
PHYSIO TOOLBOX

QUICKSTART

MANUAL

Author: Lars Kasper, Institute for Biomedical Engineering,
University of Zurich and ETH Zurich

kasper@biomed.ee.ethz.ch

Development: 2008-2016

Last Change: Nov 18th, 2016

TABLE OF CONTENTS

Table of Contents	2
1 Purpose.....	5
2 One-Page Quickstart – SPM Toolbox	6
3 Quickstart – Matlab script (command line).....	8
3.1 Setting scan_timing (sqpar and sync-)parameters.....	8
3.2 Description of Variables: the PhysIO-structure	10
3.3 Example (main_ECG3T.m).....	11
4 Step-by-Step Guide	15
4.1 Interpreting the Output Figures.....	17
5 Citing this work	19
6 Example Datasets	21
6.1 Philips	21
6.1.1 ECG3T	21
6.1.2 ECG7T	23
6.1.3 Pulse Oximeter 3T.....	25
6.1.4 ECG3T_Trigger.....	26
6.2 GE	28

6.2.1	Pulse Oximeter 3T.....	28
6.3	Siemens.....	29
6.3.1	ECG 3T	29
7	Input Structure PhysIO.....	30
7.1	Modules (Overview)	32
	Overview over all sub-modules of the PhysIO Toolbox	32
7.2	save_dir (Module)	32
	Directory where output model, regressors and figure-files are saved to; leave empty to use current directory	32
7.3	log_files (Module)	32
	General physiological log-file information, e.g. file names, sampling rates	32
7.4	scan_timing (Module)	34
	Parameters for sequence timing & synchronization scan_tming.sqpar = slice and volume acquisition starts, TR, number of scans etc. scan_timing.sync = synchronize phys logfile to scan acquisition.....	34
7.5	preproc (Module).....	35
	Preprocessing strategy and parameters for physiological data, starting from raw peripheral measures (preproc.cardiac, preproc.respiration)	35
7.6	Model (Module)	36
	Physiological noise models derived from preprocessed physiological data available models (that can be combined) are:	36
7.7	RETROICOR (Model): Glover et al. 2000	37
	Retrospective image correction method, based on Fourier expansion of cardiac and respiratory phase, plus multiplicative interaction terms (Harvey et al, 2008)	37
7.8	RVT (Model): Respiratory Volume per time model , Birn et al, 2006/8.....	38
7.9	HRV (Model): Heart Rate variability, Chang et al, 2009	38

7.10	noise_rois (Model): Anatomical Component Correction, Behzadi et al, 2007	38
Principal Components of time series of all voxels in given regions of localized noise, e.g. CSF, vessels, white matter	38	
e.g. CompCor: 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, 90-101. doi:10.1016/j.neuroimage.2007.04.042.....	38	
7.11	movement (Model): Regressor model 6/12/24, Friston et al. 1996.....	39
Also: sudden movement exceedance regressors.....	39	
7.12	other (Model): Additional, pre-computed nuisance regressors.....	39
To be included in design matrix as txt or mat-file (variable R)	39	
7.13	verbose (Module)	39
Verbosity of Toolbox, i.e. how many figures shall be generated to visualize the workflow and save it (to graphics file(s))	39	
7.14	ons_secs (Module, output only)	41
Output structure for all read or computed time-dependent variables, i.e. onsets, specified in seconds NOTE: all elements but .raw are cropped to the acquisition window of the session	41	

PURPOSE

This toolbox provides model-based PhysIOlogical noise correction of fMRI data using peripheral measures of respiration and cardiac pulsation. It incorporates noise models of cardiac/respiratory phase (RETROICOR, Glover et al. 2000), as well as heart rate variability and respiratory volume per time (cardiac response function, Chang et. al, 2009, respiratory response function, Birn et al. 2006). The toolbox is usable via the SPM batch editor, performs automatic pre-processing of noisy peripheral data and outputs nuisance regressor files directly suitable for SPM (“multiple_regressors.txt”).

ONE-PAGE QUICKSTART – SPM TOOLBOX

1. Copy the PhysIO Toolbox code folder to the toolbox folder of spm
(Optional) Rename the folder to something meaningful, e.g. PhysIO (see Figure 1).
2. (Re-)Start SPM (spm fmri) and the Batch editor.
3. The PhysIO Toolbox should now occur under SPM -> Tools -> TAPAS PhysIO Toolbox
4. change directory (!) to examples/Philips/ECG3T-folder and load an example spm_job file into the batch editor, e.g : example_spm_job_ECG3T.m
5. Press play!

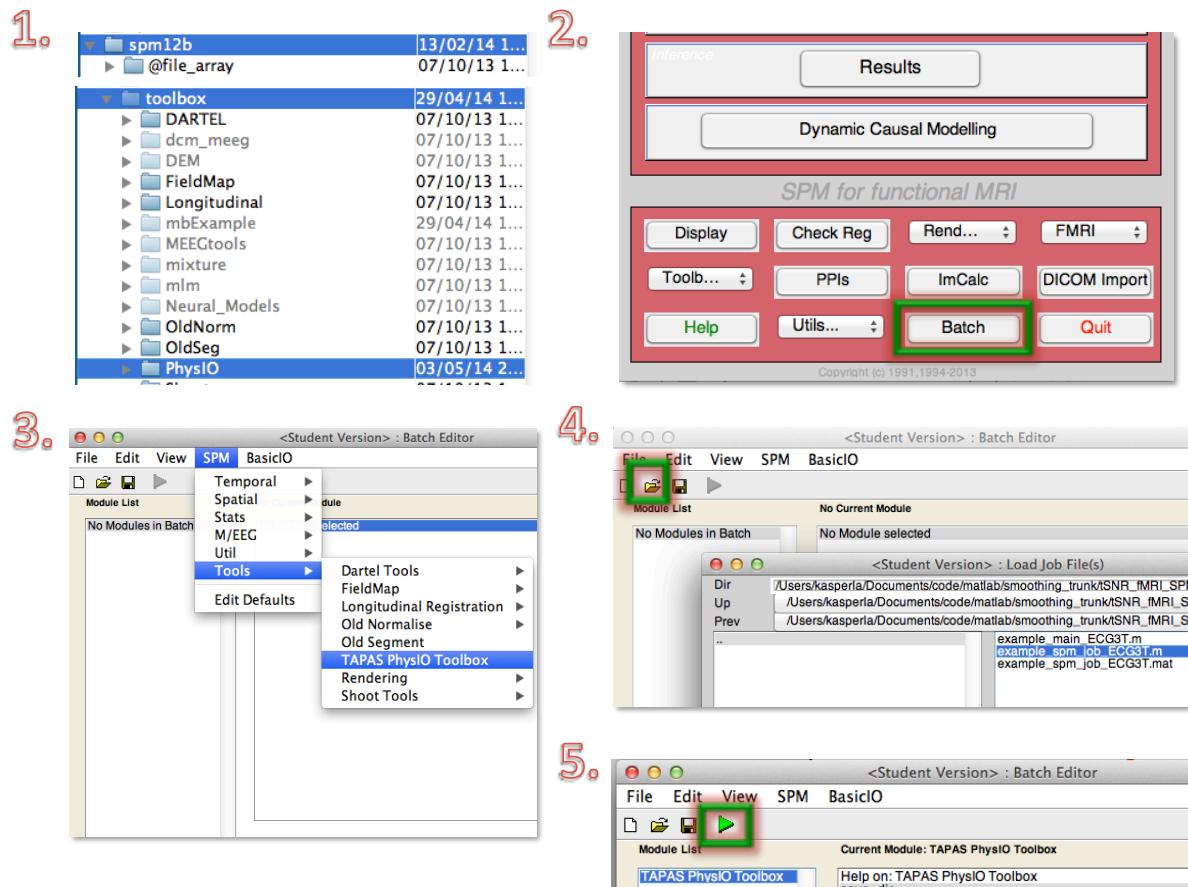


Figure 1. Quickstart PhysIO Toolbox as SPM Toolbox. See Text above

Note: For further information on the PhysIO Toolbox, consult the handbook, the corresponding toolbox journal article (<http://dx.doi.org/10.1016/j.jneumeth.2016.10.019>) or see below.

QUICKSTART – MATLAB SCRIPT (COMMAND LINE)

Adapt main_ECG3T.m (or one of the other main_* example files), especially the sequence parameter of the **sqpar** structure variable and the gradient threshold parameters in the **sync** structure variable. You may set the parameters of each variable either separately, i.e. via sqpar.Nslices = 30; sqpar.Nscans = 320; or call the struct-command in Matlab to set them at once (see Example).

3.1 Setting scan_timing (sqpar and sync-)parameters

The scan timing is defined by the following two structures in PhysIO.scan_timing:

sqpar

- is a structure holding all relevant timing parameters of your MR sequence
- is needed to time the Physiological confound regressors correctly (see chapter 4, Input structures)
- In an ideal world, this is the only structure to be changed in the main_{PPU/ECG3T/ECG7T}.m-example files to run your own logfiles
- In practice, both scan timing and Physiological signal need some preprocessing determined by the thresh-structure.
 - o The need for this preprocessing should be assessed scrutinizing the output plots of the toolbox.

sync

- determines which sampling points of the Physiological logfile will be used for the confound regressor creation.
- can be left empty (=[]) to rely on nominal sequence timing as specified in sqpar, counting volume-TRs
- set for Philips logfiles, if slice/volume scan onsets shall be determined from the logged MR gradient time-course which in the Philips SCANPHYSLOG-file.

- Unfortunately, there is no direct acquisition trigger event logged by Philips, so we have to resort to this workaround finding patterns in the gradient time course relating to slice or volume onsets.
- multiple options for detection and count of scan events (see chapter 4, Input structures, for details)
 - from start or end of the log file
 - detecting different gradient amplitudes or temporal spacing for first and other slices of a volume
 - Figure 2 shall give a visualization of these parameters and shows the example output of ECG_3T (for scan_timing.sync.vol) and ECG_7T (for sync.vol_spacing):

Figure 2: Raw Timecourses of Physiological Logfile

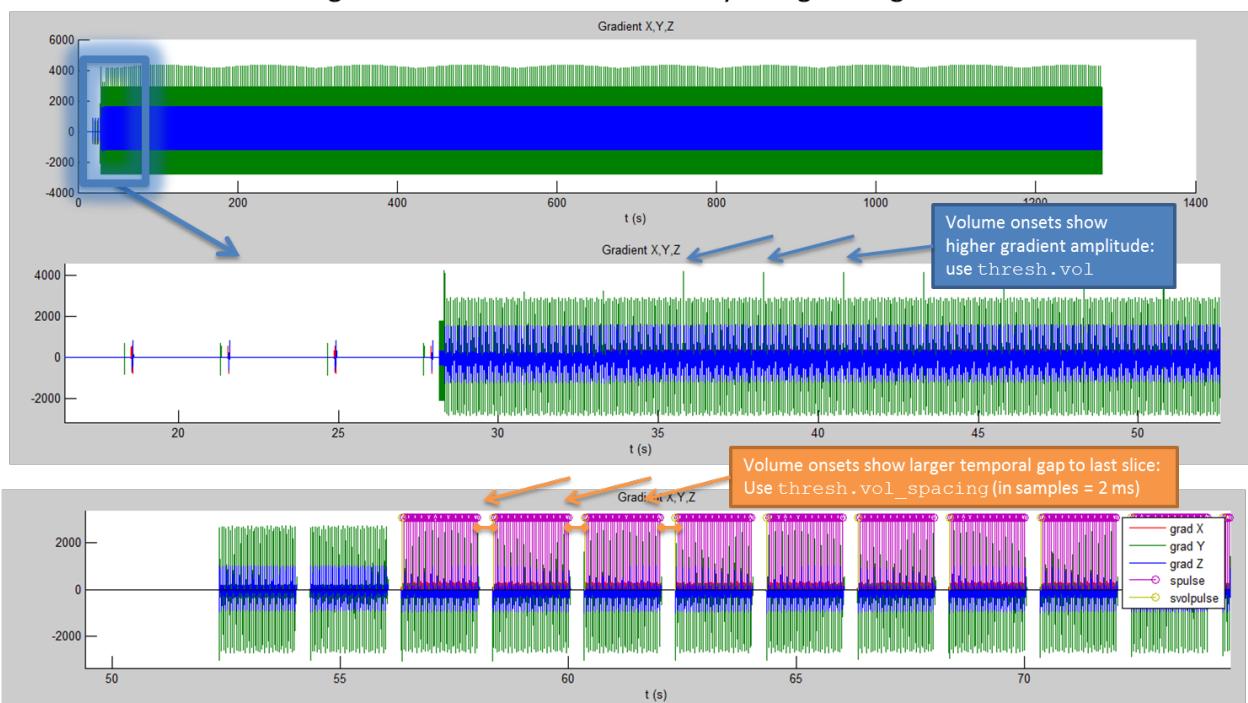


Figure 3: Thresholding Gradient for slice acq start detection

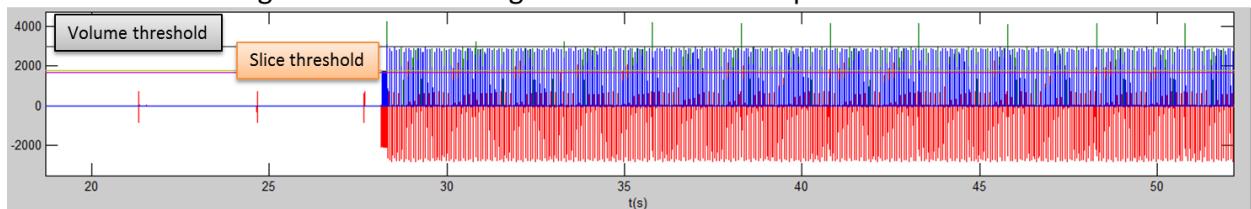


Figure 2. thresh.vol and thresh.vol_spacing: Visualisation when which gradient thresholding shall be used and in which figures the corresponding plots are found.

3.2 Description of Variables: the PhysIO-structure

All parameters are occurring in the example files collected in the PhysIO-structure, which can be created using the command

```
PhysIO = tapas_PhysIO_new();
```

In the body of this function, each parameter is documented with its usage and possible values. Additionally, `tapas_PhysIO_new` can be called with template-names, i.e. typical use cases, e.g. when a manual correction of missed ECG pulses is desired

```
PhysIO = tapas_PhysIO_new();
```

3.3 Example (main_ECG3T.m)

This example can be found in:

examples/Philips/ECG3T/ example_main_ECG3T.m.

See the examples section for details concerning the data.

```

%% 0. Put code directory into path; for some options, SPM should also be in
the path

pathRETROICORcode = fullfile(fileparts(mfilename('fullpath')), ...
'../../../../../code');

addpath(genpath(pathRETROICORcode));

PhysIO      = tapas_PhysIO_new();
log_files   = PhysIO.log_files;
sync        = PhysIO.scan_timing.sync;
sqpar       = PhysIO.scan_timing.sqpar;
preproc     = PhysIO.preproc;
model       = PhysIO.model;
verbose     = PhysIO.verbose;

%% 1. Define Input Files

log_files.vendor           = 'Philips';
log_files.cardiac          = 'SCANPHYSLOG.log';
log_files.respiration       = 'SCANPHYSLOG.log';

%% 2. Define Nominal Sequence Parameter (Scan Timing)

sqpar.Nslices              = 37;
sqpar.NslicesPerBeat        = 37;
sqpar.TR                   = 2.50;
sqpar.Ndummies             = 3;
sqpar.Nscans                = 495;
sqpar.onset_slice           = 19;
sqpar.Nprep                 = [];% set to >=0 to count scans and dummy
                                % volumes from beginning of run, i.e. logfile,
                                % includes counting of preparation gradients
sqpar.TimeSliceToSlice      = sqpar.TR / sqpar.Nslices;

%% 3. Define Gradient Thresholds to Infer Gradient Timing (Philips only)
% 3.1. Determine volume start solely by marking every Nslices-th scan slice
% event as volume event

use_gradient_log_for_timing = true;% true or false

```



```

if use_gradient_log_for_timing
    sync.grad_direction = 'y';
    sync.zero          = 1700;
    sync.slice         = 1800;
    sync.vol           = []; % leave [], if unused; set
                                value >=.slice, if volume
                                % start gradients are higher than
                                slice gradients
    sync.vol_spacing   = []; % leave [], if unused; set to e.g.
                                50e-3 (seconds), if there is a
                                time gap between last slice of a
                                volume & first slice of the next
else
    sync = [];
end

%% 4. Define which Cardiac Data Shall be Used

preproc.cardiac.modality = 'ECG';
preproc.cardiac.initial_cpulse_select.method = 'load_from_logfile';
preproc.cardiac.posthoc_cpulse_select.method = 'off';

%% 5. Order of RETROICOR-expansions for cardiac, respiratory and
%% interaction terms. Option to orthogonalise regressors

model.type = 'RETROICOR';
model.order = struct('c',3,'r',4,'cr',1, 'orthogonalise', 'none');
model.input_other_multiple_regressors = 'rp_fMRI.txt'; % either .txt-file
or .mat-file (saves variable R)
model.output_multiple_regressors = 'multiple_regressors.txt';

%% 6. Output Figures to be generated

verbose.level          = 2; % 0 = none; 1 = main plots (default); 2 =
debugging plots, for setting up new study; 3 = all plots
verbose.fig_output_file = 'PhysIO_output.ps';

%% 7. Run the main script with defined parameters

PhysIO.log_files      = log_files;
PhysIO.scan_timing.sync = sync;
PhysIO.scan_timing.sqpar = sqpar;
PhysIO.preproc        = preproc;
PhysIO.model          = model;
PhysIO.verbose        = verbose;

[PhysIO_out, R, ons_secs] = PhysIO_main_create_regressors(PhysIO);

```

STEP-BY-STEP GUIDE

By answering the following structured questions, you will be able to choose all options of the PhysIO toolbox according to the specific properties of your PhysIOlogical dataset and modeling requirements. They are ordered by the general workflow of the PhysIO toolbox as depicted in Figure 3.

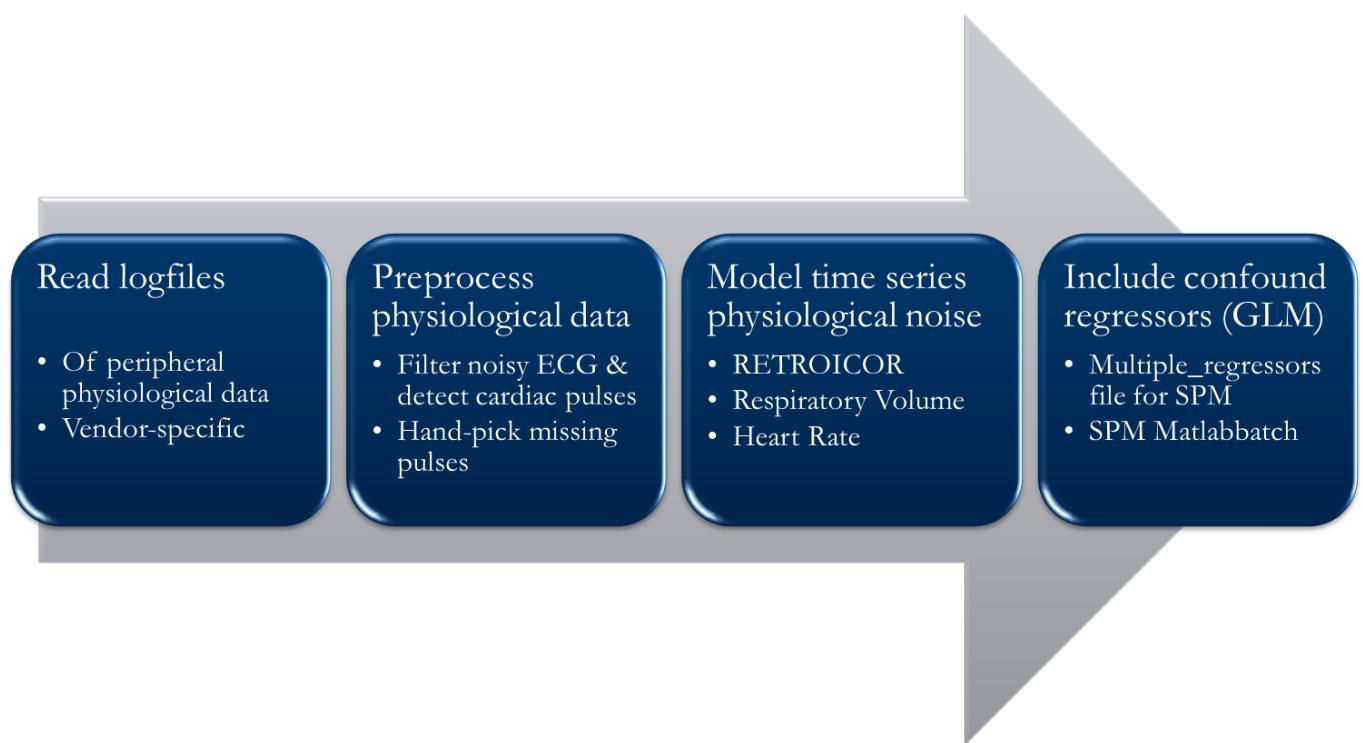


Figure 3: General Workflow of the PhysIO toolbox

- Read logfiles
 - Which vendor is used?
 - Philips
 - GE
 - Siemens
 - How shall the timing of the scan triggers and PhysIOlogical logfile be synchronized?
 - Using a nominal timing
 - Using the gradient time-course (Philips only)

- (Using the gradient-induced peaks in the unfiltered ECG)
- Preprocess Physiological data
 - ECG or PPU?
 - heart beat peaks loaded from logfile (as detected) or initial re-detection
 - post-hoc manual labeling of missing heart beats?

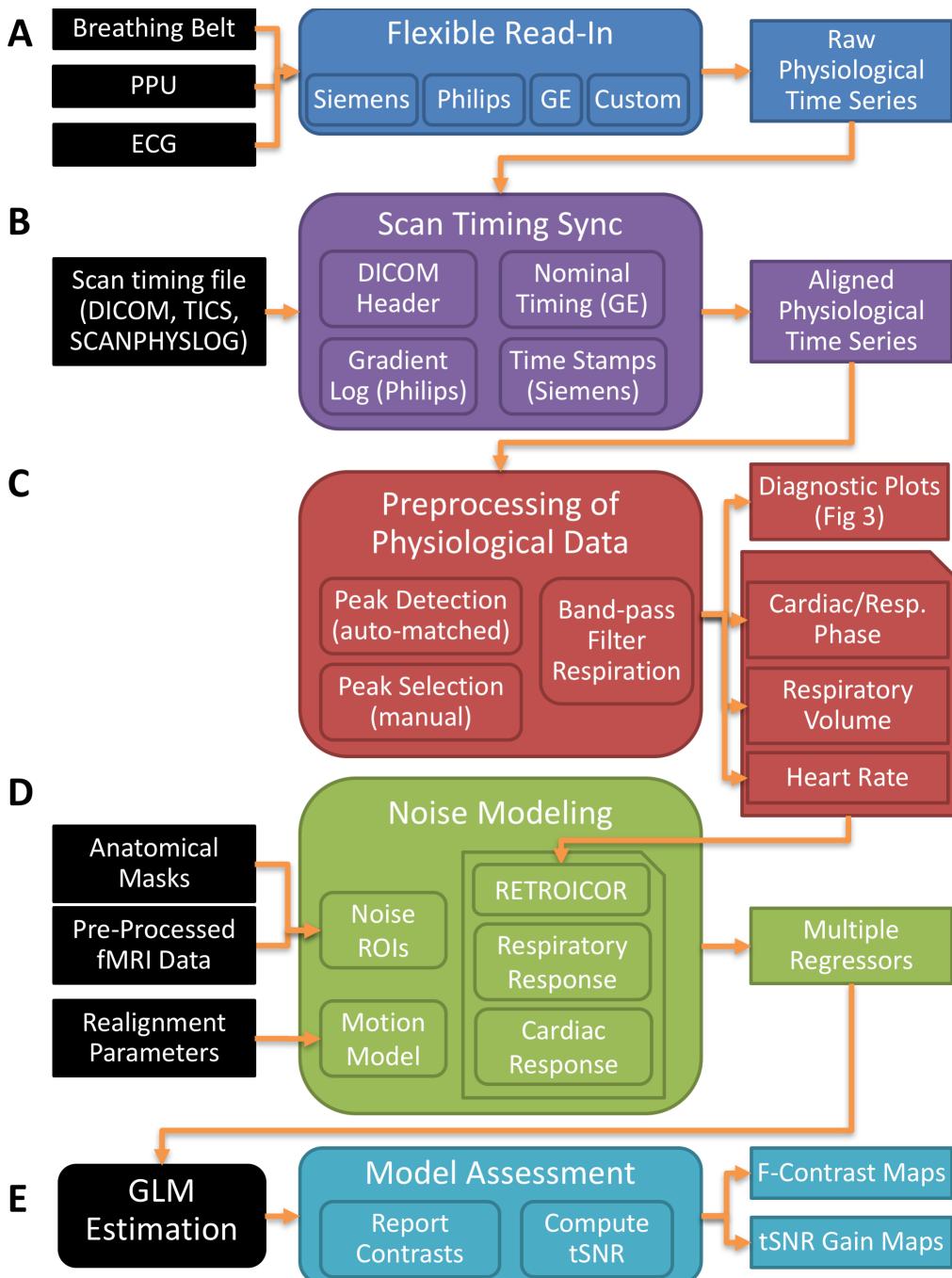


Figure 4: Detailed Workflow and Options of the PhysIO toolbox

4.1 Interpreting the Output Figures

The following figures give an overview of the visual output of the toolbox for correct PhysIOlogical logfile data (from Philips).

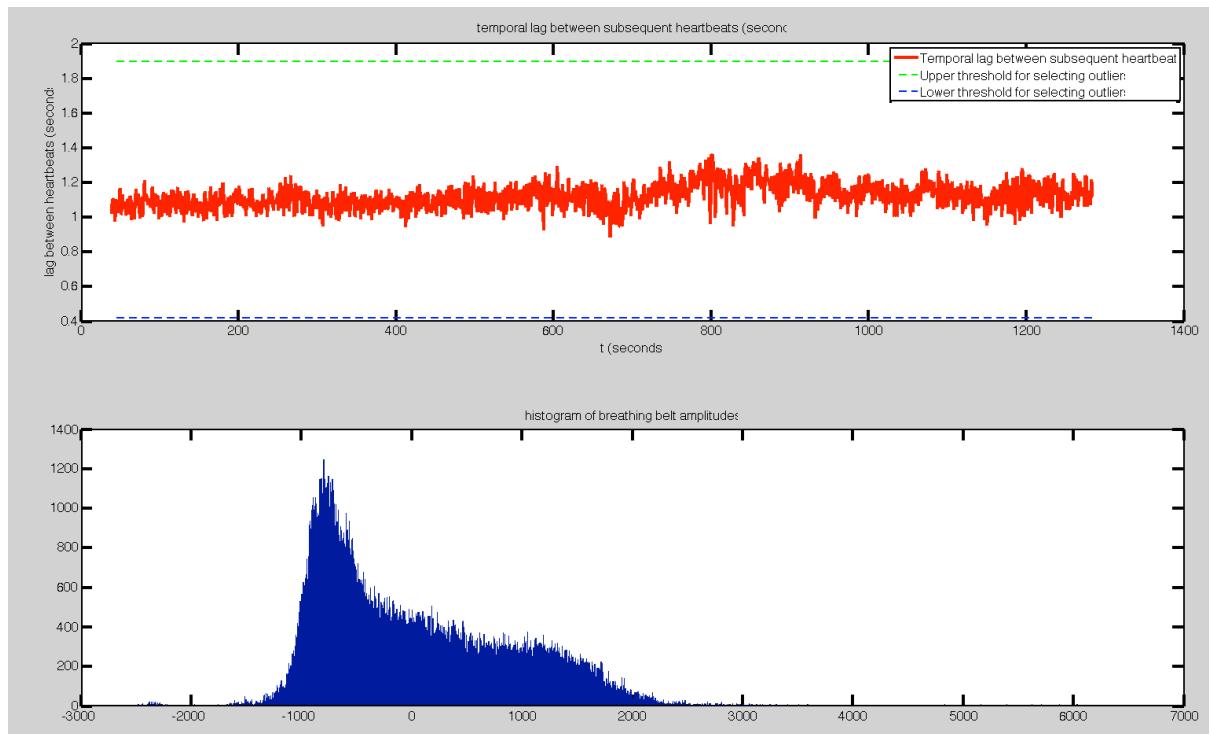


Figure 5. Reference output of scan-timing determined by thresholded, logged gradient time-course.

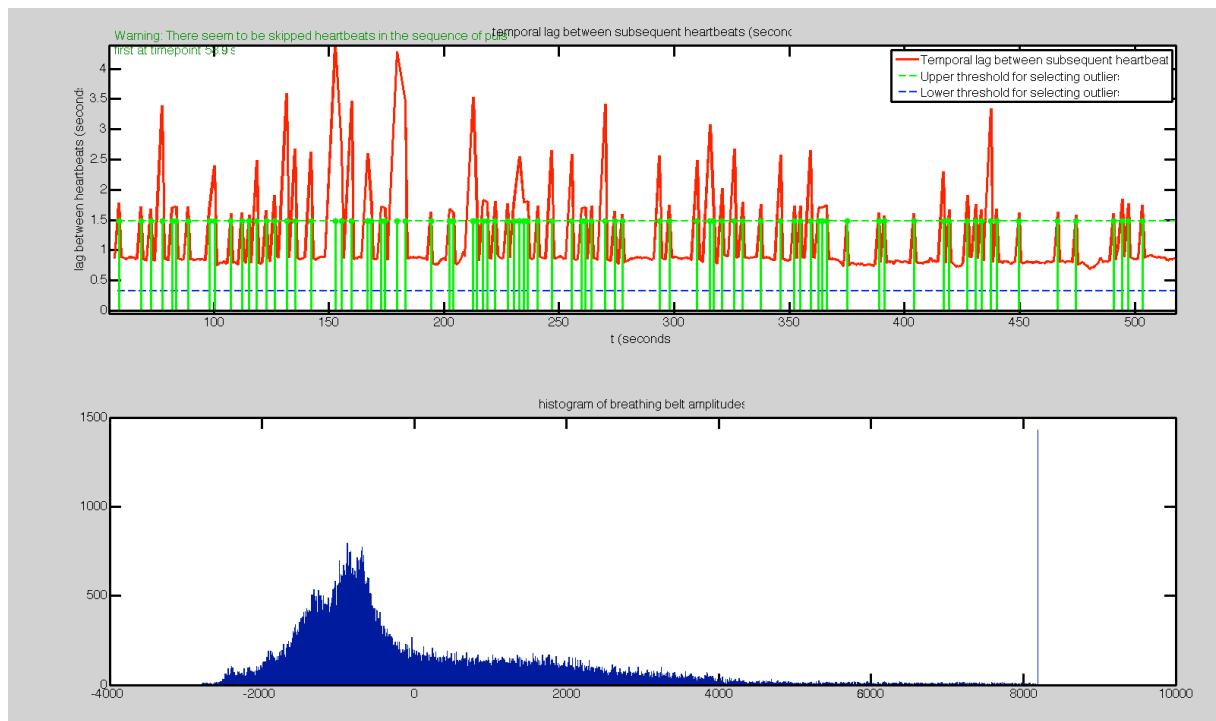


Figure 6. Example output for noisy ECG data, remedy: switch `thresh.cardiac.initial_cpulse_select.method` to ‘auto’

CITING THIS WORK

If you want to use the method implemented in this toolbox, please describe it in your publication as follows:

“Correction for physiological noise was performed via RETROICOR [1,2] using Fourier expansions of different order for the estimated phases of cardiac pulsation (3rd order), respiration (4th order) and cardio-respiratory interactions (1st order) [2]: The corresponding confound regressors were created using the Matlab PhysIO Toolbox ([4], open source code available as part of the TAPAS software collection: <https://www.tnu.ethz.ch/en/software/tapas.html>).”

1. Glover, G.H., Li, T.Q. & Ress, D. Image-based method for retrospective correction of Physiological motion effects in fMRI: RETROICOR. *Magn Reson Med* 44, 162-7 (2000).
2. Hutton, C. et al. The impact of Physiological noise correction on fMRI at 7 T. *NeuroImage* 57, 101-112 (2011).
3. Harvey, A.K. et al. Brainstem functional magnetic resonance imaging: Disentangling signal from Physiological noise. *Journal of Magnetic Resonance Imaging* 28, 1337-1344 (2008).
4. Kasper, L., Bollmann, S., Diaconescu, A.O., Hutton, C., Heinze, J., Iglesias, S., Hauser, T.U., Sebold, M., Manjaly, Z.-M., Pruessmann, K.P., Stephan, K.E., 2016. The PhysIO Toolbox for Modeling Physiological Noise in fMRI Data. *Journal of Neuroscience Methods* accepted. doi:10.1016/j.jneumeth.2016.10.019

This refers to our specific implementation of RETROICOR, which uses Fourier expansions of different order for the estimated phases of cardiac pulsation (3rd order), respiration (4th order) and cardio-respiratory interactions (1st order) following (Harvey et al., 2008). The Respiratory. If you use respiratory-volume-per time (RVT), heart-rate variability (HRV), noise ROIs or 12/24 regressor motion modeling, also include the respective references:

5. 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, 90–101. doi:10.1016/j.neuroimage.2007.04.042
6. Birn, R.M., Smith, M.A., Jones, T.B., Bandettini, P.A., 2008. The respiration response function: The temporal dynamics of fMRI signal fluctuations related to changes in respiration. *NeuroImage* 40, 644–654. doi:10.1016/j.neuroimage.2007.11.059

7. Chang, C., Cunningham, J.P., Glover, G.H., 2009. Influence of heart rate on the BOLD signal: The cardiac response function. *NeuroImage* 44, 857–869. doi:10.1016/j.neuroimage.2008.09.029
8. Siegel, J.S., Power, J.D., Dubis, J.W., Vogel, A.C., Church, J.A., Schlaggar, B.L., Petersen, S.E., 2014. Statistical improvements in functional magnetic resonance imaging analyses produced by censoring high-motion data points. *Hum. Brain Mapp.* 35, 1981–1996. doi:10.1002/hbm.22307

EXAMPLE DATASETS

6.1 Philips

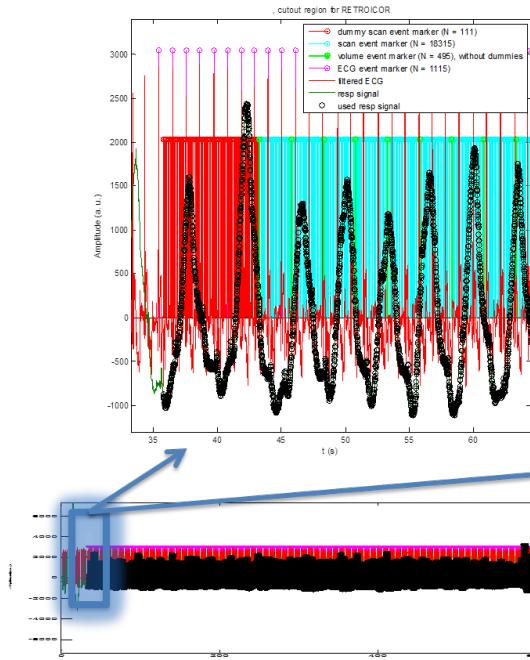
6.1.1 ECG3T

Courtesy of Sandra Iglesias, Translational Neuromodeling Unit, ETH & University of Zurich

4-electrode ECG and breathing belt, Philips 3T Achieva scanner

Description: Standard example; shows how to use scan counting either from beginning or end of run to synchronize PhysIOlogical logfile with acquisition onsets of fMRI scans.

Count scan events from beginning (Nprep is set)



Count Scan Events from end of logfile

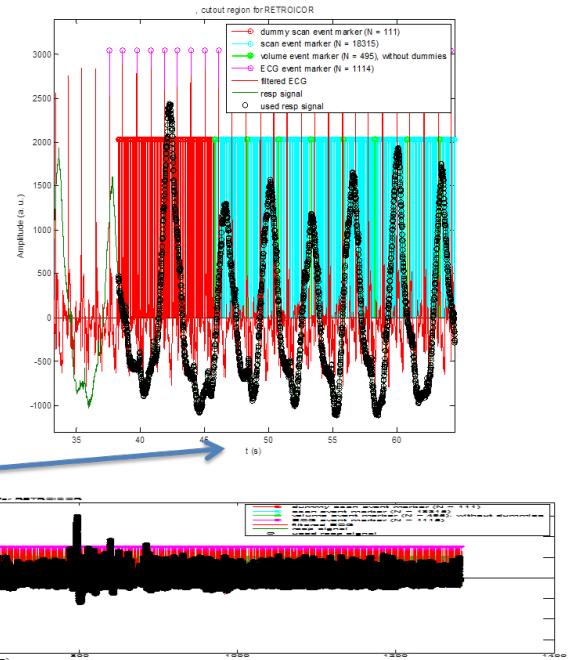


Figure 7: Influence of sqpar.Nprep. If Nprep is set (here = 3) , the scan events including preparation gradients, dummies and scan volumes are counted from the start of the logfile (left), if Nprep is undefined, all is counted relative to the end of the logfile (right).

6.1.2 ECG_{7T}

Courtesy of Zina-Mary Manjaly, University Hospital Zurich

4-electrode ECG and breathing belt, Philips 7T Achieva scanner

Description: The ECG data for ultra-high field data is typically much noisier than at 3 Tesla. Therefore, R-wave peaks are frequently missed by prospective trigger detection and not marked correctly in the logfile. This example shows how to select typical R-wave-peaks manually and let the algorithm find the heartbeat events.

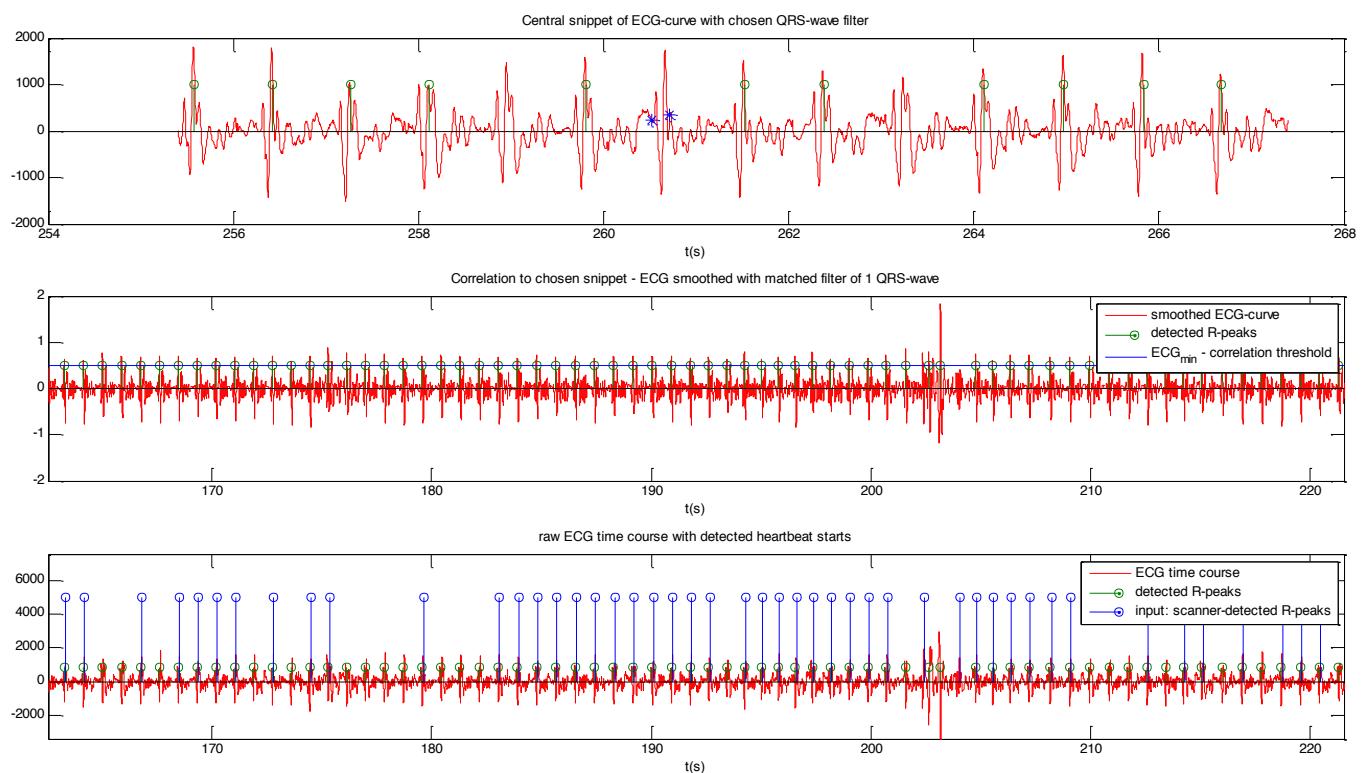


Figure 8: Manual R-peak detection setting ECG_{min} to 0.5. At 7T, this works more reliably than using the scanner logfile (blue stems), which misses some heartbeat events compared to the offline analysis of the script (green stems).

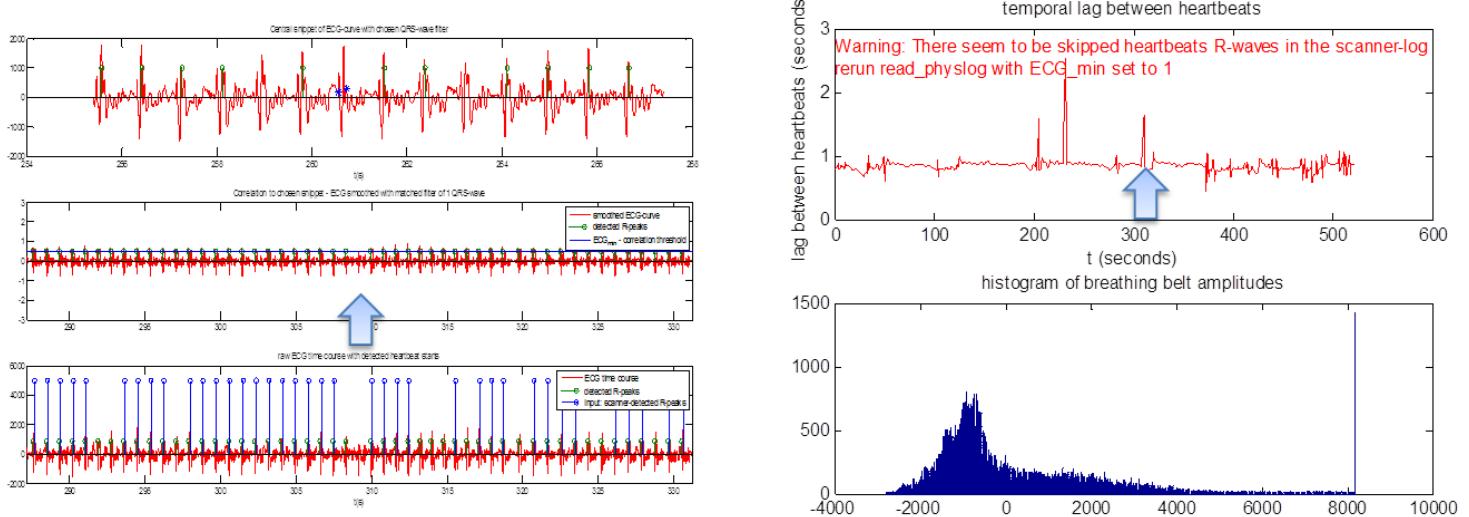


Figure 9: Output of Diagnostic raw time series (right) reveals that not all heartbeats have been detected when using a threshold of ECG_min=0.5 (left).

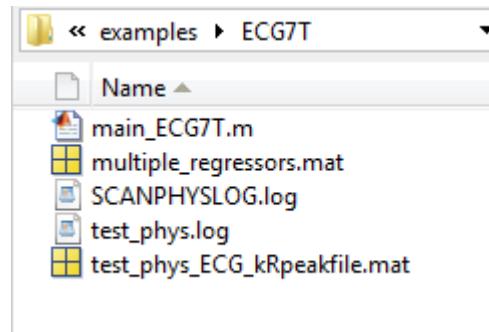


Figure 10: Output files. multiple_regressors contains the R-matrix for a GLM; test_phys.log is the modified SCANPHYSLOG.log now carrying all detected heartbeat and slice scan events and test_phys_ECG_kRpeakfile.mat may be used to rerun R-peak-detection identically

6.1.3 Pulse Oximeter 3T

Courtesy of Diana Wotruba, University and University Hospital of Zurich

PPU (finger plethysmograph) and breathing belt, Philips 3T Achieva scanner

Description: Similar to ECG3T, but a plethysmograph instead of an ECG was used to monitor the cardiac pulsation. Example shows how to extract heart and breathing rate.

6.1.4 ECG3T_Trigger¹

Courtesy of Tobias Hauser, Department of Child- and Adolescent Psychiatry, University of Zurich

Breathing belt, no ECG, Philips 3T Achieva scanner, patch (Roger Luechinger) to log scan event triggers into SCANPHYSLOG

Description: This logfile is very similar to the ECG3T-data above, but it doesn't have an ECG attached. Interestingly, the scan events for every (2nd?) slice as initiated by the Philips scanner are logged in the SCANPHYSLOG-file enabling a direct evaluation of the toolbox' algorithm to detect scan events from gradient time-course. A constant, small offset of 12 ms can be seen, which is constant over the whole session and thus absorbed in the RETROICOR cosine/sine phase expansion.

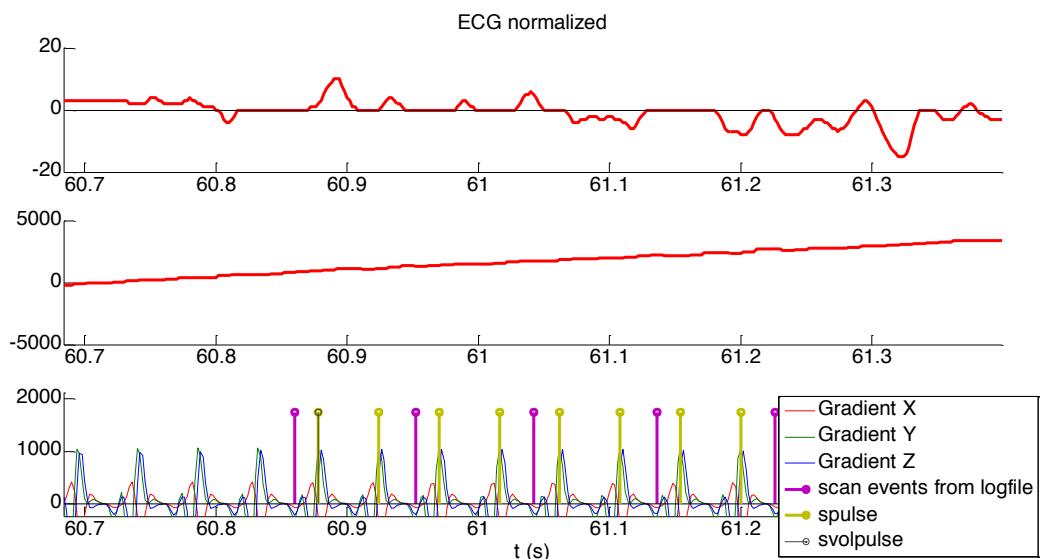


Figure 11. Lower panel: The scan events logged by the Philips system (purple) occur approximately 18 ms before the events detected by the toolbox algorithm, but only for every 2nd slice. This constant offset, however, is absorbed by the phase expansion in cosine and sine regressors later on.

¹ This example dataset is not included in the current release due to space limitations. Write kasper@biomed.ee.ethz.ch to retrieve a version

6.2 GE

6.2.1 Pulse Oximeter 3T

Courtesy of Steffen Bollmann, Kinderspital Zurich and ETH Zurich

PPU (finger plethysmograph) and breathing belt, General Electric 3T scanner

Description: Similar to PPU, but acquired with on a GE system with two separate output logfiles for pulse oximetry and breathing amplitude, sampled with 40 Hz. The quality of the signal is particularly challenging, stemming from a patient population.

6.3 Siemens

6.3.1 ECG 3T

Courtesy of Miriam Sebold, Charite Berlin, and Quentin Huys, TNU Zurich

4-electrode ECG data, Siemens 3T scanner

Description: Similar to ECG 3T, but acquired on a Siemens system with only one logfile for ECG data. The quality of the signal is challenging, stemming from a patient population.

INPUT STRUCTURE PHYSIO

PhysIO is the main input structure to run the PhysIO Toolbox on a particular dataset. Its substructures (log_files, scan_timing, preproc, model, verbose) are introduced in the following subsections and cover different parameter sets for different use cases of the toolbox. These substructures are altered in all examples/main_*.m – files and should be as well in your own scripts, before you call tapas_physio_main_create_regressors.

What follows, is a reformatted listing of the constructor function of the toolbox' core physio-structure, tapas_physio_new, which lists and documents all its modules and their individual parameters.

- [Modules \(Overview\)](#)
- [save_dir \(Module\)](#)
- [log_files \(Module\)](#)
- [scan_timing \(Module\)](#)
- [preproc \(Module\)](#)
- [Model \(Module\)](#)
- [RETROICOR \(Model\): Glover et al. 2000](#)
- [RVT \(Model\): Respiratory Volume per time model , Birn et al, 2006/8](#)
- [HRV \(Model\): Heart Rate variability, Chang et al, 2009](#)
- [noise_rois \(Model\): Anatomical Component Correction, Behzadi et al, 2007](#)
- [movement \(Model\): Regressor model 6/12/24, Friston et al. 1996](#)
- [other \(Model\): Additional, pre-computed nuisance regressors](#)
- [verbose \(Module\)](#)
- [ons_secs \(Module, output only\)](#)

```
function
physio = tapas_physio_new(default_scheme, physio_in)
% Creates complete PhysIO structure fed into tapas_physio_main_create_regressors
%
%
% physio = tapas_physio_new(default_scheme, physio_in)
%
%
% IN
%
% default_scheme - if set, default values for structure entries are set
% according to the application
%
% different templates are predefined, e.g.
%
% 'empty' - all strings are set to '', all
```

```

%
%                               numbers to [] (default)
%
% 'Philips': good initial values for scans acquired
%             on a Philips 3T system
%
% model: RETROICOR;
% vendor: Philips;
%
% heartbeat detection: ECG, load_from_logfile
%                         Philips detected peaks
%                         no posthoc-detection
%
% scan_timing:           gradient_log
%                     uses gradient data
%                     from SCANPHYSLOG-file
%
%
% 'RETROICOR' order of RETROICOR expansion taken from
% Harvey2008, JRM128(6), p1337ff.
%
% 'scan_timing_from_start'
%   'manual_peak_select'
%
% physio_in      - used as input, only fields related to default_scheme
%                   are overwritten, the others are kept as in physio_in
%
%
% OUT
%
% physio         - the complete physio structure, which can be unsed in
%                   tapas_physio_main_create_regressors
%
%
% NOTE
%
% All parameters used in the physIO toolbox are defined AND DOCUMENTED in
% this file. Just scroll down and read through the comments!
%
%
% EXAMPLE
%
% physio = tapas_physio_new('default')
% OR = tapas_physio_new():
%
%
% physio = tapas_physio_new('empty')
% physio = tapas_physio_new('RETROICOR');
% physio = tapas_physio_new('manual peak select', physio);
%
%
% See also tapas_physio_main_create_regressors
%
%
% Author: Lars Kasper
% Created: 2013-04-23
% Copyright (C) 2013 TNU, Institute for Biomedical Engineering, University of Zurich and ETH Zurich.
%
%
% This file is part of the TAPAS PhysIO Toolbox, which is released under the terms of the GNU General
% Public
%
% Licence (GPL), version 3. You can redistribute it and/or modify it under the terms of the GPL
% (either version 3 or, at your option, any later version). For further details, see the file
% COPYING or <http://www.gnu.org/licenses/>.
%
%
% $Id: tapas_physio_new.m 787 2015-07-31 18:35:24Z kasperla $
% if not specified differently, create everything empty
if ~nargin
    default_scheme = 'empty';

```

```

end
%%%%%%%%%%%%%

```

7.1 Modules (Overview)

Overview over all sub-modules of the PhysIO Toolbox

```

%%%%%%%%%%%%%
if nargin >= 2
save_dir      = physio_in.save_dir;
log_files     = physio_in.log_files;
preproc       = physio_in.preproc;
scan_timing   = physio_in.scan_timing;
sync          = scan_timing.sync;
sqpar         = scan_timing.sqpar;
model         = physio_in.model;
verbose       = physio_in.verbose;
ons_secs      = physio_in.ons_secs;
else
    %%%%%%

```

7.2 save_dir (Module)

Directory where output model, regressors and figure-files are saved to; leave empty to use current directory

```

%%%%%%%%%%%%%
save_dir = '';
% Overarching directory, relative to which output files are saved
%%%%%%%%%%%%%

```

7.3 log_files (Module)

General physiological log-file information, e.g. file names, sampling rates

```

%%%%%%%%%%%%%
log_files        = [];           % vendor                         Name depending on your MR Scanner system
%                           'Philips' (default)
%                           'GE',
%%%%%%%%%%%%%

```

```

%
%           'Siemens'
%
%           'Siemens_Tics' - new Siemens physiological
%           logging with time stamps in tics
%           (= steps of 2.5 ms since midnight) and
%           extra acquisition (scan_timing) logfile with
%           time stamps of all volumes and slices
%
%
%           or
%
%           'Custom'
%
%
%   'Custom' expects the logfiles (separate files for cardiac and respiratory)
%   to be plain text, with one cardiac (or
%   respiratory) sample per row;
%   If heartbeat (R-wave peak) events are
%   recorded as well, they have to be put
%   as a 2nd column in the cardiac logfile
%   by specifying a 1; 0 in all other rows
%   e.g.:
%
%       0.2  0
%       0.4  1 <- cardiac pulse event
%       0.2  0
%      -0.3  0
%
%
% NOTE: the sampling interval has to be specified for these files as
% well (s.b.)
log_files.vendor      = 'Philips';
%
% Logfile with cardiac data, e.g.
%   'SCANPHYSLOG<date>.log' (Philips)
%   '<id>_PAV.ecg'          (Siemens Trio etc. (VB))
%   '<date>_ECG1-4.log'      (Siemens Prisma etc (VD))
%   'ECGData_epiRT_<date>' (GE)
log_files.cardiac     = '';
%
% Logfile with respiratory data, e.g. 'SCANPHYSLOG.log';
% (same as .cardiac for Philips)
log_files.respiration = '';
%
% additional file for relative timing information between logfiles and
% MRI scans.
%
% Currently implemented for 2 cases
%
% Siemens:      Enter the first or last Dicom volume of your session here,
%                 The time stamp in the dicom header is on the same time
%                 axis as the time stamp in the physiological log file
%
% Siemens_Tics: log-file which holds table conversion for tics axis to
%                 time conversion
log_files.scan_timing  = '';% Sampling interval in seconds (i.e. time between two rows in
logfile
%
% if empty, default value will be set: 2e-3 for Philips, variable for GE, e.g. 40e-3
%
%       1 entry: sampling interval (seconds)
%       for both cardiac + respiratory log file
%
%       2 entries: 1st entry sampling interval (seconds)
%       for cardiac logfile, 2nd entry for respiratory
%       logfile
log_files.sampling_interval = [];% Time (in seconds) when the 1st scan (or, if existing, dummy)
started,

```

```

% relative to the start of the logfile recording;
% e.g. 0 if simultaneous start
%      10, if 1st scan starts 10
%      seconds AFTER physiological
%      recording
%     -20, if first scan started 20
%      seconds BEFORE phys recording
% NOTE: For Philips SCANPHYSLOG, this parameter is ignored, if
%       scan_timing.sync is set
log_files.relative_start_acquisition = 0;           % Determines which scan shall be aligned to which part
of the logfile
% Typically, aligning the last scan to the end of the logfile is
% beneficial, since start of logfile and scans might be shifted due
% to pre-scans
%
% NOTE: In all cases, log_files.relative_start_acquisition is
%       added to timing after the initial alignment to first/last scan
%
% 'first'   start of logfile will be aligned to first scan volume
% 'last'    end of logfile will be aligned to last scan volume
log_files.align_scan      = 'last';
%%%%%%%%%%%%%%%

```

7.4 scan_timing (Module)

Parameters for sequence timing & synchronization scan_tming.sqpar = slice and volume acquisition starts, TR, number of scans etc. scan_timing.sync = synchronize logfile to scan acquisition

```

%%%%%%%%%%%%%%
scan_timing = struct('sqpar', [], 'sync', []);
scan_timing.sqpar.Nslices          = [];    % number of slices per volume in fMRI scan
scan_timing.sqpar.NslicesPerBeat  = [];    % usually equals Nslices, unless you trigger with the
heartbeat
scan_timing.sqpar.TR              = [];    % volume repetition time in seconds
scan_timing.sqpar.Ndummies        = [];    % number of dummy volumes
% number of full volumes saved (volumes in nifti file,
% usually rows in your design matrix)
scan_timing.sqpar.Nscans         = [];    % Count of preparation pulses
% BEFORE 1st dummy scan.
% Only important, if log_files.scan_align = 'first', since then
% preparation pulses and dummy triggers are counted and discarded
% as first scan onset
scan_timing.sqpar.Nprep          = [];    % time between the acquisition of 2 subsequent
% slices; typically TR/Nslices or minTR/Nslices,
% if minimal temporal slice spacing was chosen
% NOTE: only necessary, if preproc.grad_direction
% is empty and nominal scan timing is used
scan_timing.sqpar.time_slice_to_slice = []; % slice whose scan onset determines the adjustment
of the

```

```

% regressor timing to a particular slice for the whole volume
% volumes from beginning of run, i.e. logfile,
% includes counting of preparation gradients
scan_timing.sqpar.onset_slice      = [];           % Method to determine slice acquisition onset times
% 'nominal'                  derive slice acquisition timing from sqpar
%
% 'gradient'                 derive from logged gradient time courses
%   or 'gradient_log' in SCANPHYSLOG-files (Philips only)
% 'gradient_auto'            derive from logged gradient time courses
%   or 'gradient_log_auto' in SCANPHYSLOG-files automatically, i.e.
%                           without defining thresholds (Philips only)
% s.a. log_files.scan_timing
%
%           individual scan timing logfile with time stamps
%           ("tics") for each slice and volume
%           (e.g. Siemens_Cologne)
scan_timing.sync.method = 'gradient_log'
;    scan_timing.sync.grad_direction = ''
; % 'x', 'y', or 'z';
% if set, sequence timing is calculated
% from logged gradient timecourse along
% this coordinate axis;
scan_timing.sync.zero     = [];   % gradient values below this value are set to zero;
% should be those which are unrelated to slice acquisition start
% minimum gradient amplitude to be exceeded when a slice scan starts
scan_timing.sync.slice    = [];   % minimum gradient amplitude to be exceeded when a new volume
starts;
% leave [], if volume events shall be determined as
% every Nslices-th scan event or via vol_spacing
scan_timing.sync.vol       = [];   % duration (in seconds) from last slice acq to
% first slice of next volume;
% leave [], if .vol-threshold shall be used
scan_timing.sync.vol_spacing = [];
%%%%%%%%%%%%%%%

```

7.5 preproc (Module)

Preprocessing strategy and parameters for physiological data, starting from raw peripheral measures (preproc.cardiac, preproc.respiration)

```

%%%%%%%%%%%%%%
preproc = [];    preproc.cardiac = [];    preproc.cardiac.modality = 'ecg_wifi'
; % 'ECG','ECG_raw', or 'OXY'/'PPU' (for pulse oximetry), 'OXY_OLD', [deprecated]
% The initial cardiac pulse selection structure: Determines how the
% majority of cardiac pulses is detected
% 'auto' - auto generation of representative QRS-wave; detection via
%           maximising auto-correlation with it
% 'load_from_logfile' - from phys logfile, detected R-peaks of scanner
% 'manual' - via manually selected QRS-wave for autocorelations
% 'load'   - from previous manual/auto run

```

```

preproc.cardiac.initial_cpulse_select.method = 'load_from_logfile'
; % file containing reference ECG-peak (variable kRpeak)
% used for method 'manual' or 'load' [default: not set]
% if method == 'manual', this file is saved after picking the QRS-wave
% such that results are reproducible
preproc.cardiac.initial_cpulse_select.file = ''
; % threshold for correlation with QRS-wave to find cardiac pulses
preproc.cardiac.initial_cpulse_select.min = []; % variable saving an example cardiac QRS-wave to
correlate with
% ECG time series
preproc.cardiac.initial_cpulse_select.kRpeak = []; % The posthoc cardiac pulse selection
structure: If only few (<20)
% cardiac pulses are missing in a session due to bad signal quality, a
% manual selection after visual inspection is possible using the
% following parameters. The results are saved for reproducibility
%
% 'off' - no manual selection of peaks
% 'manual' - pick and save additional peaks manually
% 'load' - load previously selected cardiac pulses
preproc.cardiac.posthoc_cpulse_select.method = 'off'
; % filename where cardiac pulses are saved after manual picking
preproc.cardiac.posthoc_cpulse_select.file = ''
; % Suspicious positions of missing or too many cardiac pulses are
% pre-selected by detecting outliers in histogram of
% heart-beat-2-beat-intervals
preproc.cardiac.posthoc_cpulse_select.percentile = 80; % percentile of beat-2-beat interval histogram
that constitutes the "average heartbeat duration" in the session
preproc.cardiac.posthoc_cpulse_select.upper_thresh = 60; % minimum exceedance (in %) from average
heartbeat duration to be classified as missing heartbeat
preproc.cardiac.posthoc_cpulse_select.lower_thresh = 60; % minimum reduction (in %) from average
heartbeat duration to be classified an abundant heartbeat
%%%%%%%%%%%%%%%

```

7.6 Model (Module)

Physiological noise models derived from preprocessed physiological data available models (that can be combined) are:

```

%%%%%%%%%%%%%%
% 'none'      no physiological model is computed; only the read-out
%              logfile data is read out and saved in physio.ons_secs
% 'RETROICOR' as in Glover el al, MRM 44, 20001
%              order of expansion: See Harvey et al, JMRI 28, 2008
% 'HRV'        heart rate variability, as in Chang et al, 2009
% 'RVT'        respiratory volume time, as in Birn et al., 2006/8
% 'movement'   realignment parameters, derivatives,
%              + squared (parameters+derivatives),
%              (Volterra expansion, see: Friston KJ, Williams S, Howard R, Frackowiak
%              RS, Turner R. Movement-related effects in fMRI

```

```

%
% time-series. Magn Reson Med. 1996;35:346?355.)
%
% 'noise_rois' Principal Components of time series of all voxels in given
% regions of localized noise, e.g. CSF, vessels, white
%
% matter
%
% e.g. CompCor: 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, 90?101. doi:10.1016/j.neuroimage.2007.04.042
model = []; model.orthogonalise = 'none'
;
% string indicating which regressors shall be orthogonalised;
% mainly needed, if acquisition was triggered to heartbeat (set to 'cardiac') OR
% if session mean shall be evaluated (e.g. SFNR-studies, set to 'all')
% 'n' or 'none' - no orthogonalisation is performed
% Possible Values (default: 'none'
% 'c' or 'cardiac' - only cardiac regressors are orthogonalised
% 'r' or 'resp' - only respiration regressors are orthogonalised
% 'mult' - only multiplicative regressors are orthogonalised
% 'all' - all physiological regressors are orthogonalised to each other
% 'RETROICOR'
% 'HRV'
% 'RVT'
% 'movement
% noise_rois
% output file for usage in SPM multiple_regressors GLM-specification
% either txt-file or mat-file with variable R
model.output_multiple_regressors = ''
;
% mat-file where whole physio-structure is saved after finishing main.m
model.output_physio = ''
;
% [nScans, nRegressors, nSlices]
% output design matrix of confound regressors,
% saved in 'multiple_regressors.txt'
% or, if multiple slices are specified as onset_slice, in multiple
% multiple_regressors_001.txt files, one per specified slice
model.R = [];

```

7.7 RETROICOR (Model): Glover et al. 2000

Retrospective image correction method, based on Fourier expansion of cardiac and respiratory phase, plus multiplicative interaction terms (Harvey et al, 2008)

```

model.retroicor.include = 1; % 1 = included; 0 = not used
%
% natural number, order of cardiac phase Fourier expansion
model.retroicor.order.c = []; % natural number, order of respiratory phase Fourier expansion
model.retroicor.order.r = []; % natural number, order of cardiac-respiratory-phase-interaction
Fourier expansion
model.retroicor.order.cr = [];

```

7.8 RVT (Model): Respiratory Volume per time model , Birn et al, 2006/8

```
model.rvt.include = 0;      % one or multiple delays (in seconds) can be specified to shift  
% canonical RVT response function from Birn et al, 2006 paper  
model.rvt.delays = 0; % (TODO)
```

7.9 HRV (Model): Heart Rate variability, Chang et al, 2009

```
model.hrv.include = 0;      % one or multiple delays (in seconds) can be specified to shift  
% canonical HRV response function from Chang et al, 2009 paper  
model.hrv.delays = 0; % (TODO)
```

7.10 noise_rois (Model): Anatomical Component Correction, Behzadi et al, 2007

Principal Components of time series of all voxels in given regions of localized noise, e.g. CSF, vessels, white matter

e.g. CompCor: 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, 90-101. doi:10.1016/j.neuroimage.2007.04.042

```
model.noise_rois.include = 0;      % cell of preprocessed fmri nifti/analyze files, from which time  
series  
% shall be extracted  
model.noise_rois.fmri_files = {};% cell of Masks/tissue probability maps characterizing where  
noise resides  
model.noise_rois.roi_files = {};% Single threshold or vector [1, nRois] of thresholds to be  
applied to mask files to decide  
% which voxels to include (e.g. a probability like 0.99, if roi_files  
% are tissue probability maps)  
model.noise_rois.thresholds = 0.9;% Single number or vector [1, nRois] of number of voxels to  
crop per ROI  
% default: 0  
model.noise_rois.n_voxel_crop = 0;% Single number or vector [1, nRois] of numbers  
% integer >=1:    number of principal components to be extracted  
%                  from all voxel time series within each ROI  
% float in [0,1[   choose as many components as needed to explain this  
%                  relative share of total variance, e.g. 0.99 =  
%                  add more components, until 99 % of variance explained  
% NOTE: Additionally, the mean time series of the region is also
```

```
% extracted
model.noise_rois.n_components = 1;
```

7.11 movement (Model): Regressor model 6/12/24, Friston et al. 1996

Also: sudden movement exceedance regressors

```
model.movement.include = 1;      model.movement.file_realignment_parameters = '';
% 0 = no realignment parameters included
% 6 = rotation/translation parameters
% 12 = + derivatives
% 24 = + squared parameters and derivatives, corresponding to a
%       Volterra expansion v_t, v_t^2, v_(t-1), v_(t-1)^2
model.movement.order = 6;
% threshold for large sudden translations; 1 stick regressor for each volume
% exceeding the threshold will be created
model.movement.outlier_translation_mm = 1;
% threshold for large sudden rotations; 1 stick regressor for each volume
% exceeding the threshold will be created
model.movement.outlier_rotation_deg = 1;
```

7.12 other (Model): Additional, pre-computed nuisance regressors

To be included in design matrix as txt or mat-file (variable R)

```
model.other.include = 1;      model.other.input_multiple_regressors = '';
%%%%%%%%%%%%%
```

7.13 verbose (Module)

Verbosity of Toolbox, i.e. how many figures shall be generated to visualize the workflow and save it (to graphics file(s))

```
%%%%%%%%%%%%%
verbose = [];      % verbosity levels:
%-1 = no text or graphics output (text saved in verbose.process_log)
% 0 = no graphical output;
% 1 = (default) main plots : Fig 1: gradient scan timing (if selected) ;
%                                Fig 2: heartbeat/breathing statistics & outlier;
%                                Fig 3: final multiple_regressors matrix
% 2 = debugging plots        for setting up new study or if Fig 2 had
%                                outliers
```

```

%
% Fig 1: raw phys logfile data
% Fig 2: gradient scan timing (if selected)
% Fig 3: cutout interval of logfile for
% regressor creation (including scan timing
% and raw phys data)
% Fig 4: heartbeat/breathing statistics & outlier;
% Fig 5: time course of all sampled RETROICOR
% regressors
% Fig 6: final multiple_regressors matrix
%
%
% 3 = all plots
%
% Fig 1: raw phys logfile data
% Fig 2: gradient scan timing (if selected)
% Fig 3: Slice assignment to volumes
% Fig 4: cutout interval of logfile for
% regressor creation (including scan timing
% and raw phys data)
% Fig 5: heartbeat/breathing statistics & outlier;
% Fig 6: cardiac phase data of all slices
% Fig 7: respiratory phase data and
% histogram transfer function
% Fig 8: time course of all sampled RETROICOR
% regressors
%
% Fig 9: final multiple_regressors matrix
verbose.level = 1; verbose.process_log = cell(0,1); % stores text
outputs of PhysIO Toolbox
%
% processing, e.g. warnings about missed
% slice triggers, peak height etc.
verbose.fig_handles = zeros(0,1); % [nFigs,1] vector; collecting of all generated figure handles
during a run of tapas_physio_main_create_regressors
%
% file name (including extension) where to print all physIO output
% figures to.
% e.g. 'PhysIO_output.ps' or 'PhysIO_output.jpg'
%
% The specified extension determines how the figures will be saved
% .ps - all figures are saved to the same, multi-page postscript-file
% .fig, .tiff, .jpg
% - one file is created for each figure, appended by a figure
% index, e.g. 'PhysIO_output_fig01.jpg'
verbose.fig_output_file = '';
%
% If true, plots are performed in tabs of SPM graphics window
% TODO: implement via [handles] = spm_uitab(hparent,labels,callbacks, ...
% tag,active,height,tab_height)
%
verbose.use_tabs = false;
%%%%%%%%%%%%%%%

```

7.14 ons_secs (Module, output only)

Output structure for all read or computed time-dependent variables, i.e. onsets, specified in seconds NOTE: all elements but .raw are cropped to the acquisition window of the session

```
%%%%%%%%%%%%%
ons_secs          = [];      % read-in data
ons_secs.t        = [];      % time vector corresponding to c and r
ons_secs.c        = [];      % raw cardiac waveform (ECG or PPU)
ons_secs.r        = [];      % raw respiration amplitude time course
ons_secs.c_scaling = 1;      % stores scaling factor for cardiac data
                           % after normalization
ons_secs.r_scaling = 1;      % stores scaling factor for respiratory data
                           % after normalization
                           % processed elements cardiac pulse detection and phase estimations
ons_secs.cpulse   = [];      % onset times of cardiac pulse events (e.g. R-peaks)
ons_secs.fr       = [];      % filtered respiration amplitude time series
ons_secs.c_sample_phase = []; % phase in heart-cycle when each slice of each volume was acquired
ons_secs.r_sample_phase = []; % phase in respiratory cycle when each slice of each volume was
acquired
ons_secs.hr       = [];      % [nScans,1] estimated heart rate at each scan
ons_secs.rvt     = [];      % [nScans,1] estimated respiratory volume per time at each scan
                           % statistical info about physiological data
ons_secs.c_outliers_high = []; % onset of too long heartbeats
ons_secs.c_outliers_low  = []; % onsets of too short heartbeats
ons_secs.r_hist    = [];      % histogram of breathing amplitudes
                           % scan timing parameters
ons_secs.svoltagepulse = []; % [Nscans x 1] onset times of volume scan events
ons_secs.spulse     = [];      % [Nscans*Nslices x 1] onset times of slice (incl. volume) scan
events
ons_secs.spulse_per_vol = []; % cell(Nscans,1), as spulse, holding slice scan events sorted by
volume
                           % uncropped parameters
ons_secs.raw       = [];      % raw read-in version of the whole structure, before any cropping
%
end
%%%%%%%%%%%%%
```