# ECM

data: 7T, 5min resting state, 1x1x1mm, TR = 2, 130 images, 36 slices

software: LIPSIA (G. Lohmann)
https://github.com/lipsia-fmri/lipsia

assessment of physio data:
https://cni.lin-magdeburg.de/index.php/en/wiki/mri/physiological-data/

additional scripts needed:
https://gitlab.com/estherkuehnneuroscience/fmri_preprocessing
https://gitlab.com/estherkuehnneuroscience/physiodata_preprocessing
https://gitlab.com/estherkuehnneuroscience/eigenvector-centrality-mapping

additional information:
https://fmriprep.readthedocs.io/en/stable/
PhysioIO Toolbox for Matlab
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5489212/pdf/main.pdf

# ECM Individual Native - Updated Version Tübingen

**fMRI preprocessing**

- run main01_prepareMultipleDCMdatasetsBeforeConversion.m
- run main02_SPMconvertMultipleDCMdatasetsToNifti.m
- (run main03_spm_script_slice_timing.m) - do not perform on motion-corrected data
- run movement_evaluation_updated.m
  - diff between neighbouring sample points < 1 mm

**physio data preprocessing**

- preprocess & filter physio data
  - run tk_physionoise.py
  - run retroicor_progressbar_tk.py (jstadler 2011)

**ECM calculation**

- run fsl_avg_masking_updated
  - averaging over time (fsl5.0-fslmaths data.nii -Tmean data_Tmean.nii)
  - brain extraction (fsl5.0-bet2 data_Tmean.nii data_Tmean_brain -f 0.1 -g 0 -m)
- convert moco.mat to txt: load('moco.mat'), dlmwrite('moco.txt', myFile, 'delimiter','\t')
- run vnifti-vpreprocess-vecm-final to regress out moco params, high pass filtering (baseline drift correction), smoothing (2.0 mm), ecm calculation

**view**

- vtimestep -in data.nii -out a.v
- vini -in a.v -z ecmmap-rmres.v &
- type “h” for histogram
- change voxel address an go to maximum (button below image selection window on the top right)

**recommendation by Gaby: NO Slice Time Correction**
use slice time correction on raw data, not on motion corrected data (scanner did no slice time correction, but moco!!)

to map ecm onto individual surface:

#1:
bash script: ecm_map_process (co-register to anatomy, use NN interpolation)

#2:
MIPAV pipeline: Mapping_function_ecm (map registered ecm to surface)

# PhysioNoise - J. Stadler

**settings:**

- keep default settings

**what the script does:**

- adapted from Kelley et al., 2018
- tk_physionoise.py = GUI (includes JS_PhysioNoise_pyLog.py)
- actual script = JS_PhysioNoise_pyLog.py
- preprocess physio data
- extracts respiration and pulse phase values
- generates single respiration and pulse files containing phase values
- generated files include one column containing all phase values for the time series

**Kelley et al. 2008, PlusOne - PhysioNoise Script:**
RW = Downsampled respiratory spline waveform;
RVT = Respiratory volume over time based on peakdet peaks;
RRT = Respiratory rate over time;
CPttl = Cardiac peak based on the TTL pulse from the scanner;
CPd3 = Cardiac peak based on trough of the third derivative;
CPd3R = Cardiac peak R-wave estimate based on small peak of third derivative preceding the CPd3;
CVT = Cardiac volume over time based on peakdet peaks;
CRTttl = Cardiac rate over time based on the TTL pulse from the scanner;
CRTd3 = Cardiac rate over time based on CPd3;
CRTd3R = Cardiac rate over time based on CPd3R).

# RETROICOR - J. Stadler

retroicor_tk.py (includes js_validateentry.py)

settings:

- acq order → interleaved2 (equal # of slices → 2nd slice first / even slices first)
- TR → 2.0
- Freq → 200
- additional delay → 0.050s
  - find pulse offset by calculating residuals for different offsets: retroicor_tk_multidelay.py (generates 4D nii for different offsets)

generated graphics:

- look at Bodeplot and image 3
- bodeplot shows crosstalk between pulse and breathing, e.g. heavy pulsation of abdominal aorta also visible in respiration signal obtained with breathing belt, particularly in very thin people; if pulse finger clip lies on the belly and cable is moved by mere abdominal breathing respiration shows up in pulse signal
- in image 3 control whether maxima were set correctly

what the script does:

- use phase values (0 - 2Pi) to generate 2nd grade Fourier series: sin(x) + cos(x) + Sin(2x) + Cos(2x) for every single slice
- results go into GLM and add one column filled with “1” as average
- residuals are taken as cleaned / filtered data
- additional pulse offset implemented, because pulse was measured at the fingertip with pulse oxi → pulse wave arrives at different times in the brain and the finger→ shift the pulse phase results to minimize residuals in filtered epi data (until shifting is too much and variance in the data becomes greater again)

**IMPORTANT**
Pulse was measured at the fingertip. The pulse wave from the heart arrives earlier in the brain than in the finger. Respiration on the other hand directly biases magnetic field and therefore appears immediately in the MR signal.

# How to find pulse offset

**Pulse Offset** (theoretical considerations)
= difference in pulse transit time (PTT) between brain and finger (dPTT) → pulse reaches brain earlier than finger

PTT brain = 250ms, PTT finger = 275ms, dPTT = 25ms


References:
Bammer, R., Holdsworth, S. J., Aksoy, M., & Skare, S. (2010). Phase Errors in Diffusion Weighted Imaging. *Diffusion MRI: Theory, Methods, and Applications*, 218-249.

Kortekaas, M. C., van Velzen, M. H., Grüne, F., Niehof, S. P., Stolker, R. J., & Huygen, F. J. (2018). Small intra-individual variability of the pre-ejection period justifies the use of pulse transit time as approximation of the vascular transit. *PloS one*, *13*(10), e0204105.

van Velzen MHN, Loeve AJ, Niehof SP, Mik EG. (2017) Increasing accuracy of pulse transit time measurements by automated elimination of distorted photoplethysmography waves. Med Biol Eng Compu:1–12. https://doi.org/10.1007/s11517-017-1642-x.

van Velzen, M. H., Stolker, R. J., Loeve, A. J., Niehof, S. P., & Mik, E. G. (2019). Comparison between pulse wave velocities measured using Complior and measured using Biopac. *Journal of clinical monitoring and computing*, *33*(2), 241-247.


**Pulse Offset** (empirical approach)
= optimal value maximizes pulse-related variance → minimizing residuals

Procedure:

- use python script retroicor_tk_multidelay.py (generates a 4D nii with timepoints as 4th dimension containing cardio only residuals)
- test how different pulse offset values affect residuals:
  - timepoints = # of steps
  - step = step size in s
  - additional delay = starting point
- open fsleyes to explore images:
  - check time course for one voxel within blood vessel (supposed to be nice parable) → minimum is offset with highest correlation
- generate separate niftis for different offsets (e.g., additional delay = 0.025s, timepoints = 1, step = 0.00)
  - look at histograms, compare peak values and find minimum
- results: minimum at 25ms (below 25 ms explained variance increases, above 25ms explained variance decreases), offset of 0ms and 50ms explain lowest amount of variance

**How to choose pulse offset?**

→ compare GM/WM contrast of ecm maps in ROI for different offsets

Liebe Esther,

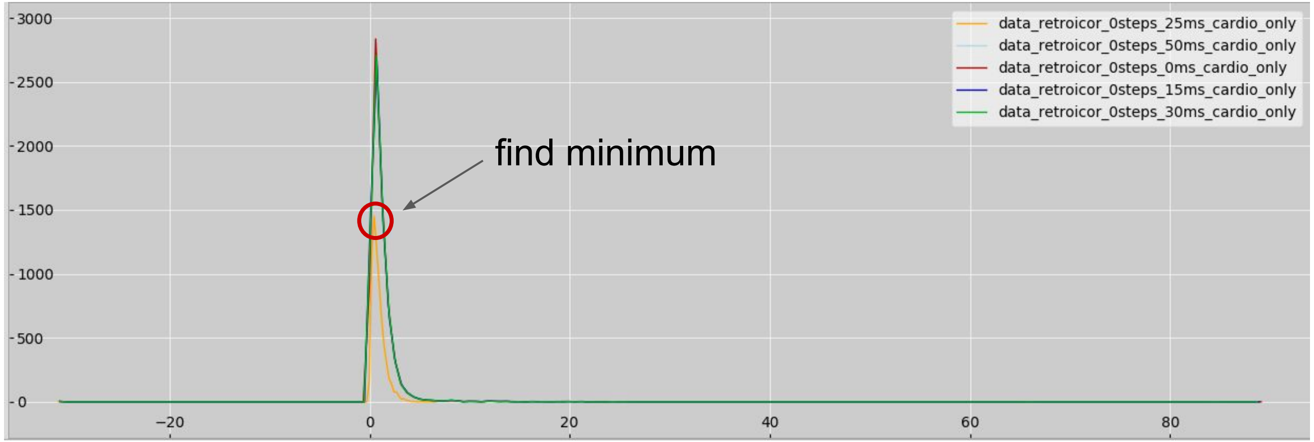
ob ich einen Offset von 10, 25, 50, oder 100 ms wähle, macht für den GM/WM Kontrast der ecm maps in S1 keinen Unterschied. Verglichen zu keinem Offset, verbessern alle 4 Werte gleichermaßen den Kontrast. Zumindest für meinen Testdatensatz.

Jörg hatte wohl damals in seinen Daten einen Offset von etwa 100ms identifiziert. Allerdings würden meine Daten eher für 10ms sprechen.

Hi Juliane,

Ja, wir hatten ja schon besprochen, dass kleine Veränderungen des offsets vermutlich nicht viel ausmachen bei einer sampling rate von 2 Sekunden. Die zeitliche Auflösung, dass wir zwischen 10 und 100 ms unterscheiden können, haben wir nicht. Wir nehmen ja nur alle 2 Sekunden ein Bild auf. Wenn du auch im statistsischen Ergebnis keinen Unterschied siehst, würde ich pragmatisch die Mitte nehmen und **50 ms wählen**. Was für immer einen „Fehler“ wir damit haben, er beträgt somit maximal 50 ms, die keine Rolle spielen sollten.

# How to find pulse offset




histogram generated with fsleyes