*task-scenes*						
Contrast1	negative vs. control				t	negative	control

Log ID format can either be 'BIDS' or 'free' for this data set.

Table 3. conditions_faces.xlsx

	Name	OnsetID	Duration	OffsetID	ContrastType	ContrastPlus1	ContrastMinus1
Condition1	face	onset_face		rest			
Condition2	control	onset_scrambled		rest			
Condition3	rest	rest	0				
Logfile ID format	BIDS						
Log ID	*task-faces*						
Contrast1	face vs. control				t	face	control

The OffsetID feature is used to define trial durations for illustrative reasons. The program will assume that each trial with OnsetID "onset_face" and "onset_scrambled" ends when the next "rest" event was written to the logfile.

Log ID format can either be 'BIDS' or 'free' for this data set.