PESTICA for AFNI

an overview and tutorial

PESTICA=Physiologic EStimation by Temporal ICA

Comments/questions/requests/bugs:
contact Erik Beall – ebeall@gmail.com
Also see http://www.nitrc.org/projects/pestica

Overview of PESTICA package

- Contains scripts and templates for
  - PESTICA cardiac and respiratory estimation
  - Physiologic noise correction
    - RETROICOR and IRF-RETROICOR, an improved noise regression method
      - *IRFRET removes less degrees of freedom (DOF)*
  - Quality assurance tools
    - QA is dependent on using IRFRET

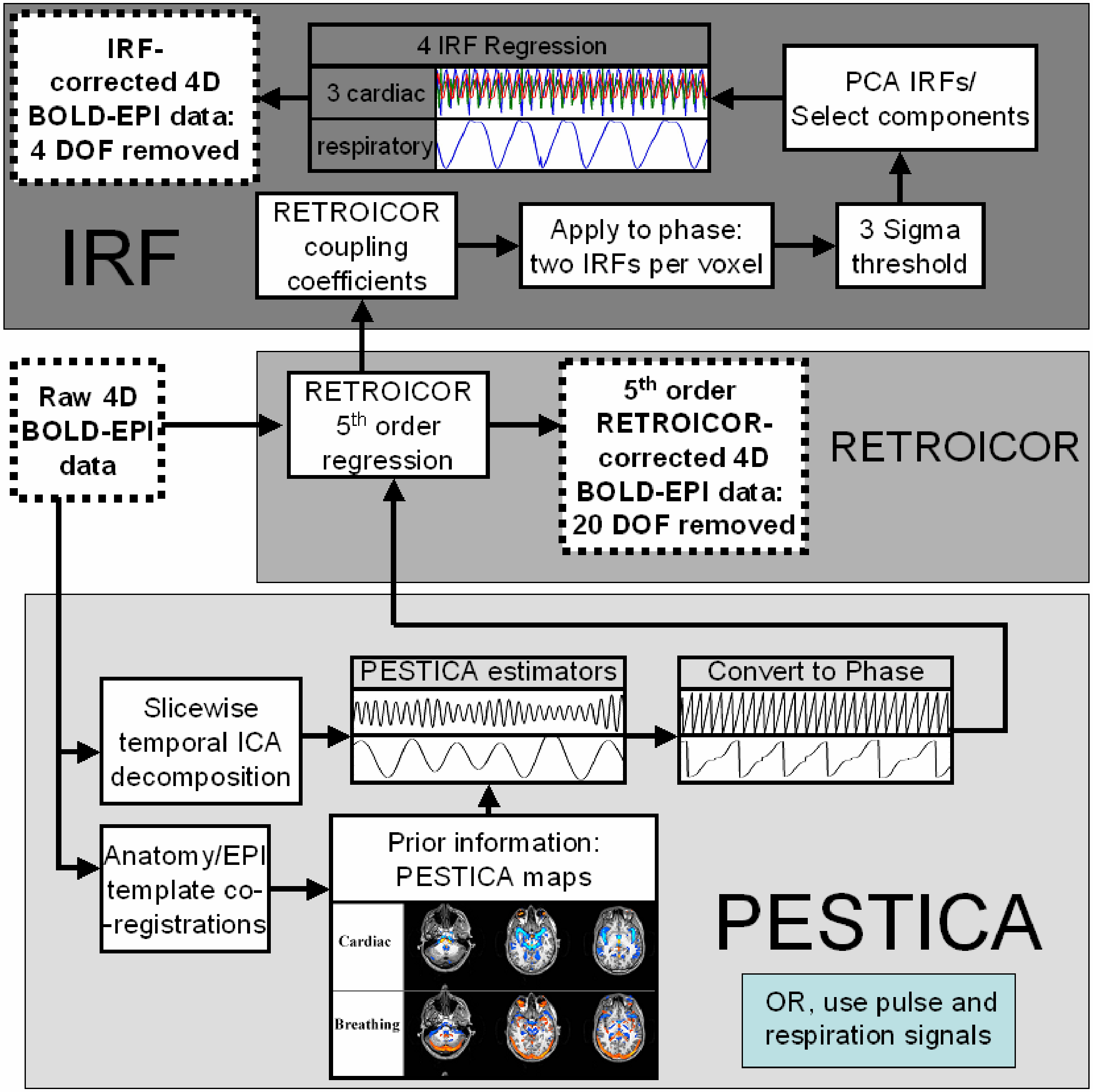
- NOTE: requires AFNI, MATLAB and AFNI_matlab

Overview

Step 3: IRF-RETROICOR

Step 2: RETROICOR

Step 1: PESTICA



IRF-corrected 4D BOLD-EPI data: 4 DOF removed

4 IRF Regression

3 cardiac

respiratory

PCA IRFs/ Select components

IRF

RETROICOR coupling coefficients

Apply to phase: two IRFs per voxel

3 Sigma threshold

Raw 4D BOLD-EPI data

RETROICOR 5th order regression

5th order RETROICOR-corrected 4D BOLD-EPI data: 20 DOF removed

RETROICOR

Slicewise temporal ICA decomposition

PESTICA estimators

Convert to Phase

Anatomy/EPI template co-registrations

Prior information: PESTICA maps

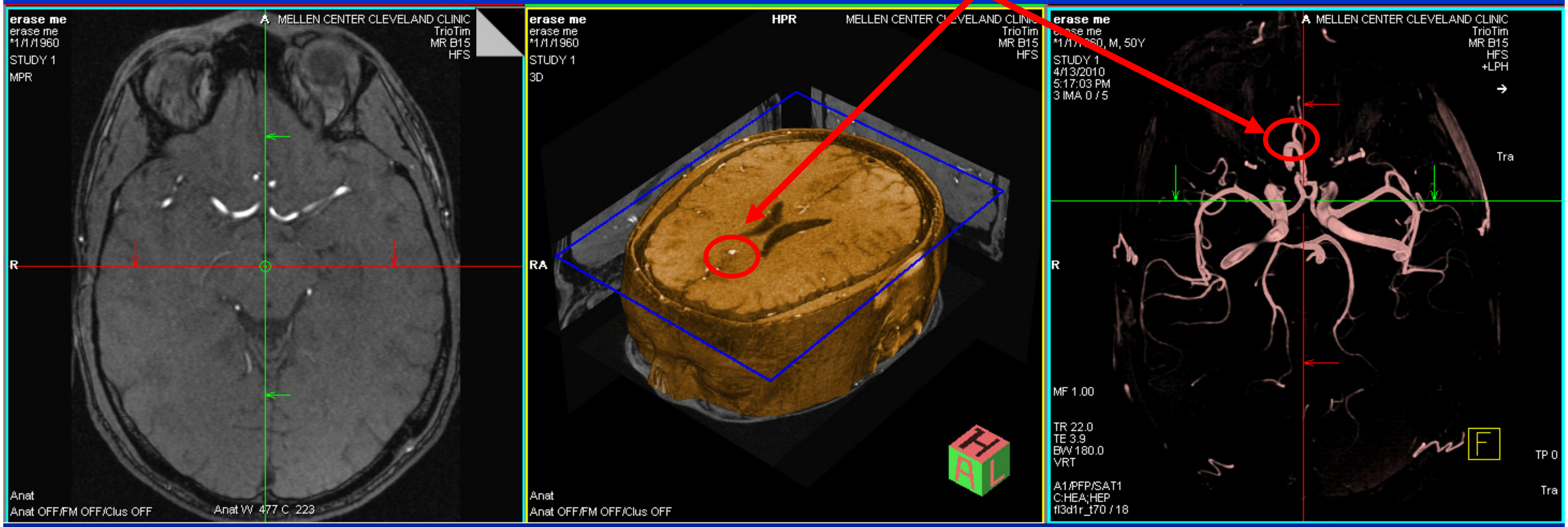
PESTICA

OR, use pulse and respiration signals

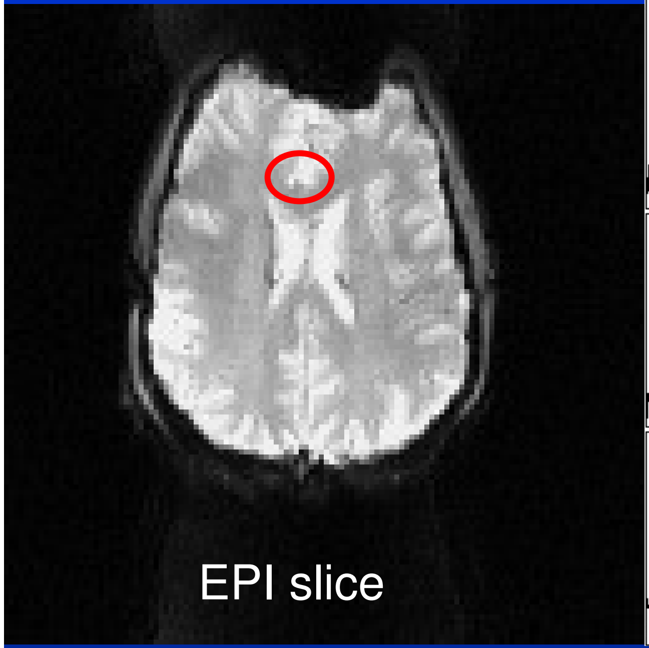
Physiologic noise in BOLD MR

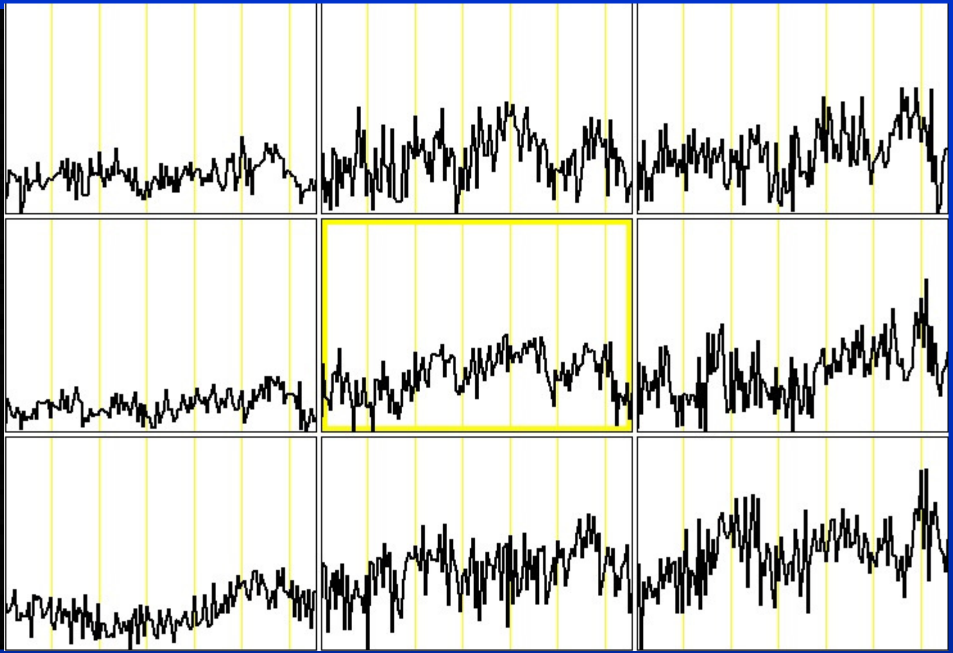
- Non-neuronal noise from heart/breathing
  - reduces specificity and significance in fMRI
  - spatially varying bias in connectivity data
- Correction 1: temporally filter the data
  - Depends on sampling > Nyquist rate of noise
  - Rare to have whole-brain sampling high enough for higher harmonics of cardiac noise
- Correction 2: regress out noise
  - Depends on having a model of the noise
  - We will get this using PESTICA…

Cardiac: arterial locations on an MRA, note **anterior cerebral artery** (ACA)

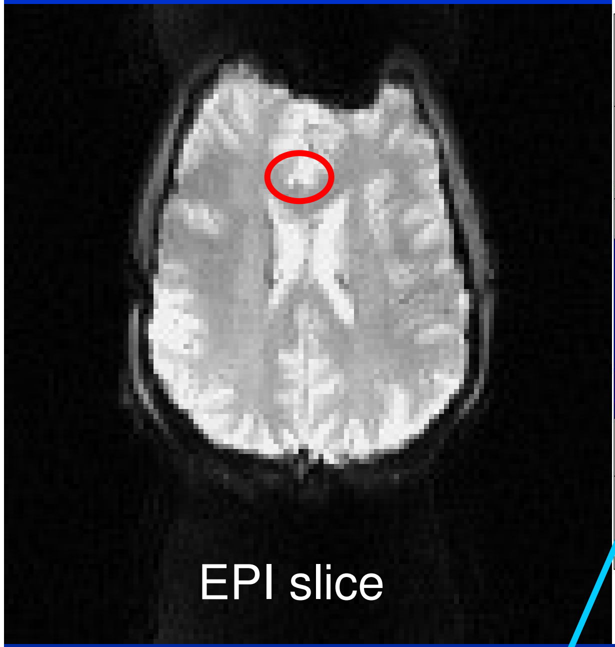


- During every heartbeat systole, blood flow and pressure increases briefly.
- In fast imaging, artifactual signal is phase-locked with the cardiac cycle.

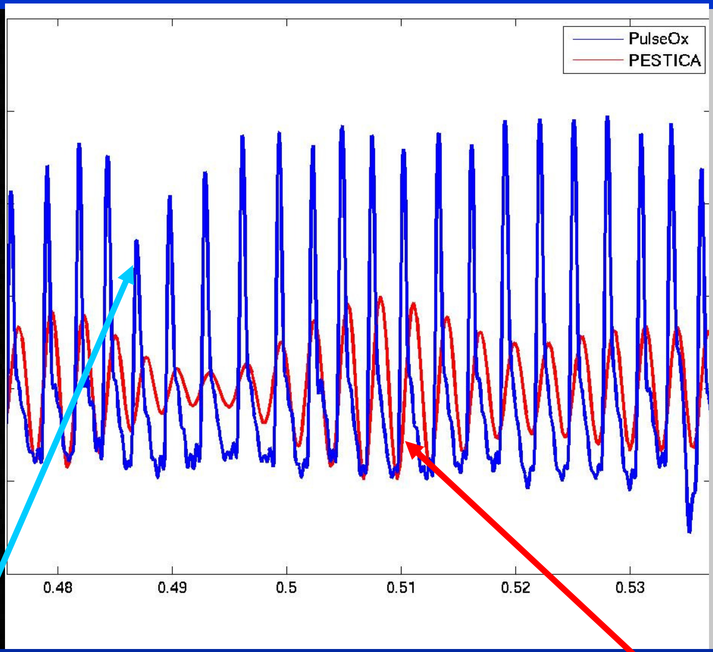


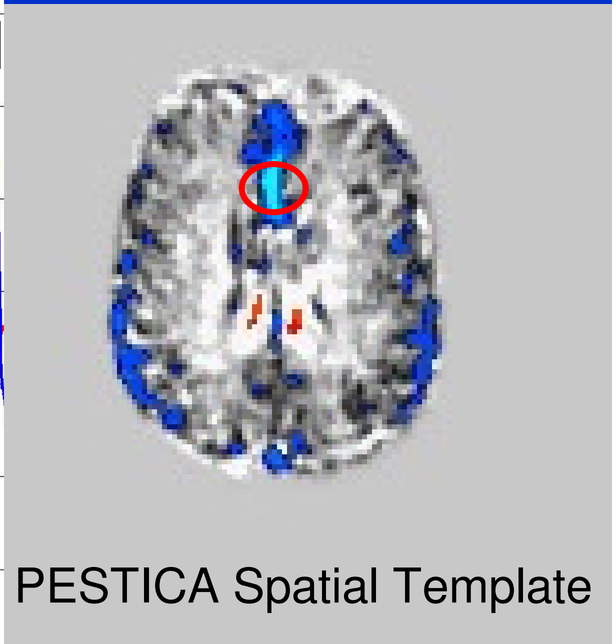


- ACA in echoplanar (EPI) timeseries
- Cardiac noise near ACA doubles or triples variance in those voxels (AFNI plot of 9 voxels centered on ACA voxel)
- Need to measure pulse if we want to account for it…



EPI slice



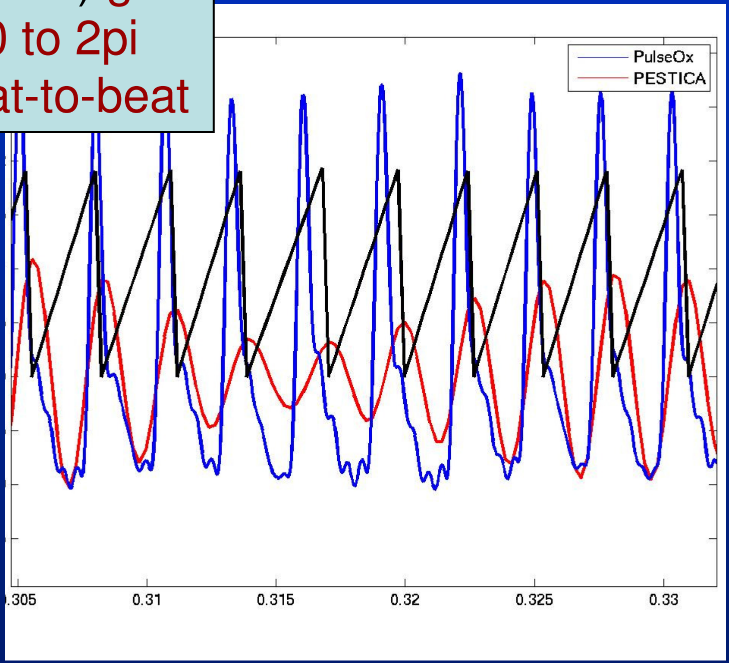


PESTICA Spatial Template

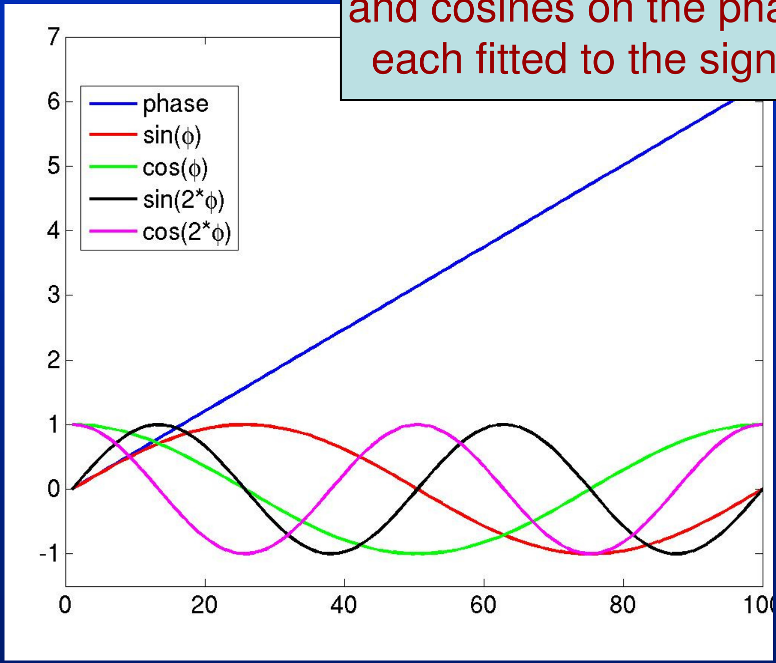
- Spatial template from PESTICA used to track cardiac
- Pulse Oximeter signal displayed with PESTICA-derived pseudo pulse ox signal
  - Same periodicity - See **Lowe08, Wu08 Wu09, Beall10**.
  - PESTICA gives us the periodicity, we must still model the beat-to-beat noise …

- RETROspective Image-based CORrection
- If noise signature is same beat-to-beat, use model based on the relative phase within a cycle: Fourier series of phase.

Phase (black) goes from 0 to 2pi from beat-to-beat



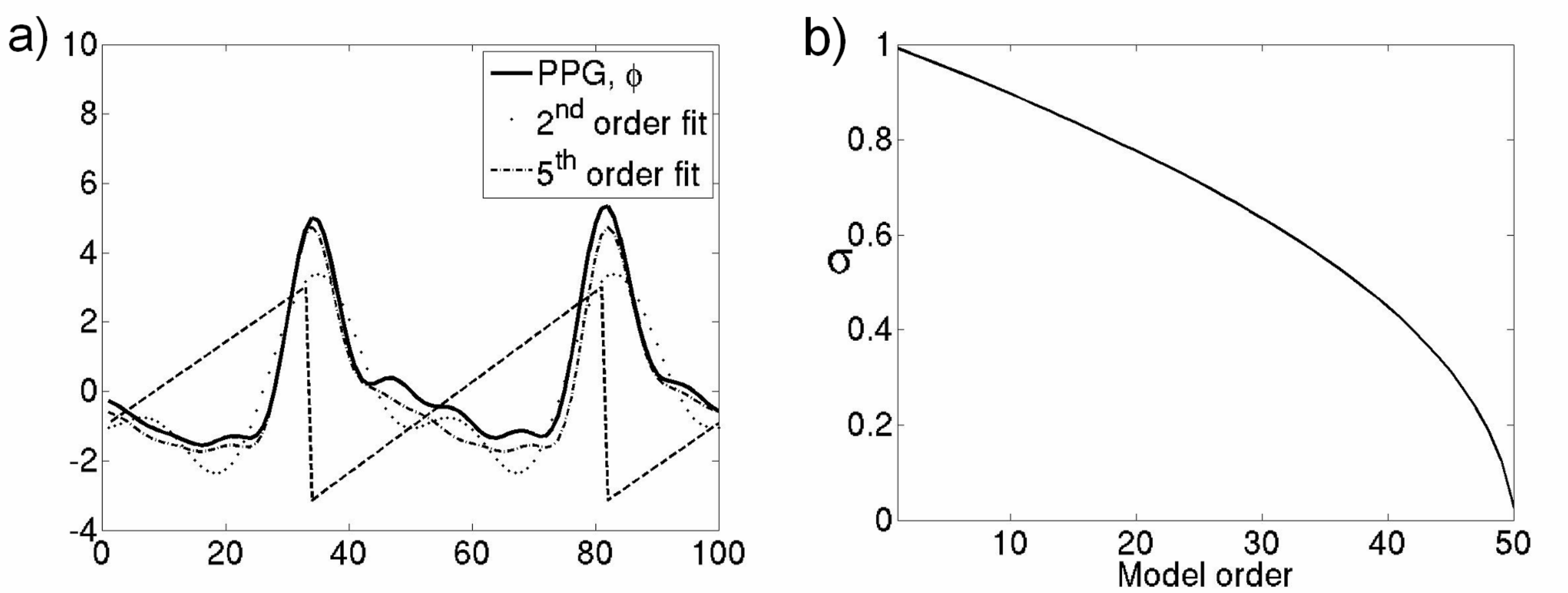
Fourier Series Expansion is M orders of sines and cosines on the phase, each fitted to the signal.



- Do the same for respiratory noise: convert respiratory signal to phase, then 2 to 5 orders of Fourier series on it. This is **RETROICOR**.

Higher order terms

- RETROICOR model is general and effective.
- However, to get ***all*** the noise at higher field strengths, need 3rd-5th order series.
  - RETROICOR **Glover00** at 1.5T recommended 2nd order, **Harvey08** recommends 4th order cardiac, 3rd order respiration but cautions about overfitting data.
  - We observe 5th order for both retains significant coupling, but removes a lot of non-noise variance.
- That is a lot of regressors: 5th order cardiac+5th order respiratory=20 regressors



- Increased model order increases accuracy of fit (left fig)
- As model order increased, standard deviation decreases, ***<u>even when we're regressing random noise which has nothing to do with our dataset!</u>***
  - Eventually there is nothing left…(right fig is random phase and data with 100 timepoints, 50th order=100 regressors)
- This means we're ***unavoidably damaging our data at a level dependent on the number of regressions***

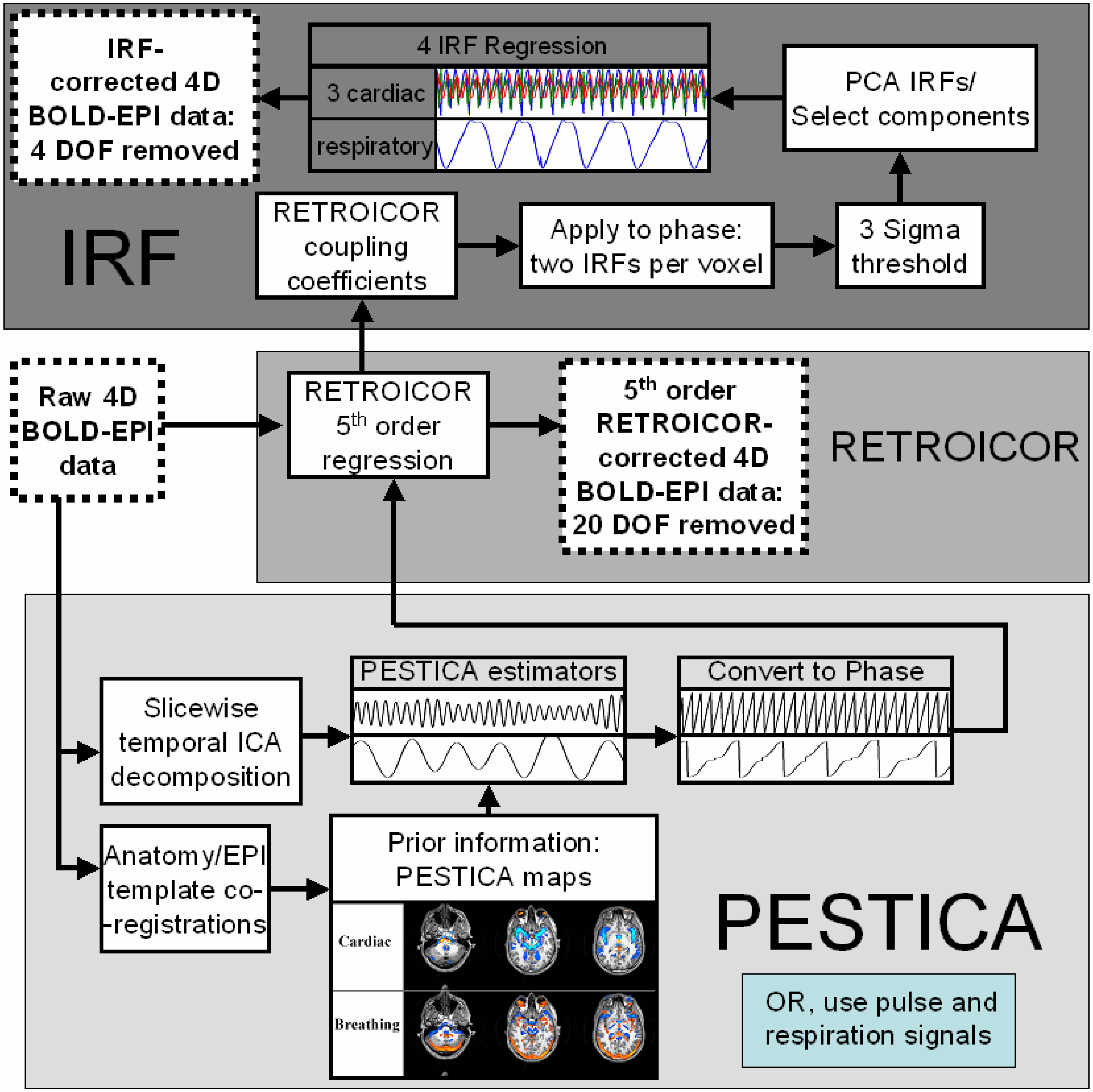
Impulse Response Functions (IRF)

- Each voxel produces a different shape noise response in the fitting process, we want to select only those fits that explain signal over and above background noise
  - Take fitted RETROICOR coefficients, but pass to Principal Component Analysis
  - Keep only those that explain most of variance
    - use Monte Carlo to set 3sigma threshold
  - End up with conservative set of noise response shapes: IRFs

3: IRF-RETROICOR

2: RETROICOR

1: PESTICA



IRF

IRF-corrected 4D BOLD-EPI data: 4 DOF removed

4 IRF Regression

3 cardiac

respiratory

PCA IRFs/ Select components

RETROICOR coupling coefficients

Apply to phase: two IRFs per voxel

3 Sigma threshold

RETROICOR

Raw 4D BOLD-EPI data

RETROICOR 5th order regression

5th order RETROICOR-corrected 4D BOLD-EPI data: 20 DOF removed

PESTICA

Slicewise temporal ICA decomposition

PESTICA estimators

Convert to Phase

Anatomy/EPI template co-registrations

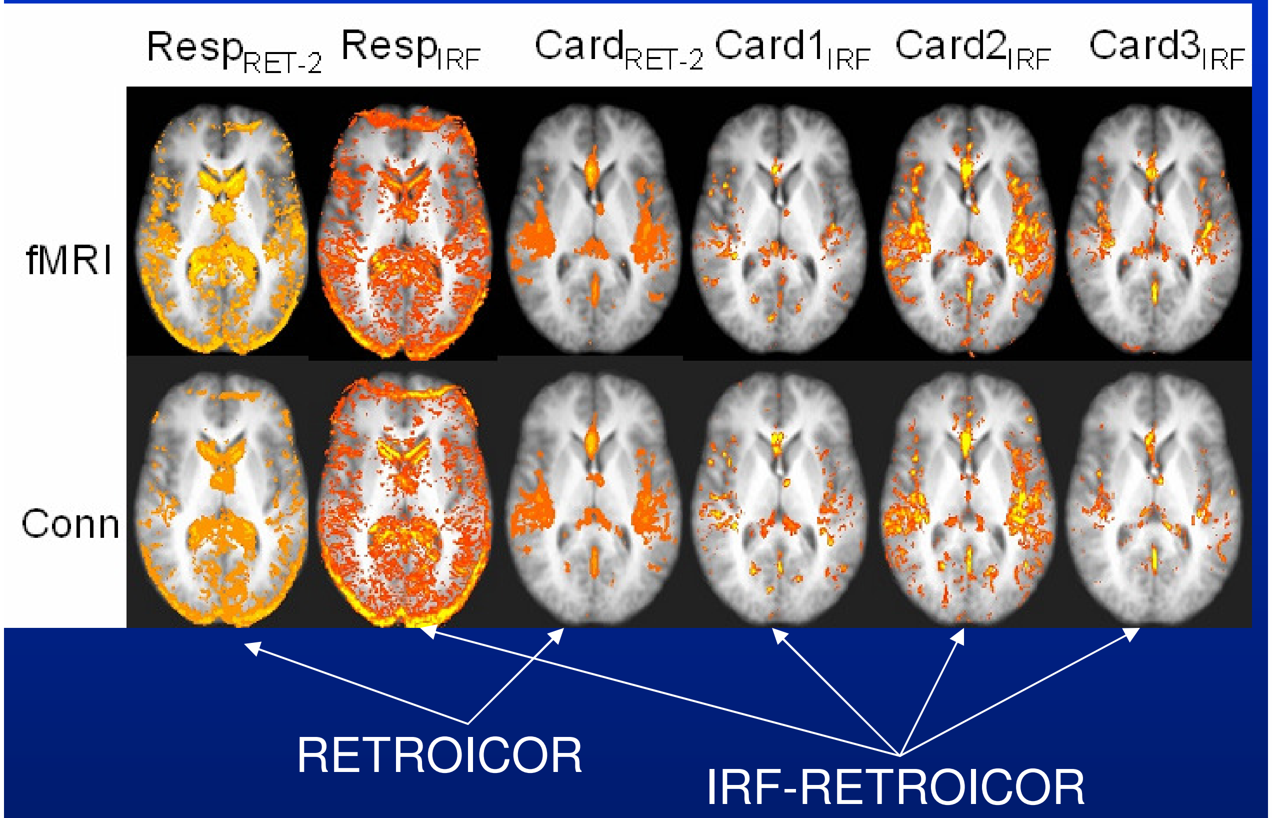
Prior information: PESTICA maps

OR, use pulse and respiration signals

IRFRET shapes

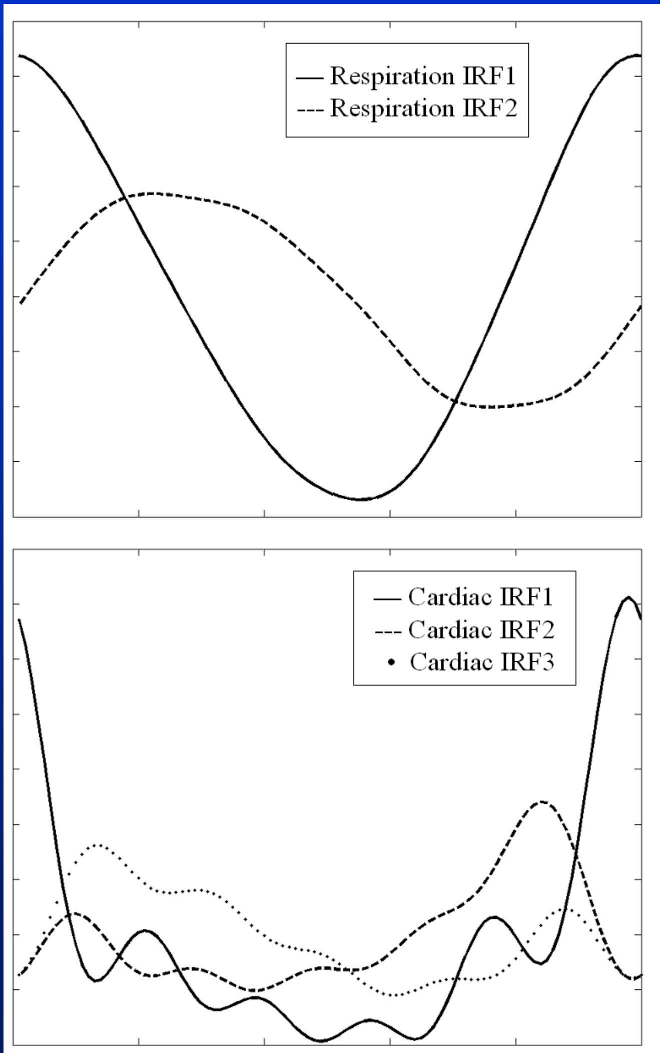
- If you feed in physiologic signals from the wrong subject, or just plain noise, you get arbitrary/inconsistent IRF shapes and totally wrong coupling maps
  - Therefore, the consistency of IRF shapes/maps is useful for validation
- Number of voxels significantly coupled also useful as validation
  - However, both of these are dependent on acquisition and may change with protocol changes

First, IRFs and coupling maps

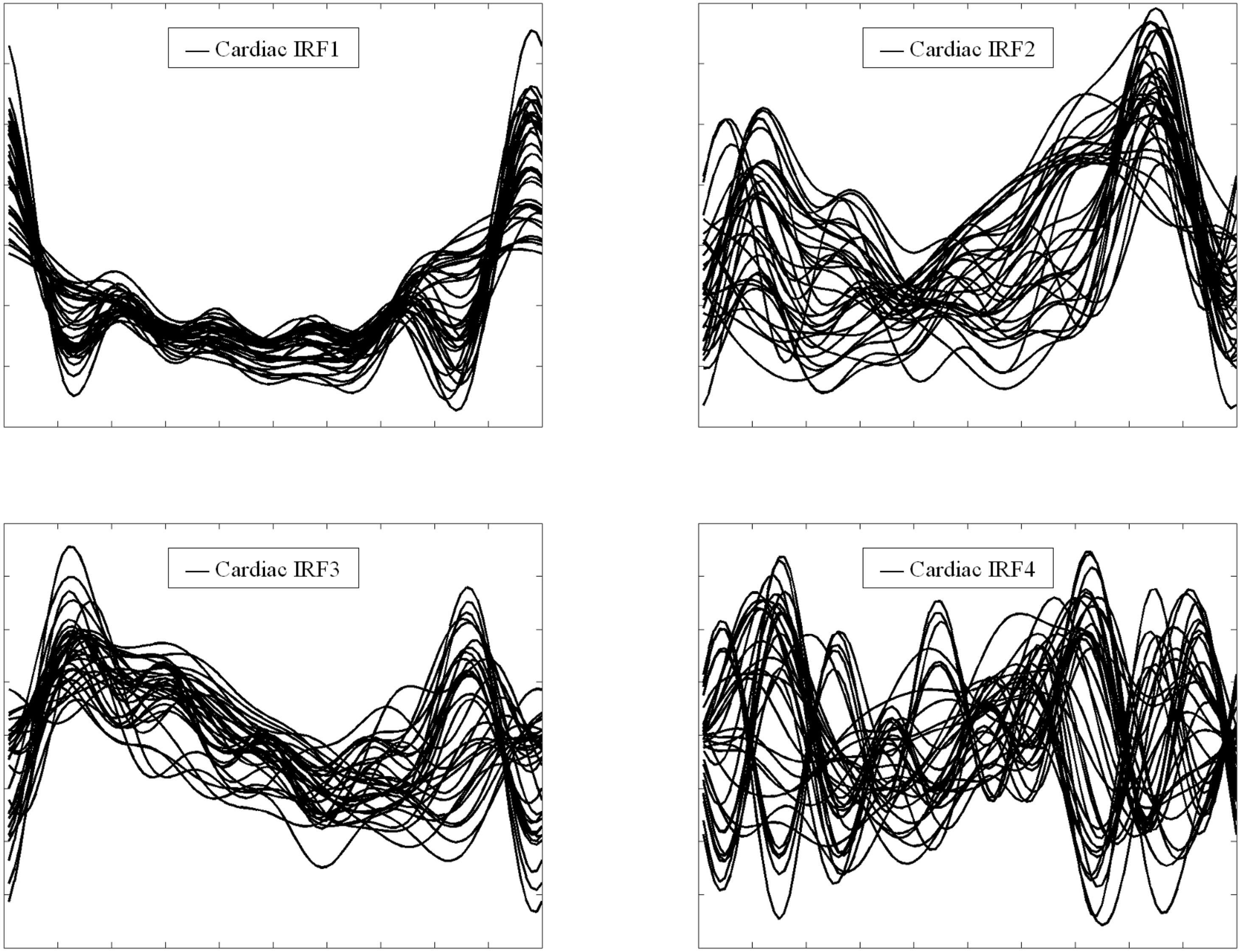


RETROICOR

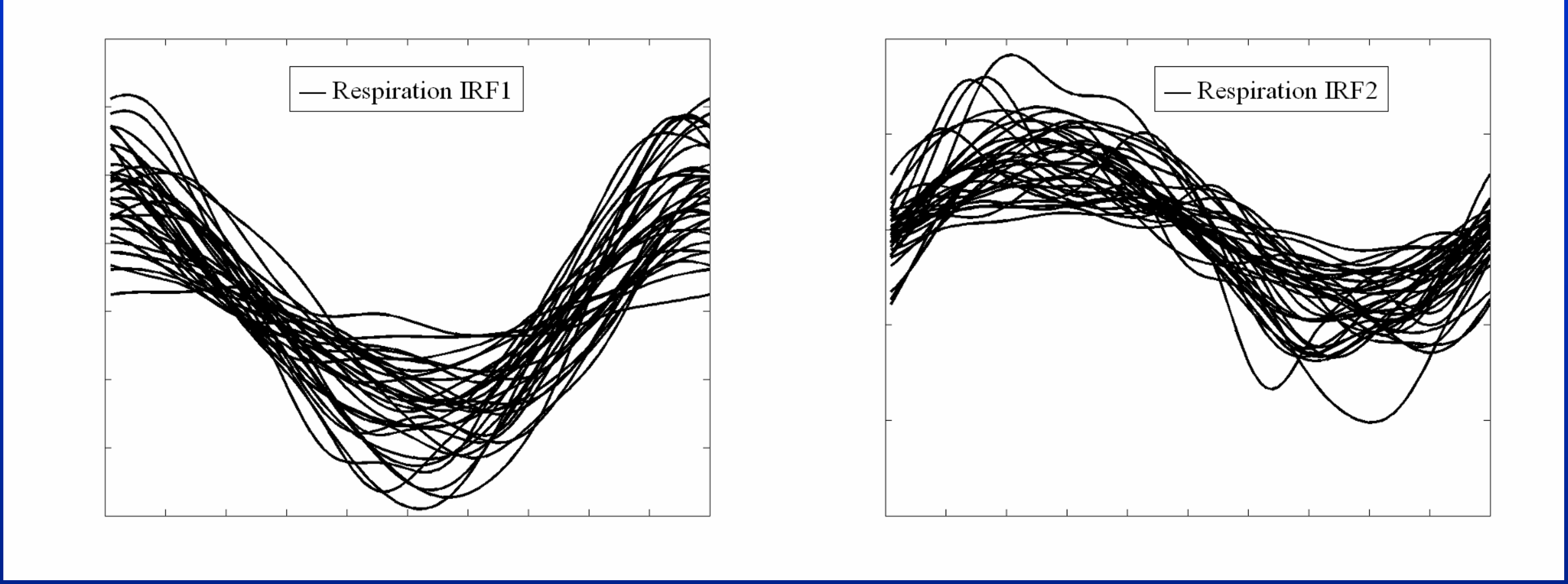
IRF-RETROICOR



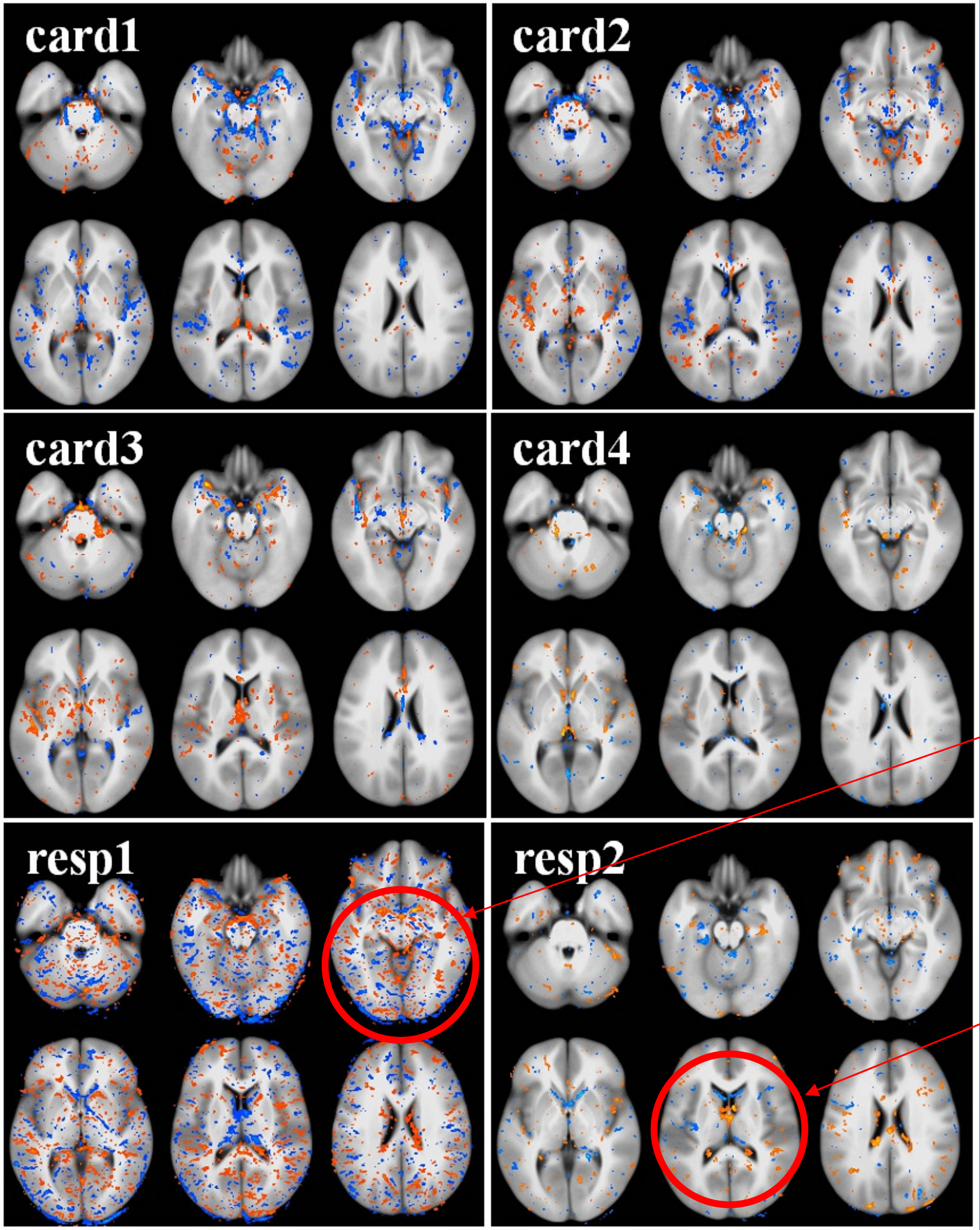
Multiple subject cardiac IRFs:



Respiratory IRFS



- Striking similarity across 34 independent subjects (same results in both block-paradigm motor fMRI and in resting connectivity scans) for first 3 cardiac and first 2 respiratory IRFS, all data analyzed separately.
- When random phase is input to algorithm, we get garbage out. Very little coupling and arbitrary IRFs.



Physical consequences

- No clear physical picture for cardiac, but respiration may make sense
- First IRF is chest expansion-induced field offset, second IRF may be CSF flow secondary to respiratory motion of brainstem

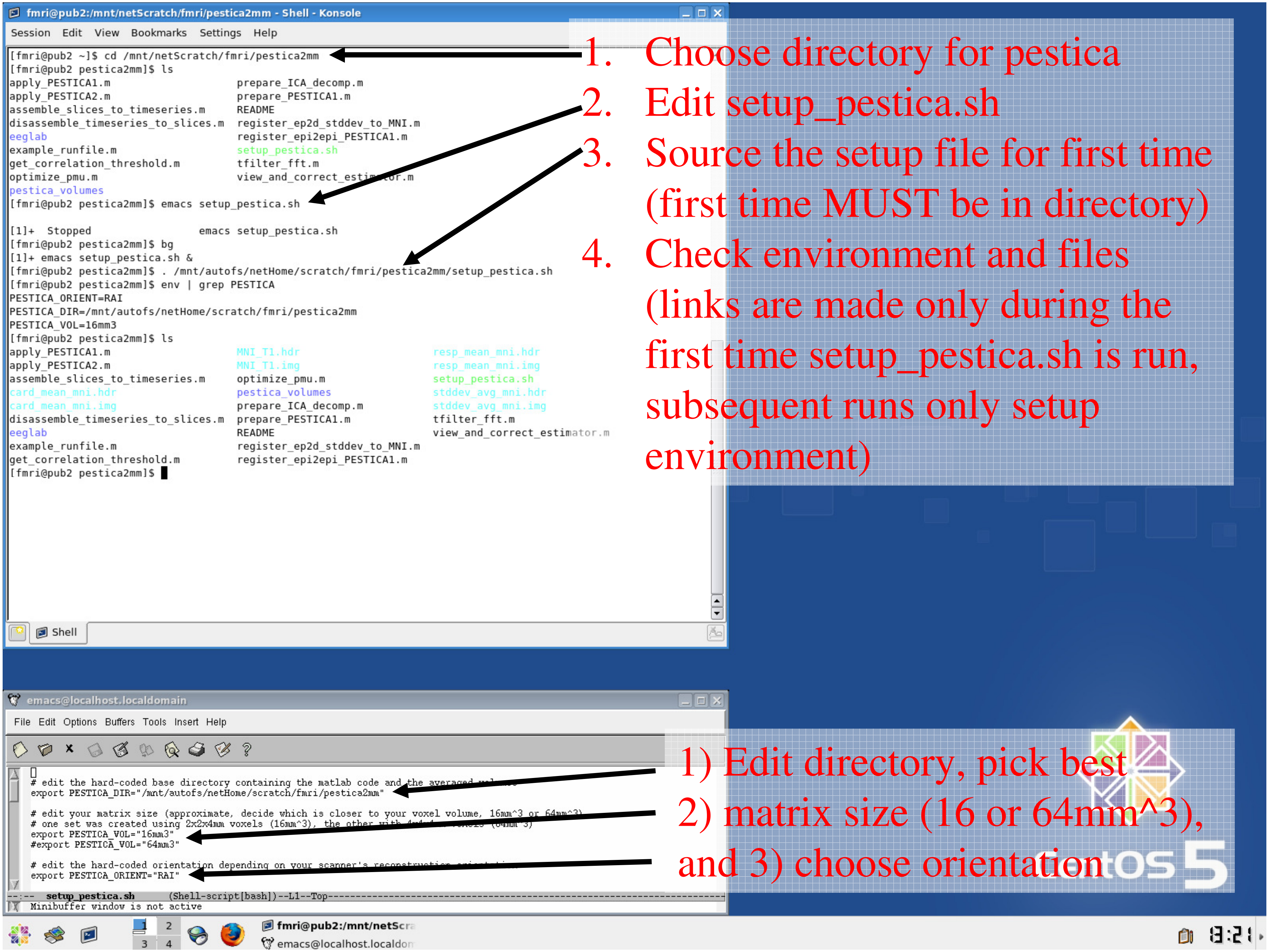
Now, how to run them…

- Prep: setup PESTICA for your data (once)

- Step 1: run temporal ICA on every slice
  - *run_pestica_ica.sh <datafile> <maskfile>*
- Step 2: run PESTICA estimation algorithm
  - *run_pestica_est.sh <datafile> <maskfile>*
- Step 3: filter estimators (interactive)
  - *filter_pestica_est.sh <zdim> <TR>*
- Step 4: run IRF-RETROICOR (two-pass)
  - *run_irfret.sh <datafile> <maskfile>*
- Step 5: run QA script (in matlab cmd line):
  - *>> physio_qa(TR,zdim);*

Preparation of PESTICA

- Un-tar PESTICA distribution
  - *tar xvfz pestica.tar.gz*
  - *cd pestica*
- Edit *bash* script inside PESTICA distribution
  - Edit *setup_pestica.sh*, choose 1) data directory, 2) voxel volume and 3) orientation (contact me if you use *tcsh* or another shell)
- Run setup ONLY once from pestica directory
  - *./setup_pestica.sh*
    - Sets up links to the appropriate orientation geometry, check with viewer against your data, creates a file ".firsttime" so it doesn't relink files during future runs of this setup script.
      - » If you later need to setup with different orientation, remove .firsttime and re-run again from inside pestica directory
- Put setup into your startup script
  - Edit .bashrc or similar startup script in your home dir, add:
  - *source /home/user/pestica_directory/setup_pestica.sh*
    - Or run *source* line manually every time you need PESTICA



fmri@pub2:/mnt/netScratch/fmri/pestica2mm - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
[fmri@pub2 ~]$ cd /mnt/netScratch/fmri/pestica2mm
[fmri@pub2 pestica2mm]$ ls
apply_PESTICA1.m                    prepare_ICA_decomp.m
apply_PESTICA2.m                    prepare_PESTICA1.m
assemble_slices_to_timeseries.m     README
disassemble_timeseries_to_slices.m  register_ep2d_stddev_to_MNI.m
eeglab                              register_epi2epi_PESTICA1.m
example_runfile.m                   setup_pestica.sh
get_correlation_threshold.m         tfilter_fft.m
optimize_pmu.m                      view_and_correct_estimator.m
pestica_volumes
[fmri@pub2 pestica2mm]$ emacs setup_pestica.sh

[1]+  Stopped                 emacs setup_pestica.sh
[fmri@pub2 pestica2mm]$ bg
[1]+ emacs setup_pestica.sh &
[fmri@pub2 pestica2mm]$ . /mnt/autofs/netHome/scratch/fmri/pestica2mm/setup_pestica.sh
[fmri@pub2 pestica2mm]$ env | grep PESTICA
PESTICA_ORIENT=RAI
PESTICA_DIR=/mnt/autofs/netHome/scratch/fmri/pestica2mm
PESTICA_VOL=16mm3
[fmri@pub2 pestica2mm]$ ls
apply_PESTICA1.m                    MNI_T1.hdr                    resp_mean_mni.hdr
apply_PESTICA2.m                    MNI_T1.img                    resp_mean_mni.img
assemble_slices_to_timeseries.m     optimize_pmu.m                setup_pestica.sh
card_mean_mni.hdr                   pestica_volumes               stddev_avg_mni.hdr
card_mean_mni.img                   prepare_ICA_decomp.m          stddev_avg_mni.img
disassemble_timeseries_to_slices.m  prepare_PESTICA1.m            tfilter_fft.m
eeglab                              README                        view_and_correct_estimator.m
example_runfile.m                   register_ep2d_stddev_to_MNI.m
get_correlation_threshold.m         register_epi2epi_PESTICA1.m
[fmri@pub2 pestica2mm]$ █
```

1. Choose directory for pestica
2. Edit setup_pestica.sh
3. Source the setup file for first time (first time MUST be in directory)
4. Check environment and files (links are made only during the first time setup_pestica.sh is run, subsequent runs only setup environment)

emacs@localhost.localdomain
File Edit Options Buffers Tools Insert Help

```
# edit the hard-coded base directory containing the matlab code and the averaged volumes
export PESTICA_DIR="/mnt/autofs/netHome/scratch/fmri/pestica2mm"

# edit your matrix size (approximate, decide which is closer to your voxel volume, 16mm^3 or 64mm^3)
# one set was created using 2x2x4mm voxels (16mm^3), the other with 4x4x4mm voxels (64mm^3)
export PESTICA_VOL="16mm3"
#export PESTICA_VOL="64mm3"

# edit the hard-coded orientation depending on your scanner's reconstruction orientation
export PESTICA_ORIENT="RAI"
--:--  setup_pestica.sh      (Shell-script[bash])--L1--Top------------------------------------
Minibuffer window is not active
```

1) Edit directory, pick best 2) matrix size (16 or 64mm^3), and 3) choose orientation

Data setup

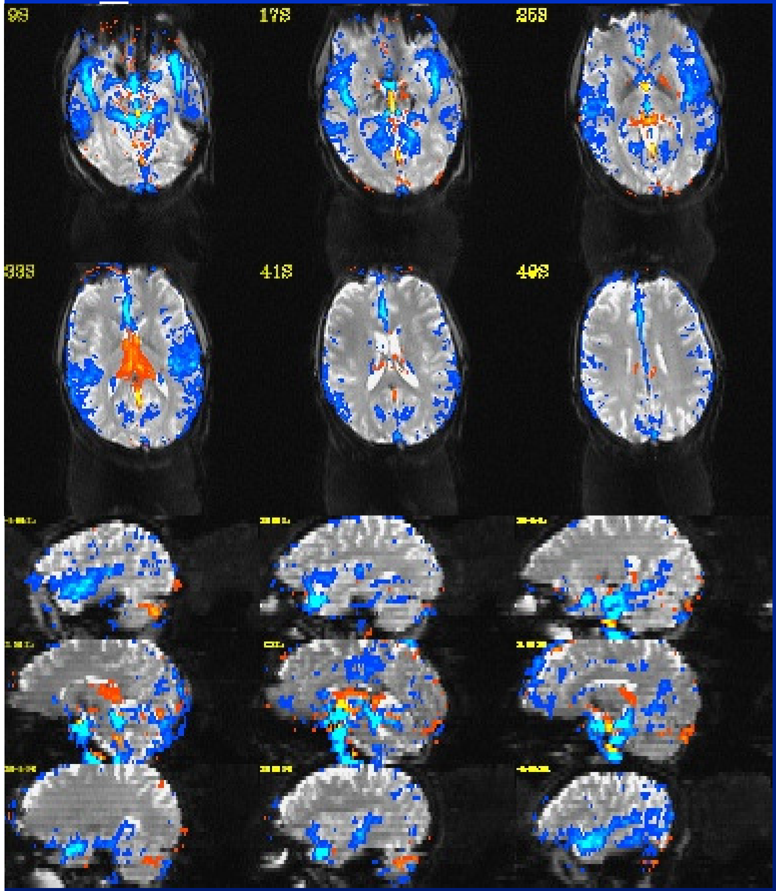
- Data must be in one of: ANALYZE, NIFTI, NIFTI_PAIR, NIFTI_GZ or AFNI BRIK
  - Make sure EPI data is in 3D+time format
  - Note, PESTICA uses the AFNI_matlab distribution for reading/writing
- Create a mask for your EPI
  - 3dSkullStrip -input <data> -prefix <mask>
- Check mask/data with AFNI viewer

Use PESTICA on your data…

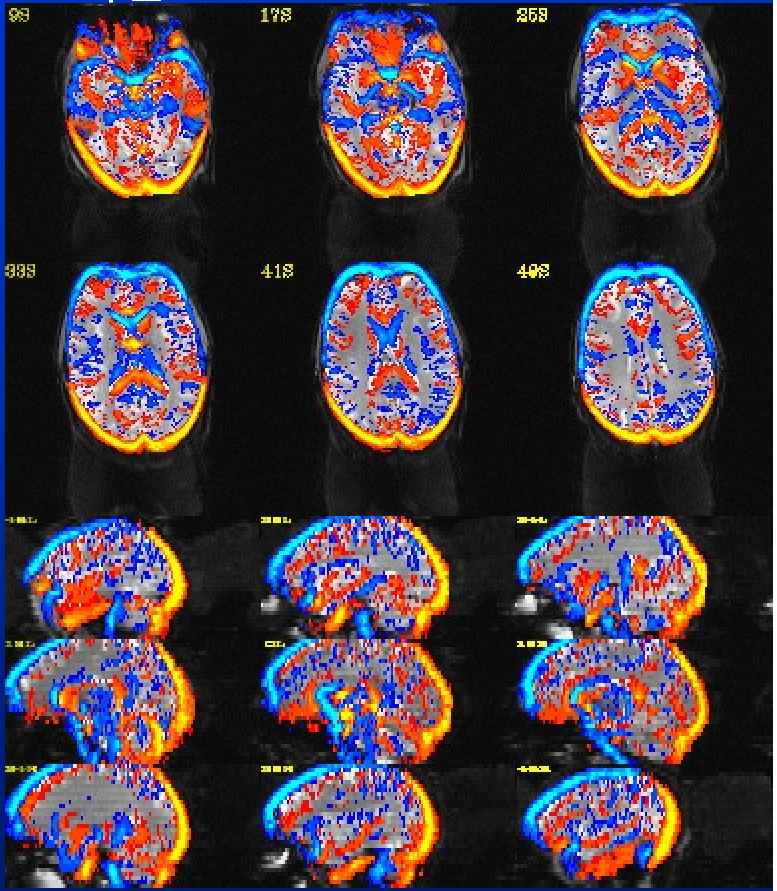
- Go to your data directory, then run_pestica_ica.sh and run_pestica_est.sh
  - Scripts will create a new subdirectory “**pestica**/” and all working data will be stored here for later use
    - warning, deleting this directory will require you to redo the ICA decomposition if you need to rerun steps
  - ICA make take considerable time, will display when done for each slice
  - run_pestica_est.sh script does coregistration to MNI EPI template
    - check this coregistration inside the pestica/ subdir, make sure resp_PESTICA.nii and card_PESTICA.nii line up over your data

Check template coregistration

card_PESTICA.nii overlain on EPI data



resp_PESTICA.nii overlain



Note cardiac coupling lines up over arteries, ventricles,

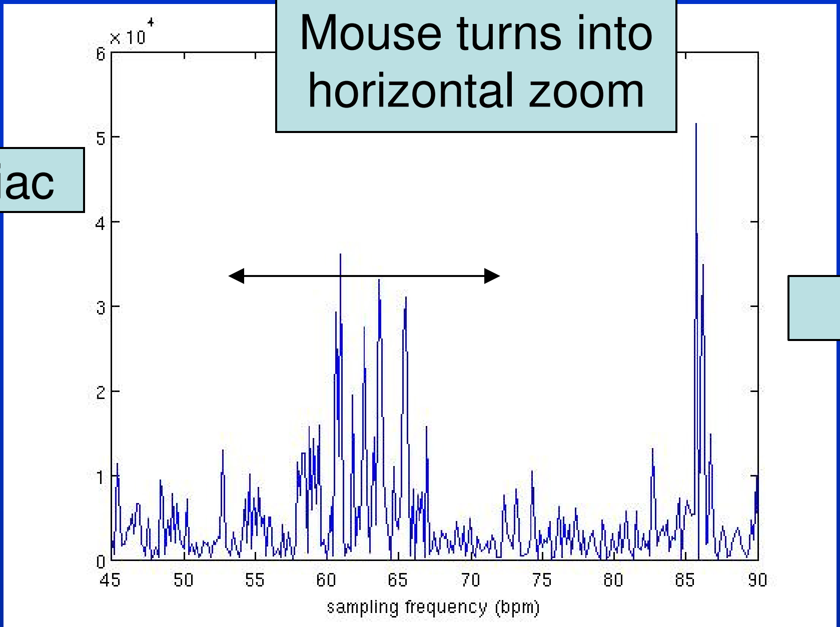
respiration lines up over anterior/posterior edges of brain structures

Use “afni –dset pestica/*.nii” and it will pick up files in current directory as well, set overlay threshold to 5, and turn off autoRange and set range to 30. For a nicer colorscale, set color panes to 12 (instead of **) and you should get similar visual appearance as shown here. If coreg doesn’t work, it will be obvious.

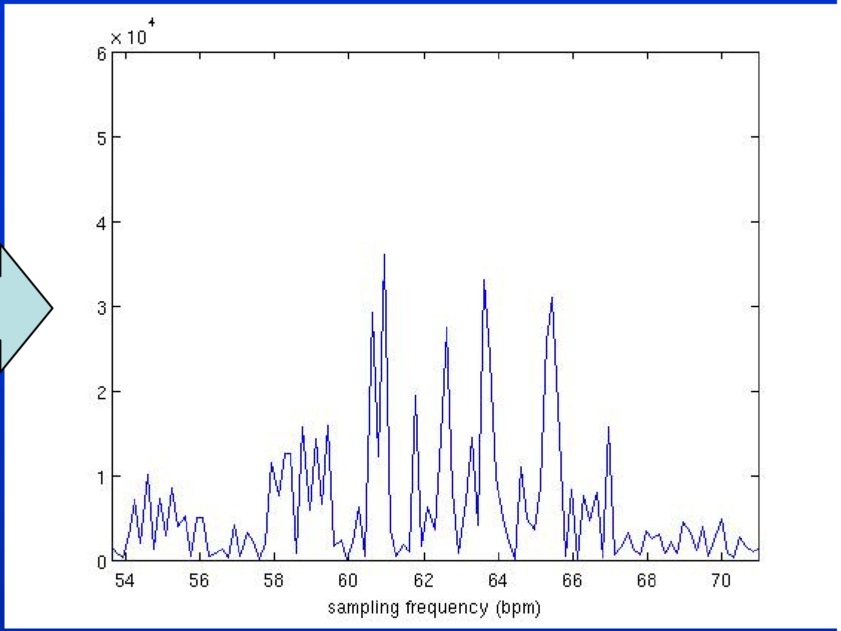
Apply temporal filter to estimators

- You have to enter your number of slices and repetition time (in either msec or sec), as this will temporally filter the estimator based on your selection
  - *filter_pestica_est.sh <zdim> <TR>*
- Creates MATLAB plot showing Fourier transform of cardiac estimator, zoomed to appropriate region
  - 45-90bpm initially, can select further, look for bunch of peaks in this range. In a large set of subjects, we found the cardiac peak freq to range from 48 to 85bpm with 5-10bpm width per subject.
  - Zoom with mouse over the peaks you suspect are cardiac.
  - If you don’t have much confidence, feel free to leave zoom region large (already zoomed to 45-90bpm, which is a reasonable filter)
  - Hit enter in MATLAB command line, move to next (respiration)
  - Respiration freq between 10 and 24bpm, but typically closer to 17bpm.

Cardiac

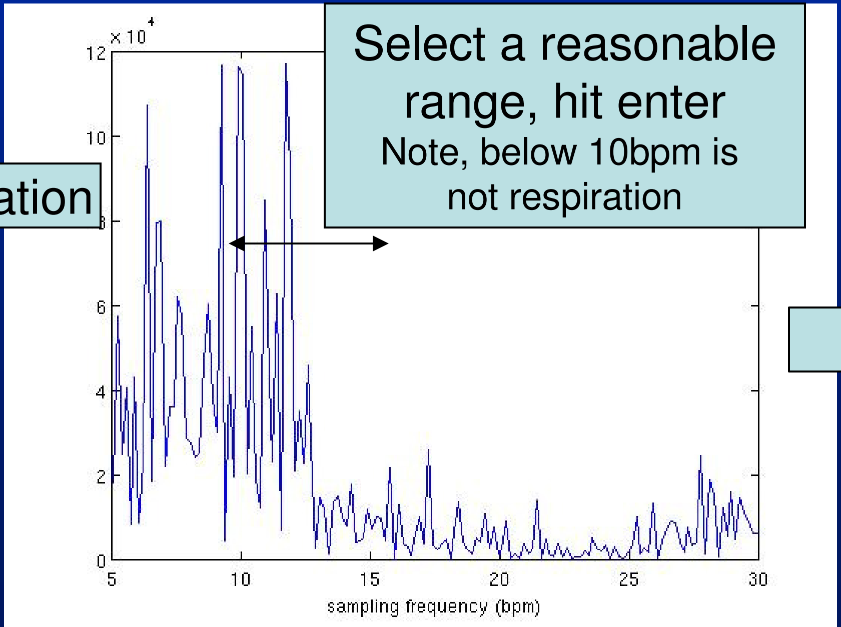


Mouse turns into horizontal zoom



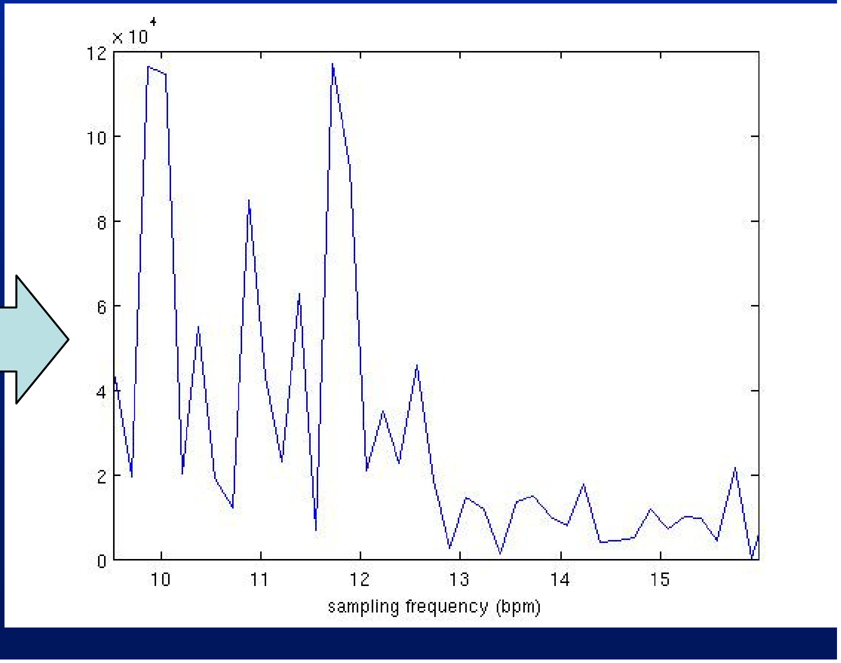
If you are ever in doubt, go wider!

Respiration



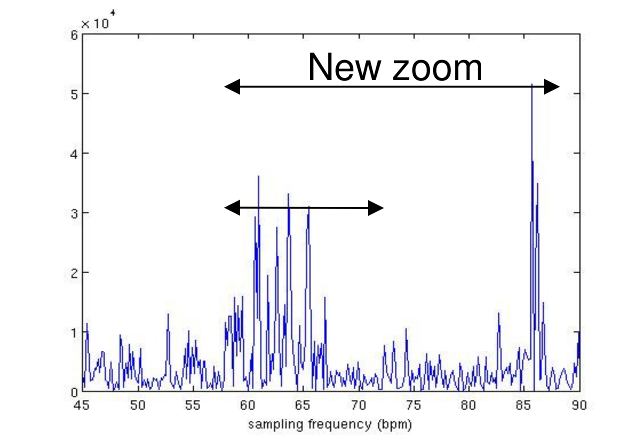
Select a reasonable range, hit enter

Note, below 10bpm is not respiration



My notes on frequency selection

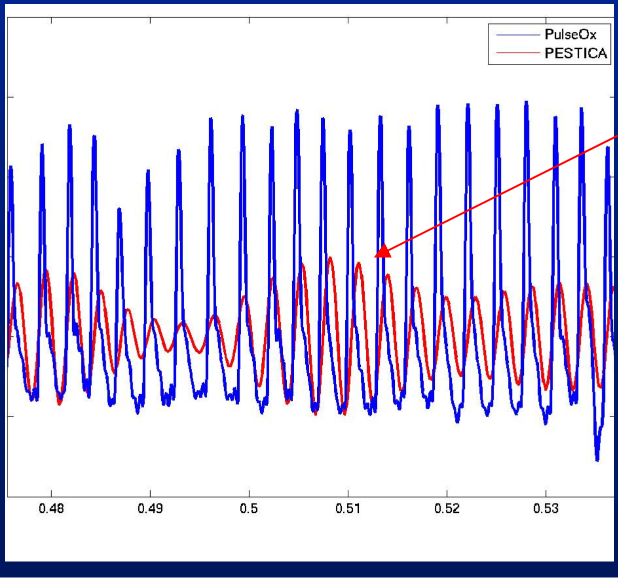
- In pulse monitoring data, we never saw anyone outside the 48-85 and 10-24bpm ranges – plot is wider than this intentionally
- Look for “chunk” of peaks that’s about 5-10bpm wide – its rarely but occasionally narrower than this, sometimes wider though…



In cardiac plot on previous page, I would include the peak at 87bpm and try running IRFRET to see if I get a better result. In this subject, it did not improve things, leading me to conclude that the 87bpm peak is not cardiac. You can always re-run filter, and then run_irfret.sh again (caveat, IRFRET takes 1-4 hours).

IRFRET

- run_irfret.sh does a second-pass of IRFRET after cleaning up the cardiac estimator. This cleaning process takes the most time – it is adjusting the dither of each cardiac peak to optimize it.
  - Temporal filtering adds dither in peak, which is substantial for cardiac. This fixes it, and results in output that is closer matched to a pulse oximeter



Note dither in peak locations – sometimes the peaks line up, sometimes they are shifted slightly, which will hurt fitting.

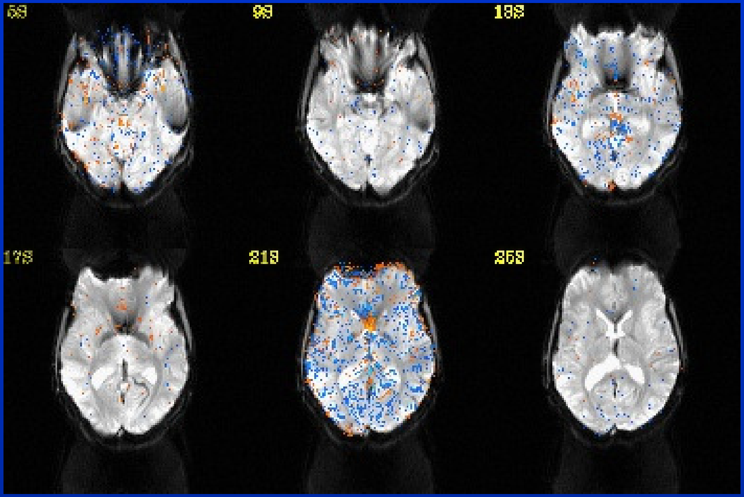
This dither is due to harmonics in cardiac process that filtering removed and thermal noise added in to signal. Cleanup script fixes this – validated against pulse ox data.

QA

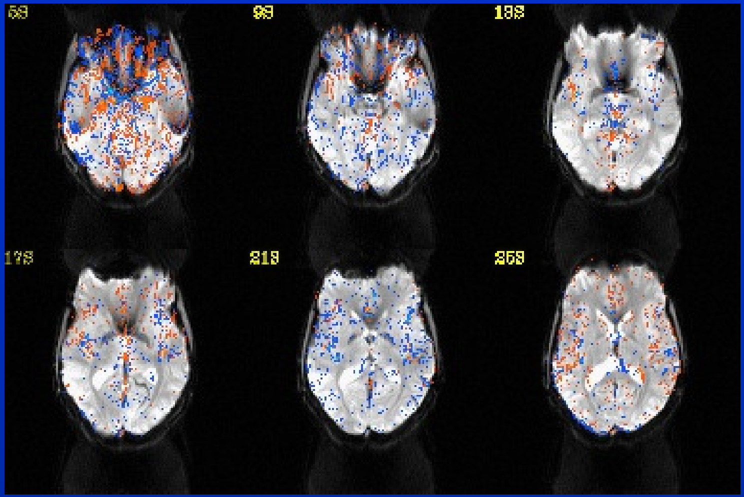
- physio_qa(TR,zdim)
- MATLAB figures showing histogram of breaths/heartbeats per minute, and IRFs
  - IRFs are data acquisition-specific
- MATLAB command line notes number of voxels significantly coupled to heart and breathing processes
  - Also acq-specific, but >2000 voxels seems good for our acquisition
  - We will try to build a database of voxel count cutoffs for QA purposes for several different acquisitions
- Can also look at coupling maps
  - Use afni and load coupling_irfret_card and resp
  - Overlay on EPI and look for arteries/ventricles in cardiac, anterior/posterior edges in respiration…
  - Images are not very cleanly indicative of physiologic noisy regions, but this is typical, and these are the same as what you'd get with pulse ox/respiration belt. Mostly just make sure you're not getting flat or weird maps.

Cardiac coupling maps

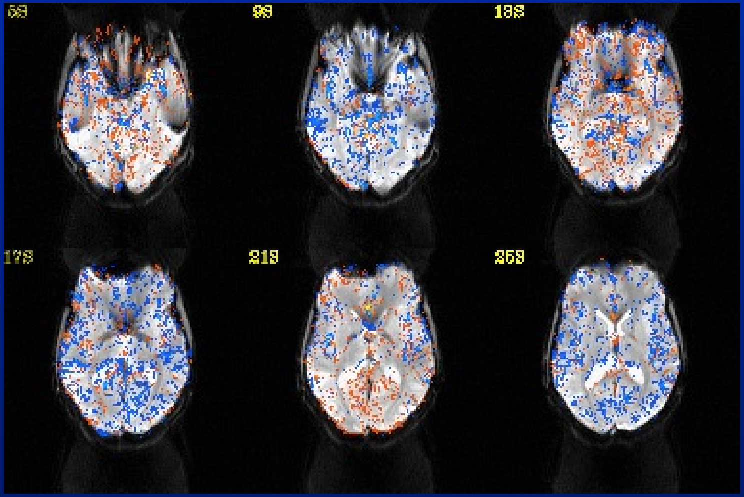
IRF1



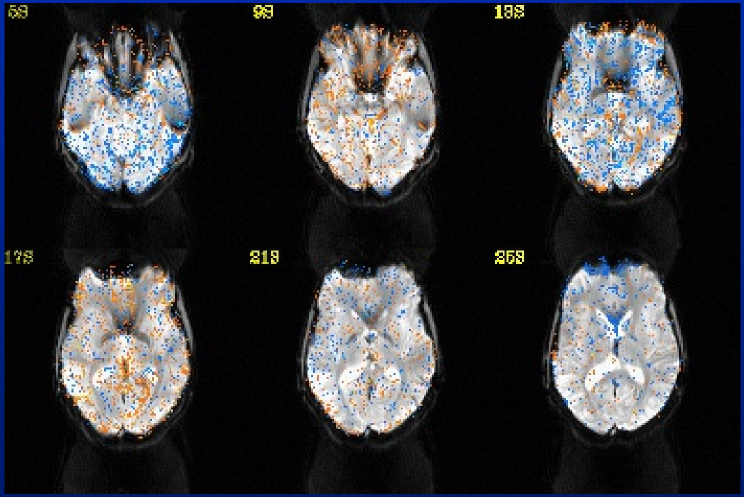
IRF3



IRF2

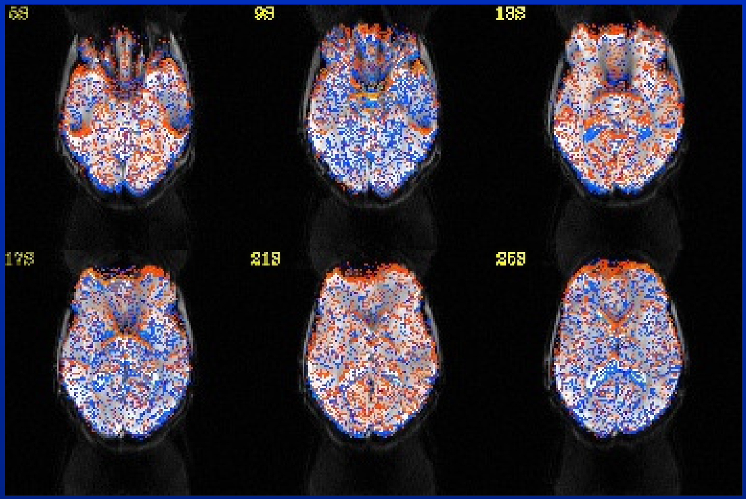


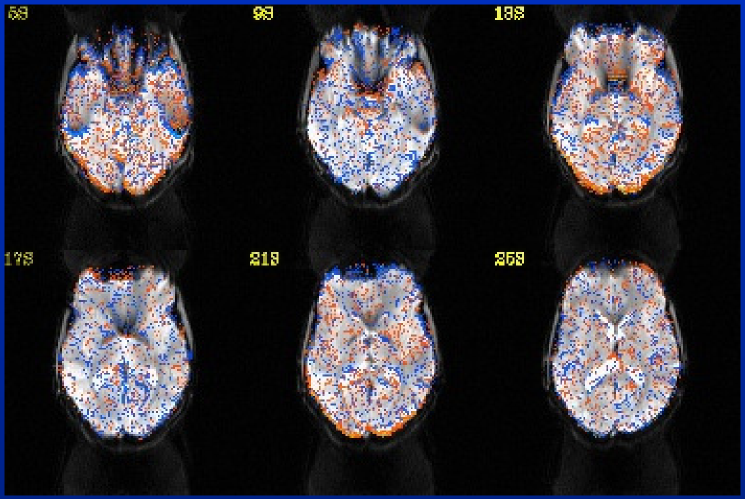
IRF4



Note artery in front of corpus callosum in some IRFs, ventricles, CSF spaces and some MCA arteries. Also note strong every other slice – due to slice acquisition being interleaved in this case. Your data may or may not have this. These are thresholded manually – play with settings to get a feel for where coupling is.

Respiratory coupling





Note anterior/posterior edges of structures. This is because phase-encoding direction is anterior/posterior. This is stronger on Siemens Trios and Verios than Allegras and several non-Siemens scanners. The respiratory artifact is primarily a small field shift that is mostly seen as image shifting in the phase-encode direction by a portion of a voxel. Hence the coupling shows up along image intensity boundaries perpendicular to the A/P axis.