PhysIO Toolbox Documentation

Table of contents

Home

Wiki Main Page

Several documentation files are provided with this toolbox, and are linked in this wiki. They have the extension .md (markdown), i.e. are plain text files, but can be conveniently viewed online as the github/gitlab Wiki. You can find them in `physio/docs`.

List of Documentation files:

- [x] HOME: This page. Landing Page of Wiki. Navigation to all other files and this explanation
- [x] FAQ: Frequently asked questions (for users)
- [x] QUICKSTART: Example script and how to use on test data, Intro to Batch Editor GUI
- [x] EXAMPLES: List and explanation of all examples
- [ ] MANUAL: Reference Manual (mostly for developers) listing all functions, and rationales of the toolbox, disecting its modular structure
- [x] README: from main physio-public repository, similar info to this page
- [x] CHANGELOG: List of all toolbox versions and the respective release notes, i.e. major changes in functionality, bugfixes etc.

Quickstart

Quickstart Manual

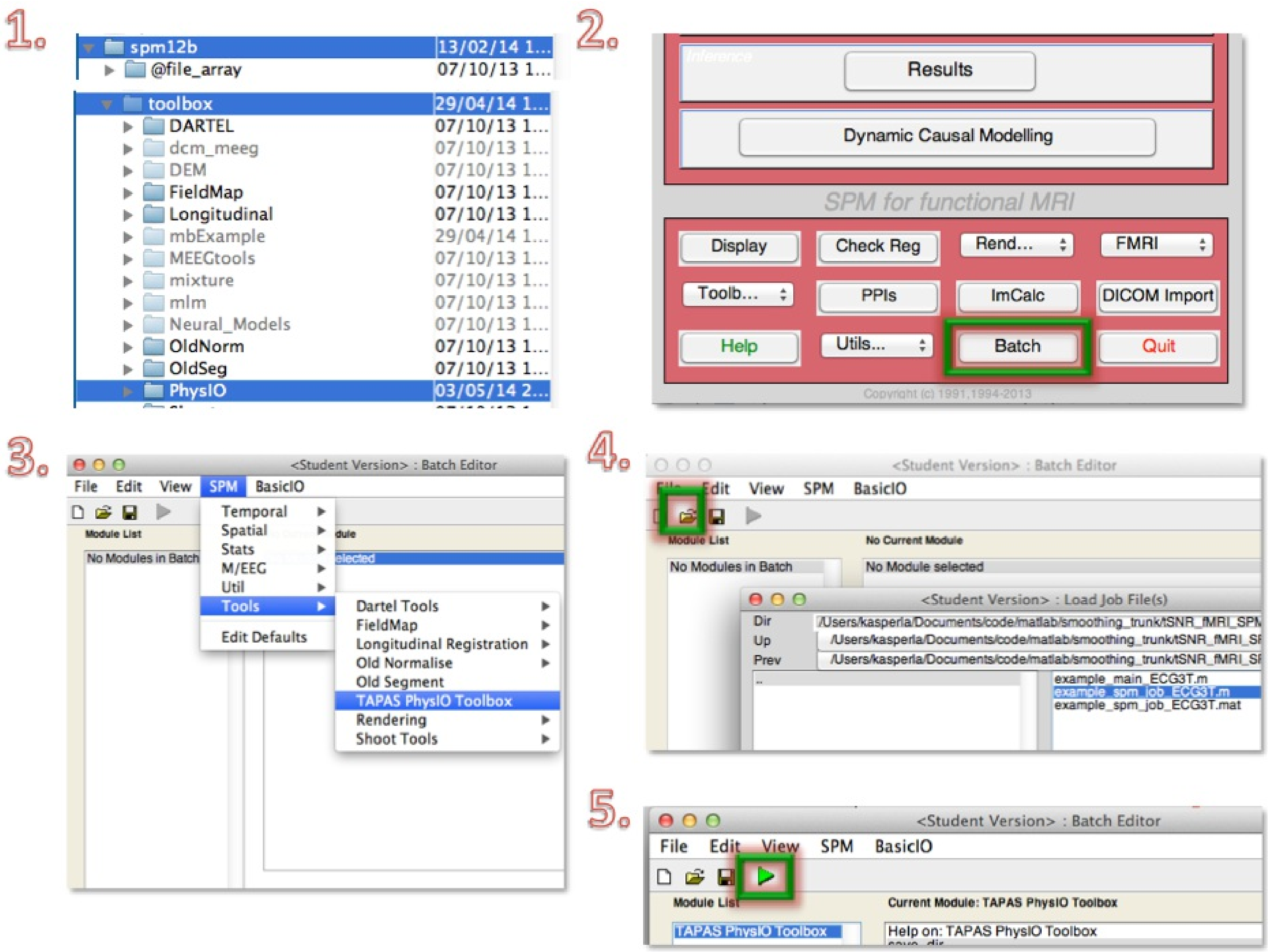
Purpose

This page provides simple walk-throughs of the SPM Batch Editor GUI, the scripts to run the main examples, and the most common output plots of the PhysIO Toolbox.

One-Page-Quickstart (with SPM)

...of SPM Batch Editor GUI for PhysIO Toolbox

1. Copy the PhysIO Toolbox `physio/code` folder to `spm/toolbox`
    - (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!



Readme

TAPAS PhysIO Toolbox Version 2017

> Copyright (C) 2012-2017 Lars Kasper kasper@biomed.ee.ethz.ch
>
> Translational Neuromodeling Unit (TNU)
>
> Institute for Biomedical Engineering
>
> University of Zurich and ETH Zurich

Copying

Purpose

The PhysIO 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).

Installation

Matlab

- Unzip the TAPAS archive
- Add tapas/physio/code to your matlab path

SPM

- Certain functionality (Batch Editor GUI, pipeline dependencies, model assessment via F-contrasts) require the installation of SPM
- Afterwards, the PhysIO Toolbox has to be registered as an SPM toolbox by copying the `physio/code` folder to `spm/toolbox/physio`

Getting Started

Run `example_main_ECG3T.m` in subdirectory `Philips/ECG3T` of the toolbox example repository `physio-examples`. See subdirectory `physio/docs` and next section of this document.

Getting Help/Documentation

Several documentation files are provided with this toolbox. They have the extension .md (markdown), i.e. are plain text files, but can be conveniently viewed online as the github/gitlab Wiki. You can find them in `physio/wikidocs`.

Alternatively, a pdf converted version of the following files is found in `physio/docs/documentation.pdf`

List of Documentation files:

- README.md: this file, purpose, installation, getting started, pointer to more help
- FAQ.md: Frequently asked questions (for users)

- QUICKSTART.md: Example script and how to use on test data, Intro to Batch Editor GUI
- MANUAL.md: Reference Manual (mostly for developers) listing all functions, and rationales of the toolbox, disecting its modular structure
- WIKIMAIN.md: Landing Page of Wiki. Navigation to all other files and this explanation
- EXAMPLES.md: List and explanation of all examples
- CHANGELOG.md: List of all toolbox versions and the respective release notes, i.e. major changes in functionality, bugfixes etc.

Background

The PhysIO Toolbox provides physiological noise correction for fMRI-data from peripheral measures (ECG/pulse oximetry, breathing belt). It is model-based, i.e. creates nuisance regressors from the physiological monitoring that can enter a General Linear Model (GLM) analysis, e.g. SPM8/12. Furthermore, for scanner vendor logfiles (PHILIPS, GE, Siemens), it provides means to statistically assess peripheral data (e.g. heart rate variability) and recover imperfect measures (e.g. distorted R-peaks of the ECG).

Facts about physiological noise in fMRI:

- Physiological noise can explain 20-60 % of variance in fMRI voxel time series (Birn2006, Hutton2011, Harvey2008)
    - Physiological noise affects a lot of brain regions (s. figure, e.g. brainstem or OFC), especially next to CSF, arteries (Hutton2011).
    - If not accounted for, this is a key factor limiting sensitivity for effects of interest.
- Physiological noise contributions increase with field strength; they become a particular concern at and above 3 Tesla (Kasper2009, Hutton2011).
- In resting state fMRI, disregarding physiological noise leads to wrong connectivity results (Birn2006).

Therefore, some kind of physiological noise correction is highly recommended for every statistical fMRI analysis.

Model-based correction of physiological noise:

- Physiological noise can be decomposed into periodic time series following heart rate and breathing cycle.
- The Fourier expansion of cardiac and respiratory phases was introduced as RETROICOR (RETROspective Image CORrection, Glover2000, see also Josephs1997).
- These Fourier Terms can enter a General Linear Model (GLM) as nuisance regressors, analogous to movement parameters.
- As the physiological noise regressors augment the GLM and explain variance in the time series, they increase sensitivity in all contrasts of interest.

Features of this Toolbox

Physiological Noise Modeling

- Modeling physiological noise regressors from peripheral data (breathing belt, ECG, pulse oximeter)
    - State of the art RETROICOR cardiac and respiratory phase expansion
    - Cardiac response function (Chang et al, 2009) and respiratory response function (Birn et al. 2006) modelling of heart-rate variability and respiratory volume per time influence on physiological noise
    - Flexible expansion orders to model different contributions of cardiac, respiratory and interaction terms (see Harvey2008, Hutton2011)
- Data-driven noise regressors
    - PCA extraction from nuisance ROIs (CSF, white matter), similar to aCompCor (Behzadi2007)

Automatization and Performance Assessment

- Automatic creation of nuisance regressors, full integration into standard GLMs, tested for SPM8/12

("multiple_regressors.mat")

- Integration in SPM Batch Editor: GUI for parameter input, dependencies to integrate physiological noise correction in preprocessing pipeline
- Performance Assessment: Automatic F-contrast and tSNR Map creation and display for groups of physiological noise regressors, using SPM GLM tools

Flexible Read-in

The toolbox is dedicated to seamless integration into a clinical research s etting and therefore offers correction methods to recover physiological data from imperfect peripheral measures.

- General Electric
- Philips SCANPHYSLOG files (all versions from release 2.6 to 5.3)
- Siemens VB (files `.ecg`, `.resp`, `.puls`
- Siemens VD (files (`*_ECG.log`, `*_RESP.log`, `*_PULS.log`)
- Biopac .mat-export
    - assuming the following variables (as columns): `data`, `isi`, `isi_units`, `labels`, `start_sample`, `units`
    - See `tapas_physio_read_physlogfiles_biopac_mat.m` for details
- Custom logfiles: should contain one amplitude value per line, one logfile per device. Sampling interval(s) are provided as a separate parameter to the toolbox.

Compatibility and Support

- Matlab Toolbox
- Input:
    - Fully integrated to work with physiological logfiles for Philips MR systems (SCANPHYSLOG)
    - tested for General Electric (GE) log-files
    - implementation for Siemens log-files
    - also: interface for 'Custom', i.e. general heart-beat time stamps & breathing volume time courses from other log formats
- Output:
    - Nuisance regressors for mass-univariate statistical analysis with SPM5,8,12 or as text file for export to any other package
    - raw and processed physiological logfile data
- Part of the TAPAS Software Collection of the Translational Neuromodeling Unit (TNU) Zurich:long term support and ongoing development

Contributors

- Lead Programmer:
    - Lars Kasper, TNU & MR-Technology Group, IBT, University of Zurich & ETH Zurich
- Project Team:
    - Steffen Bollmann, Centre for Advanced Imaging, University of Queensland, Australia
    - Saskia Bollmann, Centre for Advanced Imaging, University of Queensland, Australia
- Contributors:
    - Eduardo Aponte, TNU Zurich
    - Tobias U. Hauser, FIL London, UK

- Jakob Heinzle, TNU Zurich
- Chloe Hutton, FIL London, UK (previously)
- Miriam Sebold, Charite Berlin, Germany

Contact

Send bug reports and suggestions either to 1) our mailing list: tapas@sympa.ethz.ch, or 2) as an issue on our TAPAS github account : https://github.com/translationalneuromodeling/tapas/issues

References

Main Toolbox Reference

1. Kasper, L., Bollmann, S., Diaconescu, A.O., Hutton, C., Heinzle, J., Iglesias, S., Hauser, T.U., Sebold, M., Manjaly, Z.-M., Pruessmann, K.P., Stephan, K.E., 2017. The PhysIO Toolbox for Modeling Physiological Noise in fMRI Data. Journal of Neuroscience Methods 276, 56–72. doi:10.1016/j.jneumeth.2016.10.019

Related Papers (Implemented noise correction algorithms)

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. 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
5. Birn, R.M., Smith, M.A., Jones, T.B., Bandettini, P.A., 2008. The respiration response function: The temporal dynamics of fMRI s ignal fluctuations related to changes in respiration. NeuroImage 40, 644–654. doi:10.1016/j.neuroimage.2007.11.059
6. 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
7. 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

FAQ

Frequently Asked Questions (FAQ)

1. What is the PhysIO Toolbox?

PhysIO is a toolbox for model-based physiological noise correction of fMRI data.

PhysIO stands for Physiological Input/Output toolbox, which summarizes its core purpose. A quote from ourpaper:

> In short, the toolbox transforms physiological input, i.e. peripheral recordings, into physiological output, i.e. regressors encoding components of physiological noise [...] A modular Matlab implementation supports command-line operation and is compatible with all major fMRI analysis packages via the export of regressor text-files. For the Statistical Parametric Mapping SPM software package in particular, PhysIO features a full integration as a Batch Editor Tool, which allows user-friendly, GUI-based setup and inclusion into existing preprocessing and modeling pipelines.

2. How does PhysIO differ from other toolboxes for physiological noise correction for fMRI using peripheral recordings?

Citing from the introduction of our paper again

>

> ### Highlights
>
> - A Toolbox to integrate preprocessing of physiological data and fMRI noise modeling.
> - Robust preprocessing via iterative peak detection, shown for noisy data and patients.
> - Flexible support of peripheral data formats and noise models (RETROICOR, RVHRCOR).
> - Fully automated noise correction and performance assessment for group studies.
> - Integration in fMRI pre-processing pipelines as SPM Toolbox (Batch Editor GUI).

3. How do I cite PhysIO?

The **core reference for PhysIO** is: *The PhysIO Toolbox for Modeling Physiological Noise in fMRI Data* (http://dx.doi.org/10.1016/j.jneumeth.2016.10.019)

Please cite this paper if you use PhysIO in your work. Moreover, this paper is also a good source for more information on PhysIO (see next question).

A **standard snippet to include** in your method section could look like the following, assuming you use 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)

> 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.translationalneuromodeling.org/tapas).

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., Heinzle, J., Iglesias, S., Hauser, T.U., Sebold, M., Manjaly, Z.-M., Pruessmann, K.P., Stephan, K.E., 2017. The PhysIO Toolbox for Modeling Physiological Noise in fMRI Data. Journal of Neuroscience Methods 276, 56–72. doi:10.1016/j.jneumeth.2016.10.019

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:

1. 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

2. Birn, R.M., Smith, M.A., Jones, T.B., Bandettini, P.A., 2008. The respiration response function: The temporal dynamics of fMRI s ignal fluctuations related to changes in respiration. NeuroImage 40, 644–654. doi:10.1016/j.neuroimage.2007.11.059 PhysIO Toolbox | Citing this work 20

3. 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

4. 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

4. Where do I find more documentation for PhysIO?

- The paper describing its structure, objective and modules
- README.md in the main folder when downloading
  - For help on installation and getting started
- Quickstart
  - PDF (or markdown .md file)
  - Tutorial matlab-scripts
- Reference Manual (for developers)

5. I am using FSL, AFNI, BrainVoyager, etc., for my fMRI analyses. Do I need SPM for PhysIO to work?

No, the basic functionality of PhysIO, i.e. creating nuisance regressors for your GLM analysis, is available in plain Matlab. The following extra functionality related to automatizing and assessing noise correction, require the installation of SPM:

- GUI (SPM Batch Editor)
- Pipeline dependencies (automatic input of realignment parameters, feed-in of multiple regressors file to GLM)
- Model assessment via F-tests and automatic F-map/tSNR report
- Noise-ROIs model (read-in of nifti files via SPM)

6. I am using device X for physiological recordings. Does PhysIO support the physiological logfile format Y?

Currently, PhysIO natively supports the following physiological logfile types:

- General Electric
- Philips SCANPHYSLOG files (all versions from release 2.6 to 5.3)
- Siemens VB (files `.ecg`, `.resp`, `.puls`
- Siemens VD (files (`*_ECG.log`, `*_RESP.log`, `*_PULS.log`)
- Biopac .mat-export
  - assuming the following variables (as columns): `data`, `isi`, `isi_units`, `labels`, `start_sample`, `units`
  - See `tapas_physio_read_physlogfiles_biopac_mat.m` for details

Furthermore, physiological recordings can be entered via a *custom* data format, i.e., providing one text file per device. The files should contain one amplitude value per line. The corresponding sampling interval(s) are provided as a separate parameter in the toolbox.

If your favourite logfile format is not supported, please contact the developers. We try everything to accomodate the read-in flexibility of the toolbox to your needs.

7. I am running the toolbox for a lot of subjects / on a remote server without graphics. Can I somehow reproduce the output figures relevant to assess the data quality?

Yes you can, using the toolbox function `tapas_physio_review`. This function takes the physio-structure as an input argument, which is per default saved as `physio.mat` in the specified output folder of your batch job.

8. How do I interpret the various output plots of the toolbox?

Have a look at our publication: *The PhysIO Toolbox for Modeling Physiological Noise in fMRI Data* (http://dx.doi.org/10.1016/j.jneumeth.2016.10.019)

The figures there give a good overview of the toolbox output figures.

9. I want to access subject's physiological measures, e.g. heart rate or respiratory volume (per time), before they enter the regressors. Where can I do that?

All intermediate data processing steps (e.g. filtering, cropping) of the peripheral data, including the computation of physiologically meaningful time courses, such as heart rate and respiratory volume, are saved in the substructure `ons_secs` ("onsets in seconds) of the physio-structure mentioned in question 7. This structure is typically saved in a file `physio.mat`.

`physio.ons_secs` then contains the different time courses, cropped to the acquisition window synchronized to your fMRI scan (the same values before synchronization/cropping, is found in `physio.ons_secs.raw`). Here are the most important ones:

- `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.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
- `ons_secs.c_outliers_high` = []; % onset of too long heart beats
- `ons_secs.c_outliers_low` = []; % onsets of too short heart beats
- `ons_secs.r_hist` = []; % histogram of breathing amplitudes

For a detailed list of all properties and their documentation, read the source code of `tapas_physio_new.m`

10. What is the order of the regressor columns in the multiple regressors file?

This depends on the physiological models (and their order) specified in the `model`-submodule of physio (or in the batch editor). The general order is outlined in Fig. 7A of the http://dx.doi.org/10.1016/j.jneumeth.2016.10.019. The []-brackets indicate the number of regressors:

1. RETROICOR cardiac regressors [2 x nOrderCardiac]

2. RETROICOR respiratory regressors [2 x nOrderRespiratory]
3. RETROICOR cardXResp interaction regressors [4 x nOrderCardiacXRespiratory]
4. HRV [nDelaysHRV]
5. RVT [nDelaysRVT]
6. Noise ROIs (PCA signatures and mean of each region) [nNoiseROIs x (nComponents+1)]
7. Other (included other text file) [nColumnsOtherFile]
8. Motion [6 or 12 or 24, depending on motion model]

If any of the models was not specified, the number of regressors is reduced accordingly.

11. I cannot find the answer to my question in the FAQ. Whom do I ask for help?

Subscribe to our TAPAS mailing list by clicking **Subscribe** on the left side of this website

Afterwards you can send e-mails with your questions to `tapas@sympa.ethz.ch`. Both, core developers of PhysIO and experienced users are receiving your e-mail and are eager to help.

The mailing list also has a searchable archive (click **Archive** on the left of above-mentioned website), where you might already find the answer to your question.

Examples

Example Datasets for PhysIO

The following datasets are available and can be downloaded with the toolbox in a specific `physio-examples` repository, which can be cloned from `git@tnurepository.ethz.ch:physio/physio-examples.git`.

Besides the raw physiological logfiles, each example contains example scripts to run physio as

- SPM job (`*spm_job.mat`)
- editable SPM job (`*spm_job.m`)
- plain matlab script (`*matlab_script.m`)

Philips

ECG 3T

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 volume counting either from beginning or end of run to synchronize physiological logfile with acquisition onsets of fMRI scans.

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.

PPU 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.

General Electric

PPU 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 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.

Siemens

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.

Changelog/Versions

RELEASE INFORMATION

Current Release

PhysIO_Toolbox_R2017.1

February 19, 2017

Minor Release Notes (R2017.1)

- Substantially improved Siemens interface, both for VB/VD and 3T/7T releases
  - several bugfixes
  - based on extensive user feedback from Berlin and Brisbane
- New functionality tapas_physio_overlay_contrasts.m to display non-physio contrasts automatically as well

Major Release Notes (r904 / R2016.1)

- Software version for accepted PhysIO Toolbox Paper: doi:10.1016/j.jneumeth.2016.10.019
- Tested and expanded versions of examples
- Improved stability by bugfixes and compatibility to Matlab R2016
- Slice-wise regressor creation
- Detection of constant physiological time series (detachment, clipping)
- Refactoring of report_contrasts and compute_tsnr_gains as standalone functionality
- Improved Read-in capabilities (Siemens respiration data, BioPac .mat)
- Migration from svn (r904) to git (tnurepository) for version control

Major Release Notes (r835)

- Software version for Toolbox Paper submission
- Noise ROIs modeling
- Extended motion models (24 parameters, Volterra expansion)
- HRV/RVT models with optional multiple delay regressors
- Report_contrasts with automatic contrast generation for all regressor groups
- compute_tsnr_gains for individual physiological regressor groups
- consistent module naming (scan_timing, preproc)
- Visualisation improvement (color schemes, legends)

Minor Release Notes (r666)

- Compatibility tested for SPM12, small bugfixes Batch Dependencies
- Cleaner Batch Interface with grouped sub-menus (cfg_choice)
- new model: 'none' to just read out physiological raw data and preprocess, without noise modelling
- Philips: Scan-timing via gradient log now automatized (gradient_log_auto)
- Siemens: Tics-Logfile read-in (proprietary, needs Siemens-agreement)

- All peak detections (cardiac/respiratory) now via auto_matched algorithm
- Adapt plots/saving for Matlab R2014b

Major Release Notes (r534)

- Read-in of Siemens plain text log files; new example dataset for Siemens
- Speed up and debugging of auto-detection method for noisy cardiac data => new method thresh.cardiac.initial_cpulse_select.method = ???auto_matched???
- Error handling for temporary breathing belt failures (Eduardo Aponte, TNU Zurich)
- slice-wise regressors can be created by setting sqpar.onset_slice to a index vector of slices

Major Release Notes (r497)

- SPM matlabbatch GUI implemented (Call via Batch -> SPM -> Tools -> TAPAS PhysIO Toolbox)
- improved, automatic heartbeat detection for noisy ECG now standard for ECG and Pulse oximetry (courtesy of Steffen Bollmann)
- QuickStart-Manual and PhysIO-Background presentation expanded/updated
- job .m/.mat-files created for all example datasets
- bugfixes cpulse-initial-select method-handling (auto/manual/load)

Major Release Notes (r429)

- Cardiac and Respiratory response function regressors integrated in workflow (heart rate and breathing volume computation)
- Handling of Cardiac and Respiratory Logfiles only
- expanded documentation (Quickstart.pdf and Handbook.pdf)
- read-in of custom log files, e.g. for BrainVoyager peripheral data
- more informative plots and commenting (especially in tapas_physio_new).

Minor Release Notes (r354)

- computation of heart and breathing rate in Philips/PPU/main_PPU.m
- prefix of functions with tapas_*

Major Release Notes (r241)

- complete modularization of reading/preprocessing/regressor creation for peripheral physiological data
- manual selection of missed heartbeats in ECG/pulse oximetry (courtesy of Jakob Heinzle)
- support for logfiles from GE scanners (courtesy of Steffen Bollmann, KiSpi Zuerich)
- improved detection of pulse oximetry peaks (courtesy of Steffen Bollmann)
- improved documentation
- consistent function names (prefixed by "physio_")

NOTE: Your main_ECG/PPU.m etc. scripts from previous versions (<=r159) will not work with this one any more. Please adapt one of the example scripts for your needs (~5 min of work). The main benefit of this version is a complete new variable structure that is more sustainable and makes the code more readable.

License

```
GNU GENERAL PUBLIC LICENSE
  Version 3, 29 June 2007
```



```
        Preamble
```

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

```
      TERMS AND CONDITIONS
```

1. Definitions.

   "This License" refers to version 3 of the GNU General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

   "The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

2. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

3. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

4. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

5. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

6. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is released under this License and any conditions added under section

    1. This requirement modifies the requirement in section 4 to "keep intact all notices".

   c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

   d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

   A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

7. Conveying Non-Source Forms.

   You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

   a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

   b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

   c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

8. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

9. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

10. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

11. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

12. Patents.

    A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

    A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

    Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

    In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

    If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

    If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

    A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

    Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

13. No Surrender of Others' Freedom.

    If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

14. Use with the GNU Affero General Public License.

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

15. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

    Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

16. Disclaimer of Warranty.

    THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

17. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

18. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

```
        END OF TERMS AND CONDITIONS

    How to Apply These Terms to Your New Programs
```

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C)

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and` show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see http://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/philosophy/why-not-lgpl.html.