ETH zürich

University of Zurich[UZH]

Translational Neuromodeling Unit

# fMRI Preprocessing & Noise Modeling

## *An SPM Tutorial*

Lars Kasper

September 25th/ October 17th, 2015

MR-Technology Group & Translational Neuromodeling Unit

Institute for Biomedical Engineering

University of Zurich and ETH Zurich

# Tutorial: Code and Data

- Download example code and data presented in this talk:
  - http://www.tnu-zurich.org/team/lars-kasper/
  - Section: Talk and Lecture Materials
- SPM12 (Statistical Parametric Mapping)
  - http://www.fil.ion.ucl.ac.uk/spm/software/spm12/
  - developed by the Functional Imaging Lab, UCL, London
- TAPAS PhysIO Toolbox (SPM or Matlab standalone)
  - http://www.translationalneuromodeling.org/tapas/
  - Documentation & Example Data (Philips/Siemens/GE):
    - http://www.translationalneuromodeling.org/software/documentation/
    - http://www.translationalneuromodeling.org/software/tapas-data/

# Outline

- Setting up a Preprocessing Pipeline in SPM: The Batch Editor

Setup Pipeline
DEMO 1

- Monitoring and Comparing Pipelines: Quality Assurance

Quality Monitoring
DEMO 2

- Multi-subject Pipelines: Automatisation

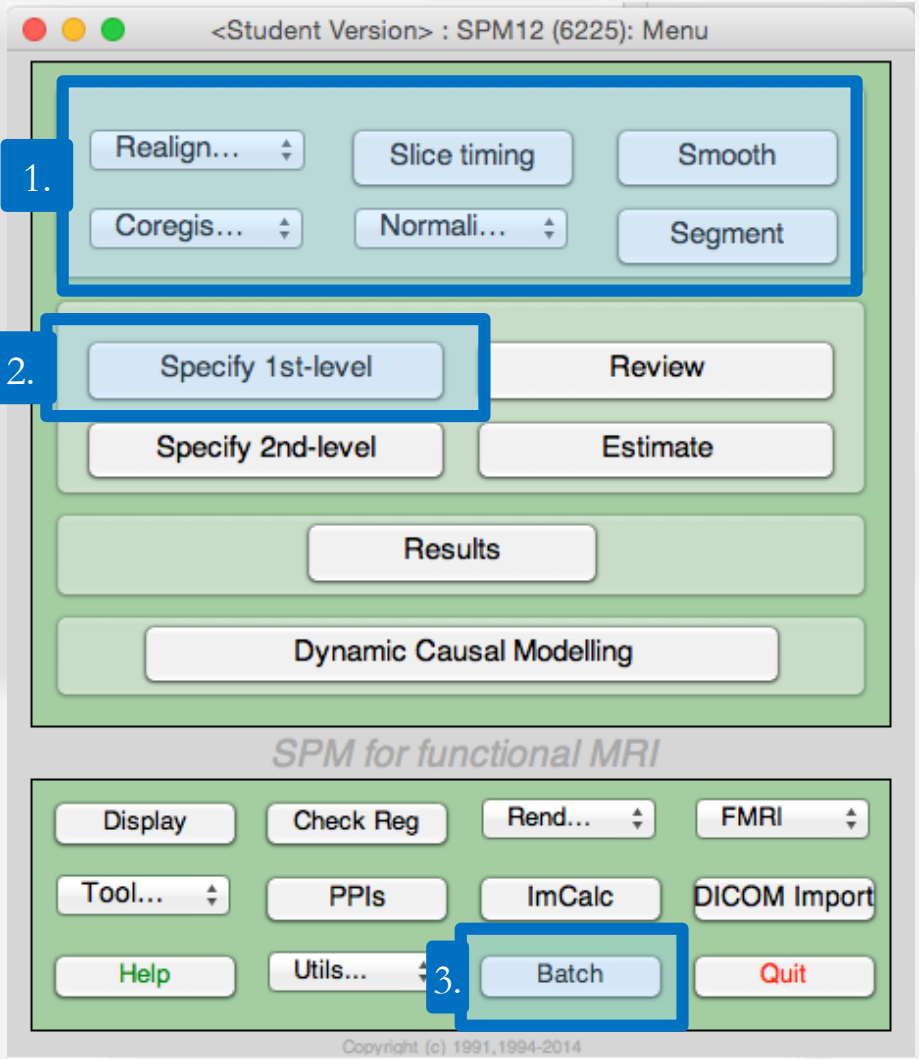
Multi-Subject
DEMO 3

- Integrating Own Code and Toolboxes: Physiological Noise Modeling and Evaluation

Noise Modeling
DEMO 4

# The SPM GUI



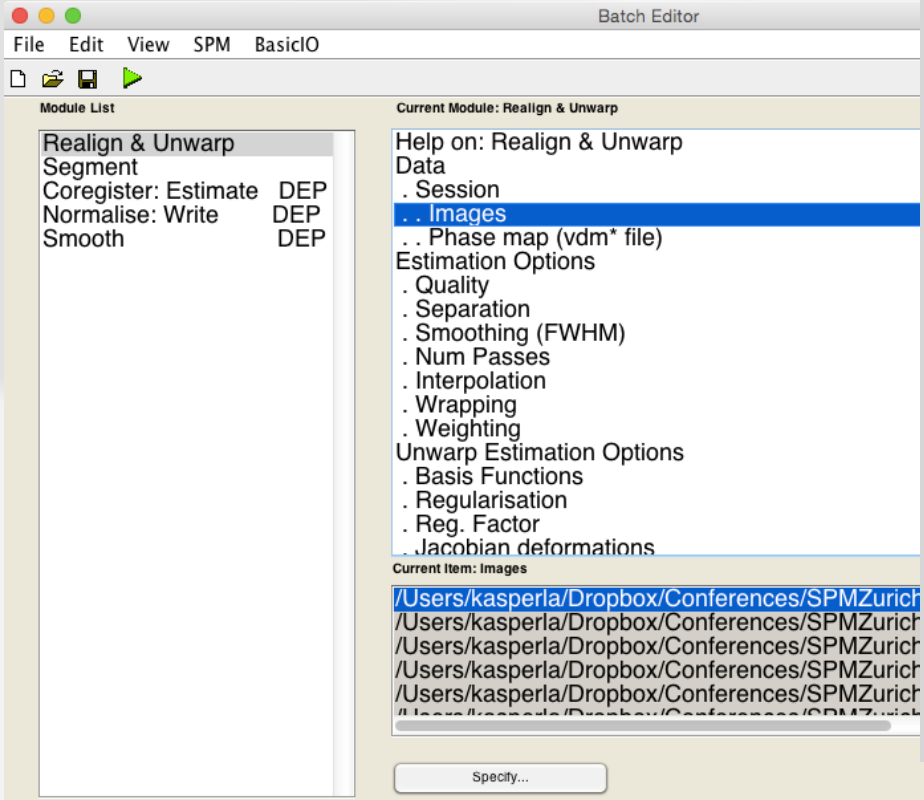


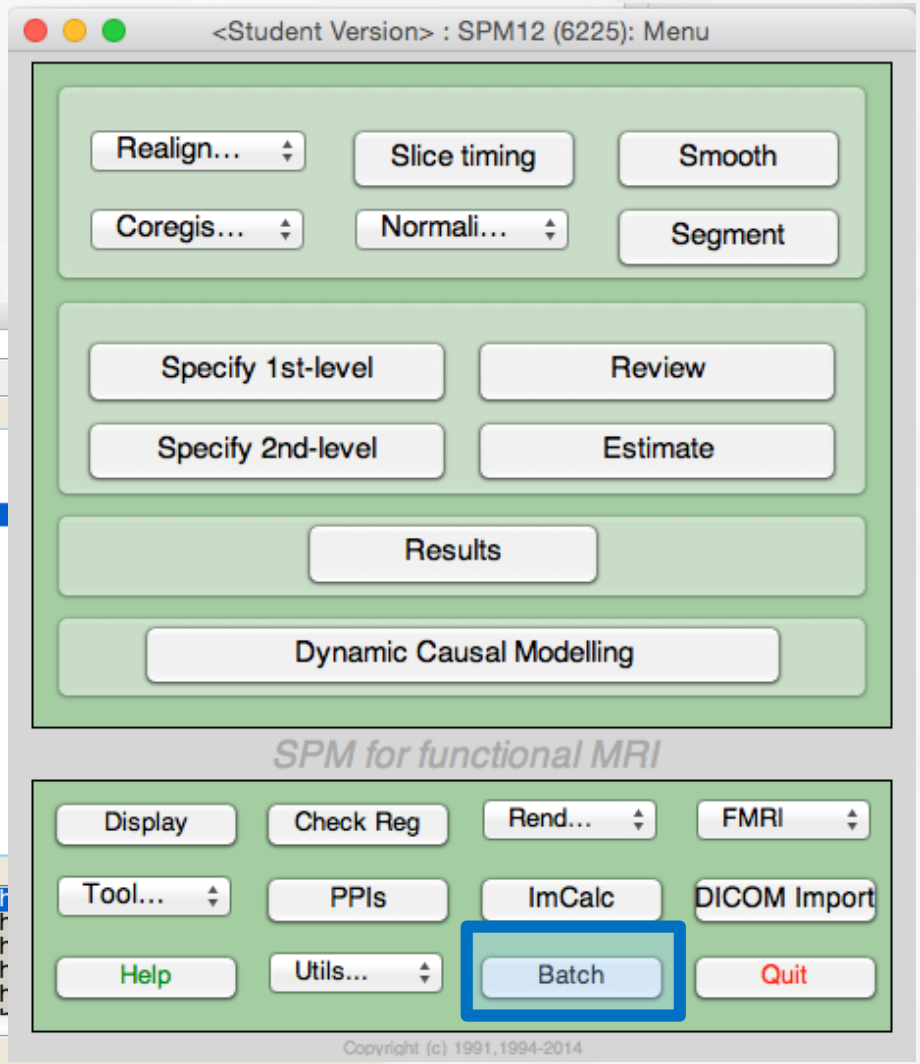

- Preprocessing
  - Realignment
  - Slice-Timing Correction
  - Co-Registration
  - Unified Segmentation & Normalisation
  - Smoothing . . .
- Noise Modeling
  - Physiological Confound Regressors
- The Batch Editor

# The Batch Editor in SPM



- Developer: V. Glauche
- Command line usage: `spm_jobman`



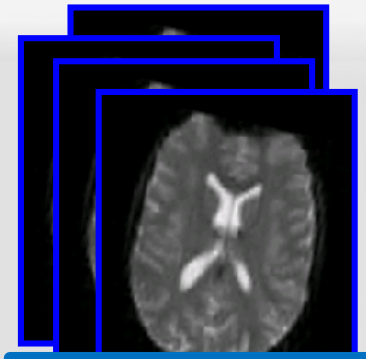



# SPM Preprocessing Pipeline

Input →
Output →

Setup Pipeline | Quality Monitoring | Multi-Sub | No

fMRI time-series

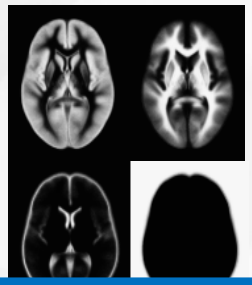


Realignment

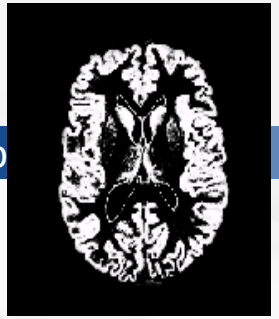
REALIGN

Structural MRI

TPMs



Normalisation: Unified Segmentation

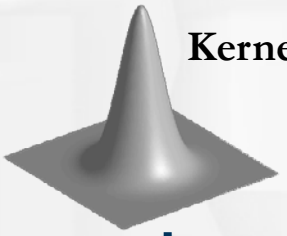


Segmentation

Deformation Fields

(y_struct.nii)

Kernel



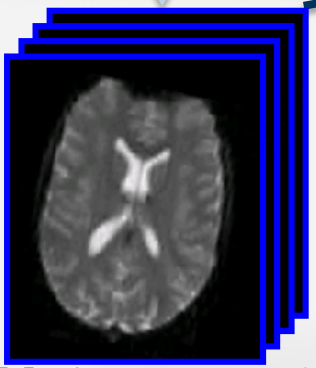
COREG

SEGMENT

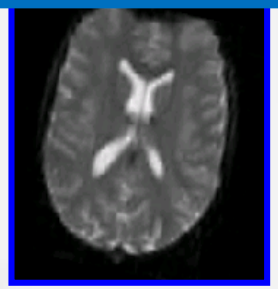
NORM WRITE

SMOOTH

Co-Registration



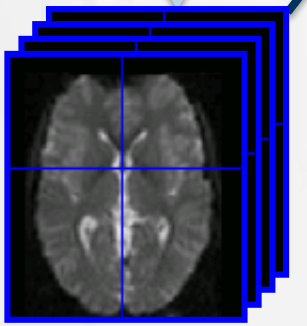
Motion corrected



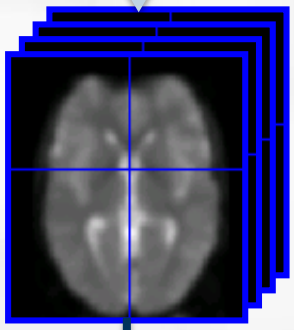
Mean functional

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(Headers changed)



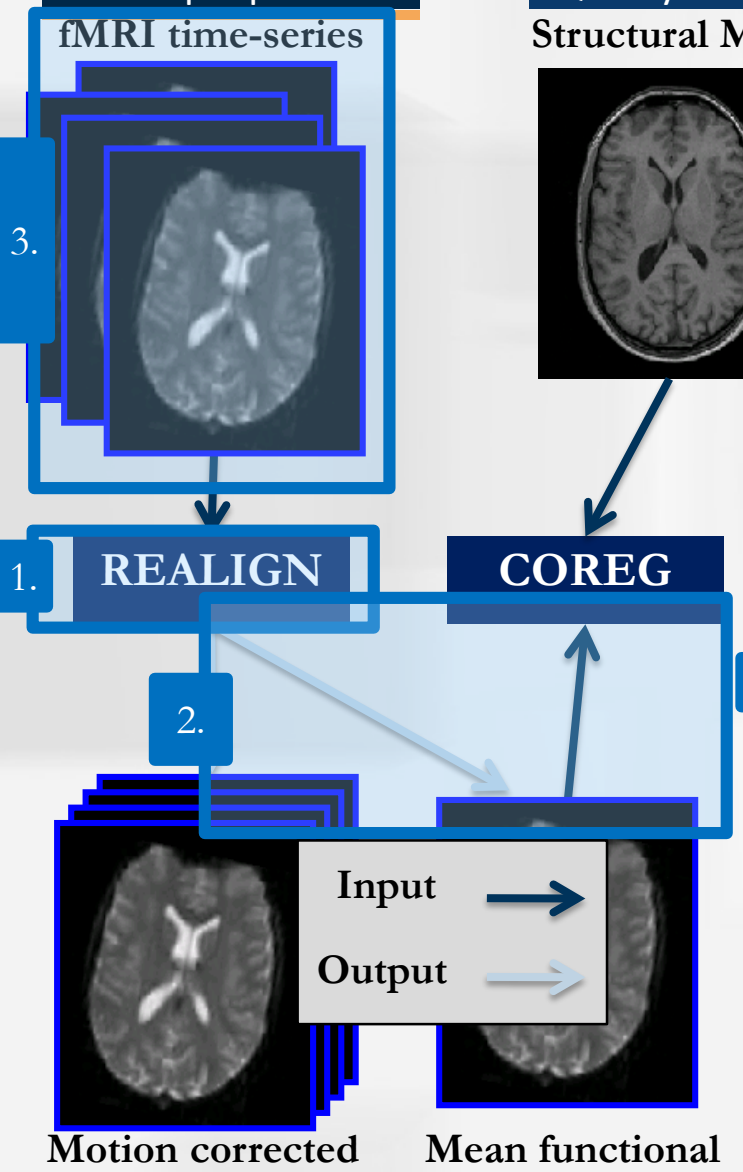
MNI Space



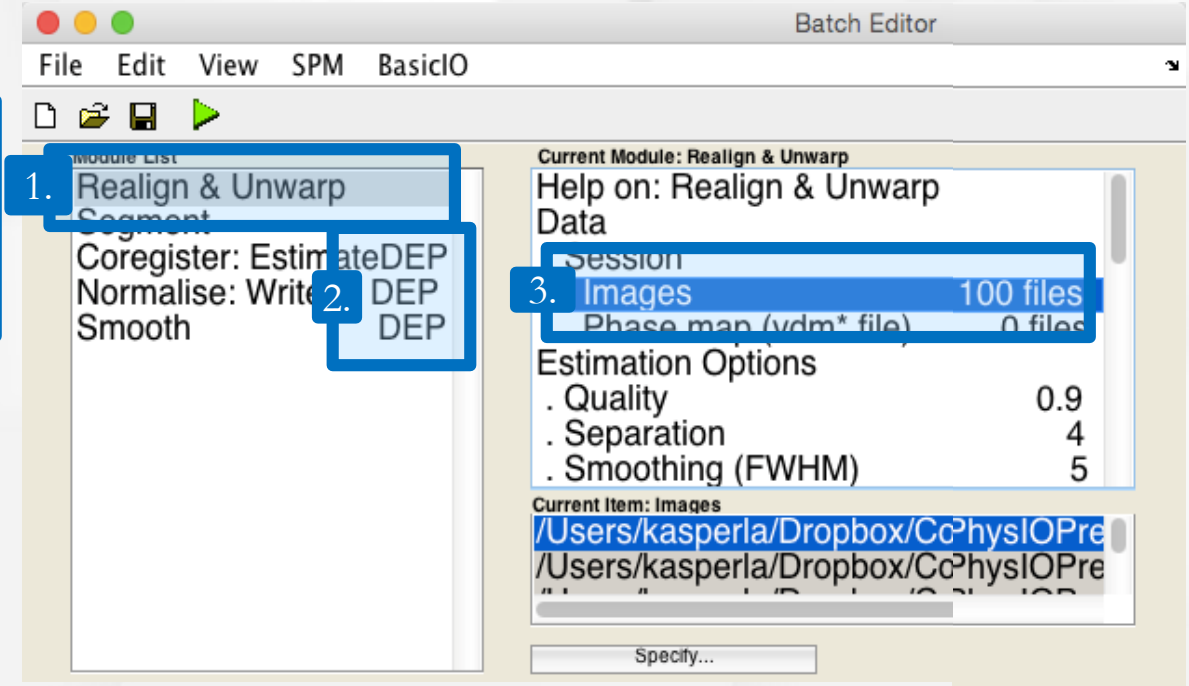
GLM

# From Pipelines to Batches






- Elements of a Pipeline
  1. Modules
  2. Dependencies
  3. Data (Subject-specific)
- Philosophy: Batch = Materialised Pipeline



# Demo 0



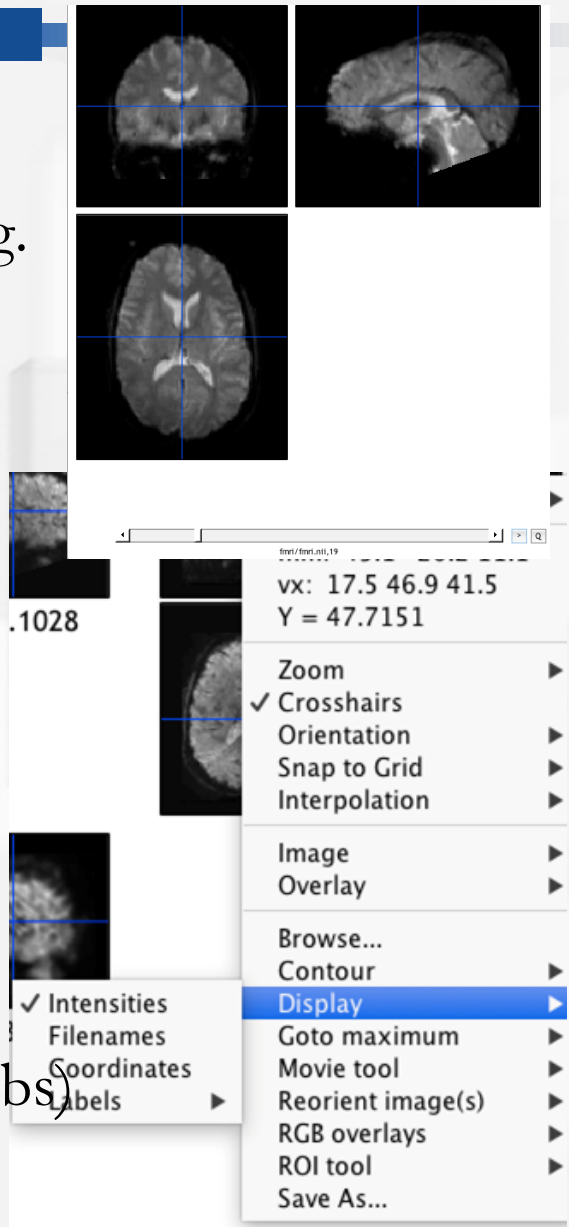
- Setting up the data for all demos & checking it
  - Source: Social Learning (A. Diaconescu[1], TNU Zurich, Philips 3T)
- After download:
  1. Unzip archive `examples_physio_short.zip`
  2. Open Matlab, run `code/init_reset_example.m`
     - subject01-folder created, Batches filled with right data (filenames and path) already
  3. Inspect raw functional *and* structural data carefully
     - Required for each subject
     - Preferred viewer: SPM CheckReg

*[1] Diaconescu (2014), PLoS CB*

# Demo 0: CheckReg-Magic



- Recommended Plotting: Check Reg
  - via Batch Editor: SPM => Util => Check Reg.
  - Matlab command line:
    `spm_check_registration('img1.nii', 'img2.nii', …)`
  - 4D NIFTI files:
    - show individual image tiles:
      `spm_check_registration('fmri.nii,1', 'fmri.nii,2', …)`
    - show movie (**NEW**! SPM12):
      `spm_check_registration('fmri.nii')`
  - **Right click** reveals amazing features (edges, anatomic labels, header info, contrast, add blobs)



# Demo 1



- Simple preprocessing pipeline for fMRI (G. Ridgway)
  - Based on `spm12/batches/preproc_fmri_simplified.m`
- Run via:
  1. Load `subject01/batches/demo01_simple_batch_preproc/batch_spm_preproc_fmri_simplified.m` in Batch Editor
     - Either via GUI or

```
spm_jobman('initcfg');
spm_jobman('interactive',...
'batch_spm_preproc_fmri_simplified.m' );
```

  2. Run Batch (Press Play)
- Exercise: Draw Pipeline Diagram (Modules/Dependencies/Data)

*[1] Diaconescu (2014), PLoS CB*

# Demo 1 – GUI Batch Editor

Batch Editor

File Edit View SPM BasicIO

Module List

Realign & Unwarp
Segment
Coregister: Estimate DEP
Normalise: Write DEP
Smooth DEP

Current Module: Realign & Unwarp

Help on: Realign & Unwarp
Data
. Session
. . Images 100 files
. . Phase map (vdm* file) 0 files
Estimation Options
. Quality 0.9
. Separation 4
. Smoothing (FWHM) 5
. Num Passes Register to first
. Interpolation 2nd Degree B-spline
. Wrapping No wrap
. Weighting 0 files
Unwarp Estimation Options
. Basis Functions 12x12x*
. Regularisation 1
. Reg. Factor Medium
. Jacobian deformations No

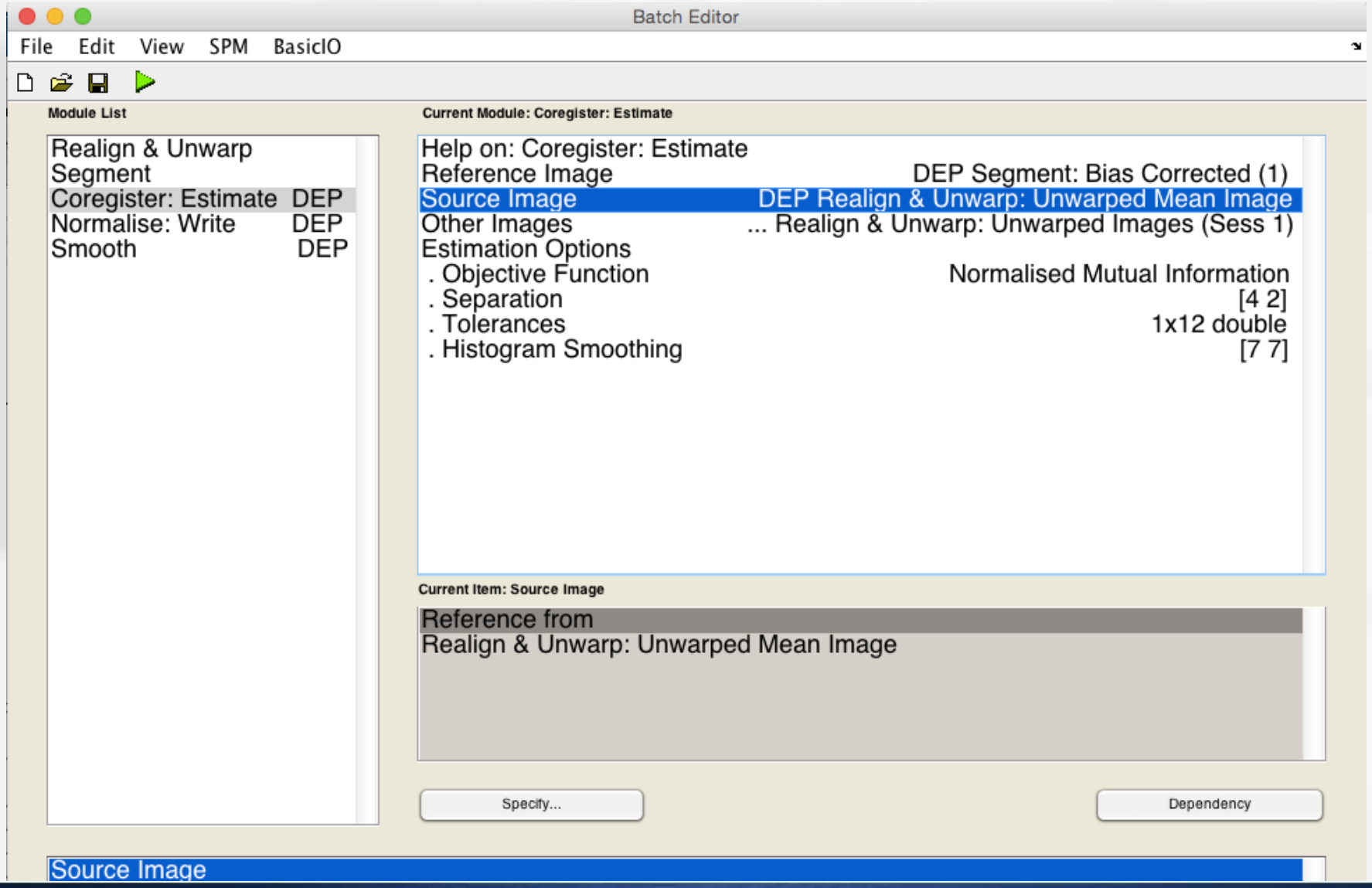
Current Item: Images

/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproce
/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproce
/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproce
/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproce
/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproce

Specify...

# Demo 1 - Dependencies

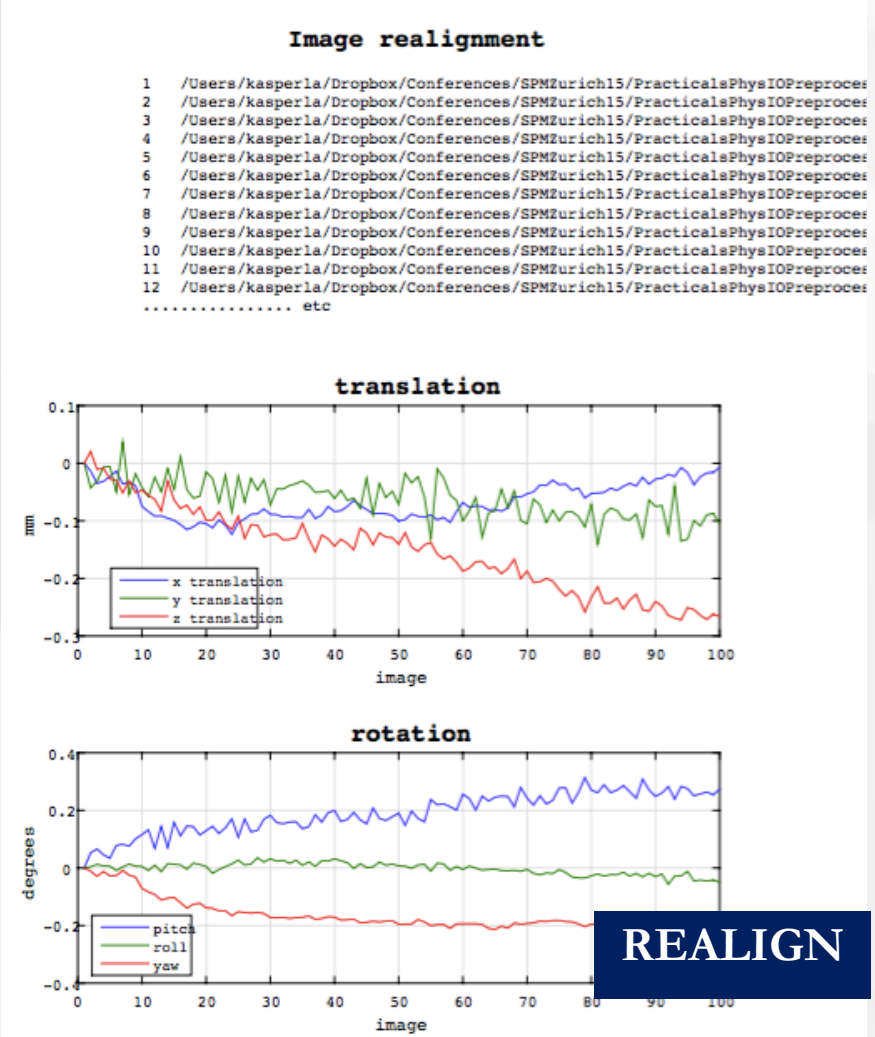




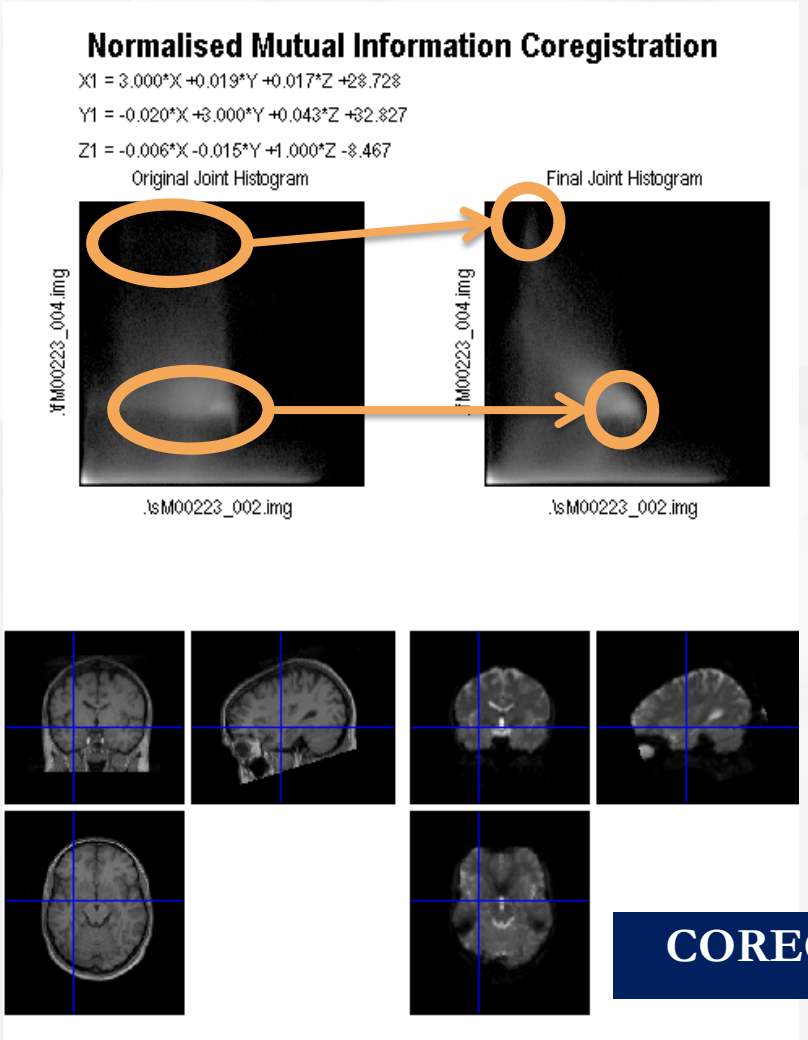
# Demo 1 - Results



- Automatic status plots saved in `spm_<date>.ps`




REALIGN




COREG

# Outline



- Setting up a Preprocessing Pipeline in SPM: The Batch Editor
- **Monitoring and Comparing Pipelines**
  - Performance Measures: Mean/SD/SNR/Diff Image
  - SPM Plotting Routines and Automatic Reporting
  - Spotting Failed Pipelines
  - Comparing Alternative Pipelines
- Multi-subject Pipelines
- Integrating Own Code and Toolboxes: Physiological Noise Modeling and Evaluation

# Pipeline Monitoring



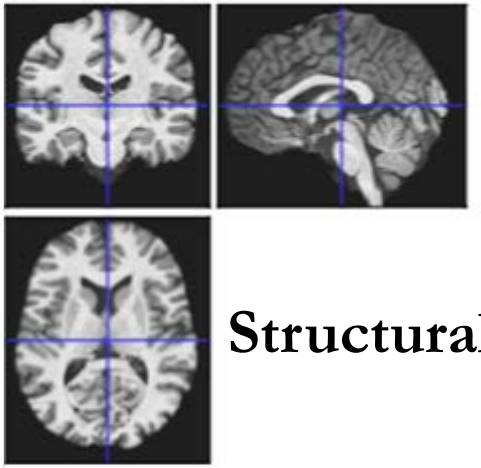
- Pipelines → Automatisation of Preprocessing
- When something goes wrong…how do we even notice?
  - Monitoring, but: cumbersome, when lots of data
  - Thus: Automatise quality monitoring as well via pipelines
- Required: Suitable performance measures
  - Single image: visual inspection geometry/contrast/noise/SNR
    - structural image
  - Time series: Statistical Images (Mean, SD, tSNR, max(abs(diff)))
    - functional images

*Welvaert (2013), PLoS One* *Friedman/Glover (2006), JMRI*

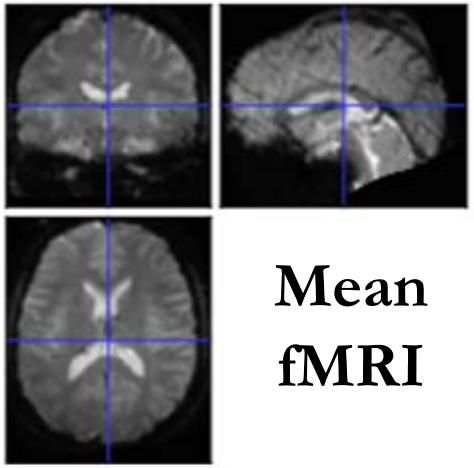
# Statistical Images



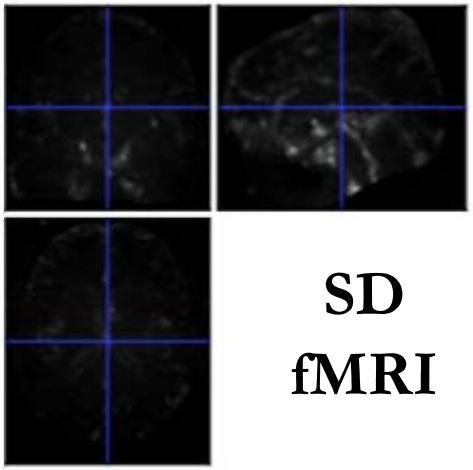
- Structural Image:
  visual inspection geometry/ contrast/noise/SNR
- Functional Time Series:
  - Mean => Artifact levels (localization)
  - SD => Fluctuation levels
  - tSNR = Mean/SD
    => sensitivity for BOLD signal changes
  - Diff = max(abs(diff)) or odd – even
    => outlier detection, image noise



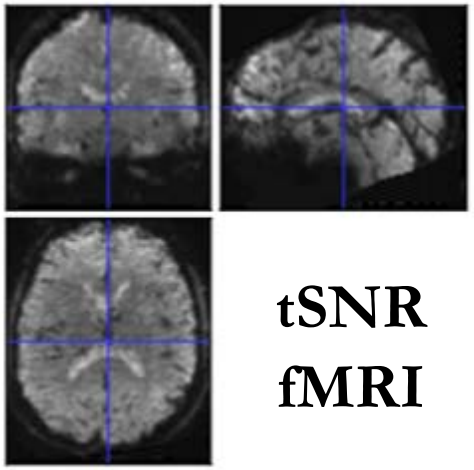
Structural



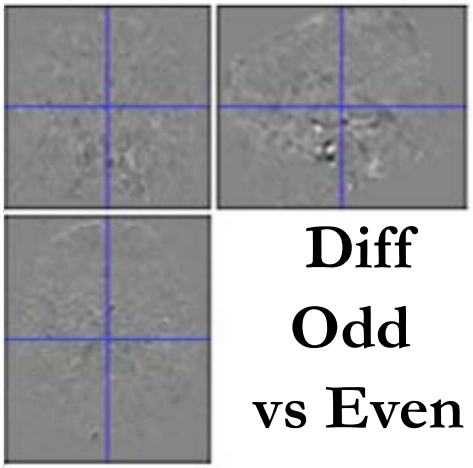
Mean fMRI



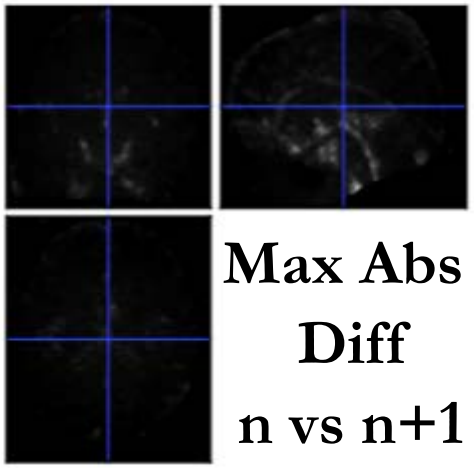
SD fMRI



tSNR fMRI



Diff Odd vs Even



Max Abs Diff n vs n+1

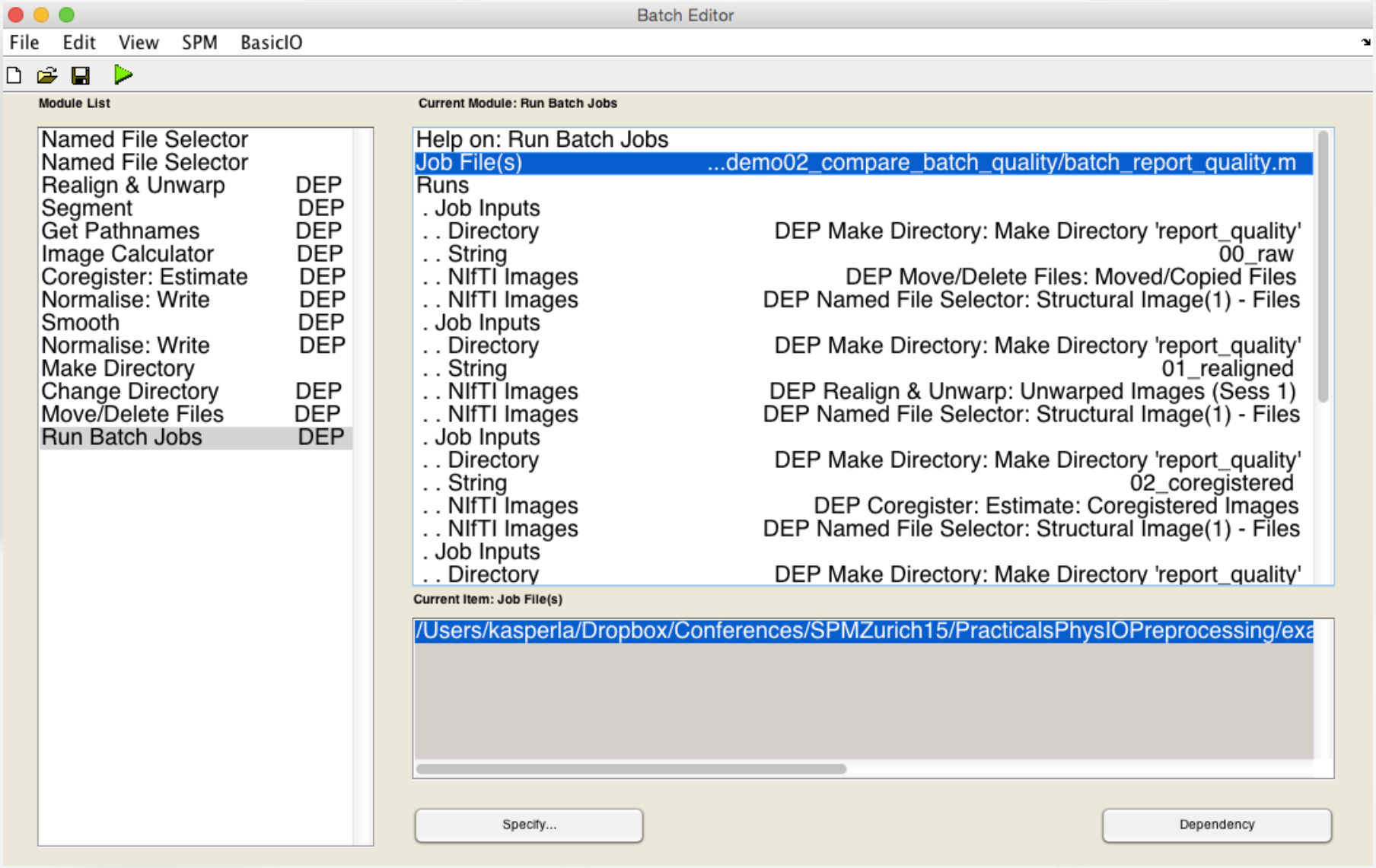
# Demo 2



- Complete preprocessing & monitoring pipeline for fMRI
  - Based on `spm12/batches/preproc_fmri`
  - Includes `batch_report_quality.m` to visualize and save quality measures after each preprocessing step
- Run via
  1. Cleanup processed data of subject01: `init_reset_example(2)`
  2. Load `subject01/batches/demo02_simple_batch_preproc/batch_preproc_fmri_report_quality.m`
  3. Either via GUI or `spm_jobman('interactive', 'batch_preproc_fmri_report_quality.m' );`
     - Alternative: Run batch directly from command line: `spm_jobman('run', 'batch_preproc_fmri_report_quality.m' );`

# Demo 2 - GUI




Batch Editor
File Edit View SPM BasicIO
Module List
Named File Selector
Named File Selector
Realign & Unwarp DEP
Segment DEP
Get Pathnames DEP
Image Calculator DEP
Coregister: Estimate DEP
Normalise: Write DEP
Smooth DEP
Normalise: Write DEP
Make Directory
Change Directory DEP
Move/Delete Files DEP
Run Batch Jobs DEP
Current Module: Run Batch Jobs
Help on: Run Batch Jobs
Job File(s) ...demo02_compare_batch_quality/batch_report_quality.m
Runs
. Job Inputs
. . Directory DEP Make Directory: Make Directory 'report_quality'
. . String 00_raw
. . NIfTI Images DEP Move/Delete Files: Moved/Copied Files
. . NIfTI Images DEP Named File Selector: Structural Image(1) - Files
. Job Inputs
. . Directory DEP Make Directory: Make Directory 'report_quality'
. . String 01_realigned
. . NIfTI Images DEP Realign & Unwarp: Unwarped Images (Sess 1)
. . NIfTI Images DEP Named File Selector: Structural Image(1) - Files
. Job Inputs
. . Directory DEP Make Directory: Make Directory 'report_quality'
. . String 02_coregistered
. . NIfTI Images DEP Coregister: Estimate: Coregistered Images
. . NIfTI Images DEP Named File Selector: Structural Image(1) - Files
. Job Inputs
. . Directory DEP Make Directory: Make Directory 'report_quality'
Current Item: Job File(s)
/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreprocessing/exa
Specify...
Dependency

# Demo 2 - Output



- Output:
  - `subject01/report_quality/report_quality.ps`
    - PostScript file with all output plots, generated from the following nifti image files
  - `report_quality/00_raw/` => raw time series statistics
    - `mean.nii` => mean of time series (per pixel)
    - `sd.nii` => standard deviation (per pixel)
    - `snr.nii` => mean/sd (per pixel)
    - `diffOddEven.nii` => SumOddImages – SumEvenImages
    - `maxAbsDiff` => maximum delta image (vol n vs n+1)
  - `subject01/01_realigned` => realigned time series stats
  - …
  - `subject01/04_smoothed` => smoothed time series stats

# Plotting Example



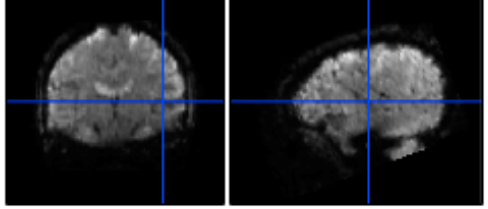
- Example: Comparing temporal SNR throughout preprocessing
  - Temporal SNR per pixel in functional image time series after each preprocessing step
  - Command line code:

```
dirs = {'00_raw', '01_realigned',
'02_coregistered', '03_normalized',
'04_smoothed'}';
files = 'snr.nii';
fpFiles = strcat(dirs, '/', files);
spm_check_registration(fpFiles{:})
```

# Plotting Example

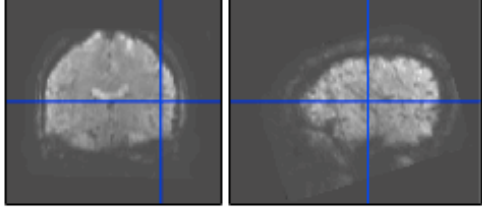


- Example: Temporal SNR per pixel in functional image time series after each preprocessing step
- Right click reveals amazing features
- Result: Increased SNR through realignment and smoothing



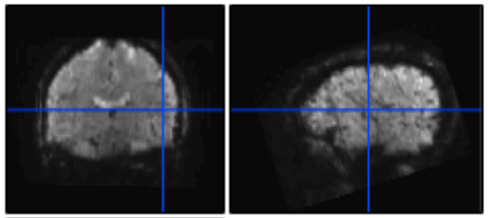
Y = 27.4718

raw



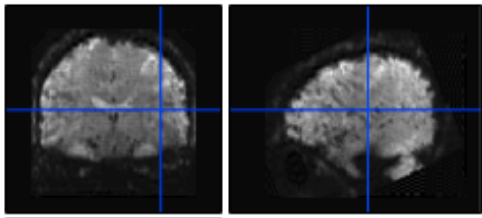
Y = 47.8616

after realign



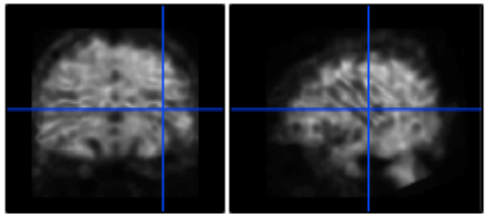
Y = 48.1028

after coreg



Y = 47.7151

norm alized



Y = 185.24

after smooth

# Quiz…

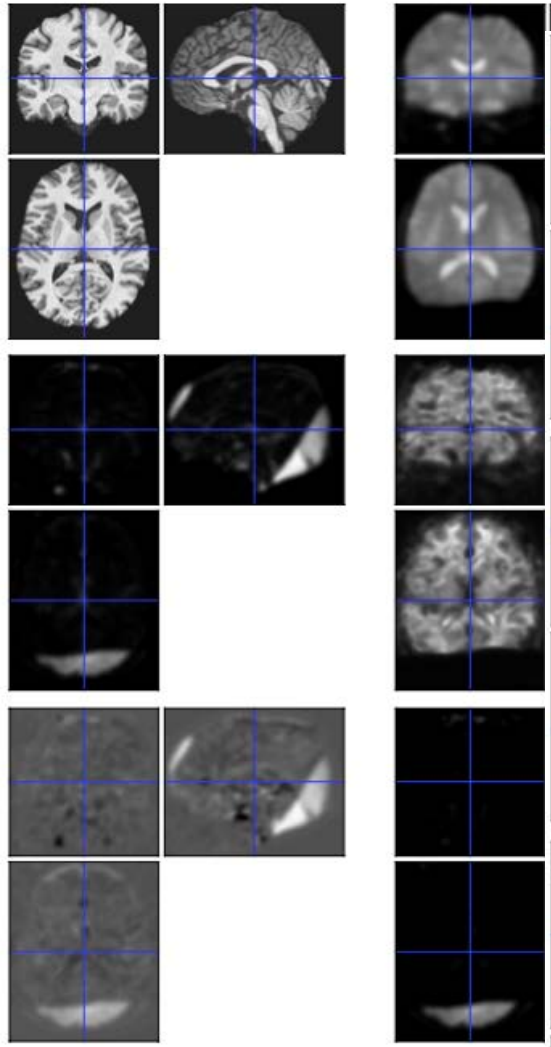


- Task: Spotting unusual quality report images
  - run `demo02…/batch_run_artefact_subjects.m`
    - executes `code/create_artefact_subjects.m`
    - then: executes demo02-preprocessing batch for each subject
  - View for individual subjects: `report_quality.ps`
- What went wrong here?
  - subjectA1
  - subjectA2
  - subjectA3

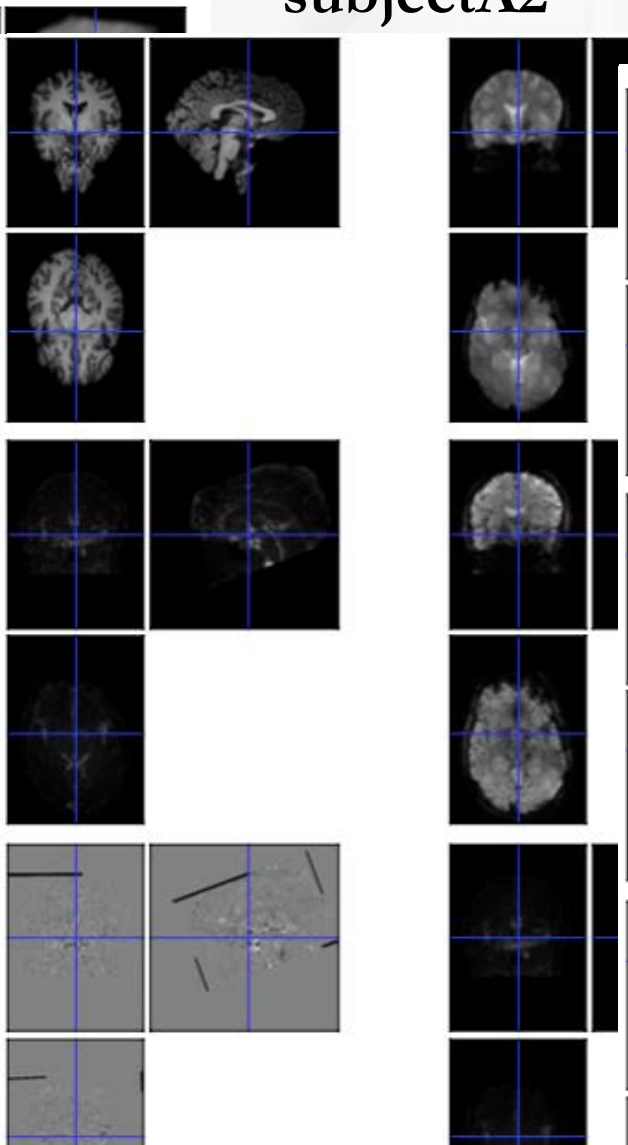
# Quiz…



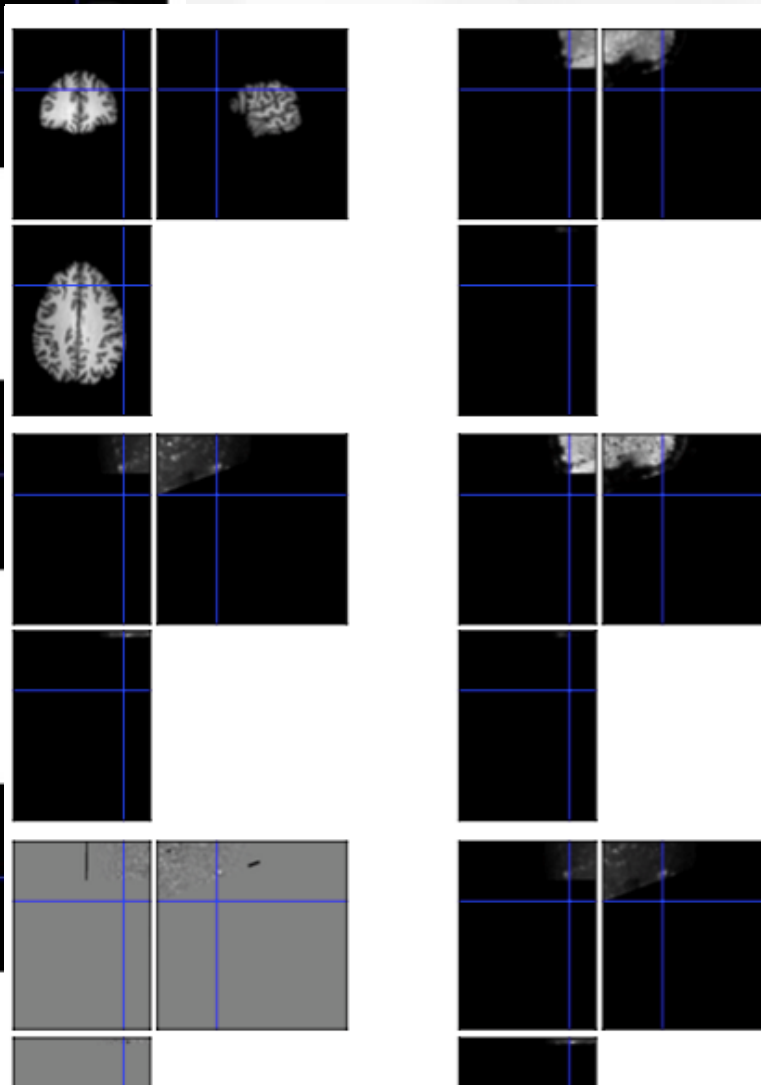
subjectA1



subjectA2



subjectA3



# Quiz…



- Task: Spotting unusual quality report images
- What went wrong here?
  - subjectA1
    - realignment failed
    - one volume rotated (30 degrees around x-axis)
  - subjectA2
    - segmentation failed
    - structural mirrored compared to template
  - subjectA3
    - co-registration failed
    - functional image shifted by 10 cm (x, y, and z) compared to structural

# Comparing Pipelines



- When running different preprocessing pipelines on the same subject:
  - Always start from raw nifti data (structural and functional)
  - Copy it to new processing folder, e.g.
    - subject01/preproc_alternative/fmri/
      /struct/
  - Alter original batch file and save it under new name
  - Swapping preprocessing steps is not simple in the batch editor
    - Solution: save as .m-file and edit in Matlab
    - helper script (code folder):

```
reorder_matlabbatch(fileBatch, indicesBefore, indicesAfter)
```

  - Run different batch files via run_job

# Comparing Pipelines



- Example: When to use Slice timing correction?
- Run:
  - `demo02_compare_batch_quality/batch_compare_preproc`
- this executes:
  - `batch_compare_template_stc_realign`
    - slice timing correction before realignment
  - `batch_compare_template_realign_stc`
    - realignment before slice timing
- Note: Proper Assessment only after statistical analysis possible

# Outline

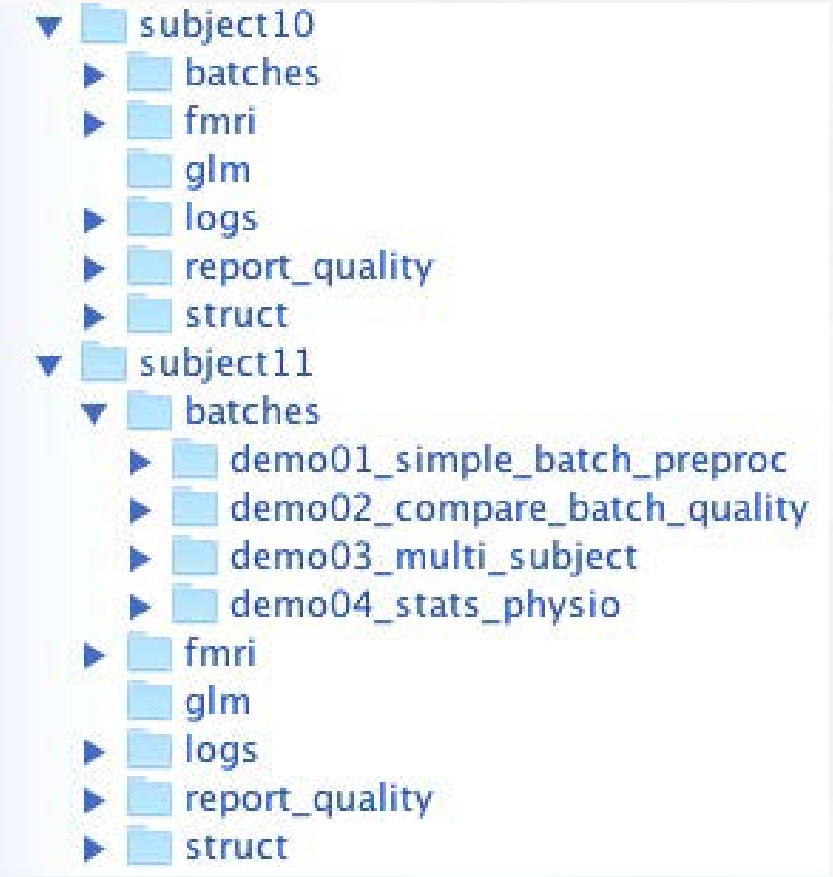


- Setting up a Preprocessing Pipeline in SPM: The Batch Editor
- Monitoring and Comparing Pipelines
- **Multi-subject Pipelines**
  - Staying on Top: Organisation of Data
  - Looping Pipelines over Subjects' Data
  - Tips for Efficient Performance Monitoring
- Integrating Own Code and Toolboxes: Physiological Noise Modeling and Evaluation

# Data Organisation



- One Study Folder
  - sub-folder for code, templates for subject-specific batches
    - modules and dependencies stay the same
- One Folder for each subject
  - therein: same sub-folder structure
  - reduces subject-specific data specification to changing the subject folder
    - batches/ => subject batch with data spec
    - fmri/ => functional data per session
    - struct/ => structural data




- glm/ => statistical analyses
- logs/ => behavioural response files, physiological recordings

# Demo 3 – Example Loop Script

```
% List of open inputs
% Named File Selector: File Set - cfg_files
% Named File Selector: File Set - cfg_files
% Make Directory: Parent Directory - cfg_files

pathStudy       = '/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPrepr
dirSubjectArray = {
    'subject10'
    'subject11'
    };
fileFunctional  = 'fmri/fmri.nii';
fileStructural  = 'struct/struct.nii';

nrun = 2; % enter the number of runs here
jobfile = {'/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreprocessir
jobs = repmat(jobfile, 1, nrun);
inputs = cell(3, nrun);
for crun = 1:nrun
    pathSubject = fullfile(pathStudy, dirSubjectArray{nrun});
    fullpathFunctional = fullfile(pathSubject, fileFunctional);
    fullpathStructural = fullfile(pathSubject, fileStructural);

    % functional file
    inputs{1, crun} = {fullpathFunctional}; % Named File Selector: File Set - cfg_files

    % structural file
    inputs{2, crun} = {fullpathStructural}; % Named File Selector: File Set - cfg_files

    % subject folder
    inputs{3, crun} = {pathSubject}; % Make Directory: Parent Directory - cfg_files
end
spm('defaults', 'FMRI');
spm_jobman('run', jobs, inputs{:});
```

# Demo 3



- Performs Demo 2 (complete preproc + quality report) for multiple subjects (subject10 and subject11)
  - create subject folders: Run `code/create_multi_subject_data.m`
- Run via
  1. Load "job": `subject10/batches/demo03_multi_subject/batch_preproc_fmri_report_quality_job.m` - Inspect!
  2. Load & Run `batch_preproc_fmri_report_quality_gui.m`
  3. Alternative: Open in *Matlab* Editor, inspect construction of subject dependent file names, run in Matlab GUI: `batch_preproc_fmri_report_quality_run.m`

# Outline

- Setting up a Preprocessing Pipeline in SPM: The Batch Editor
- Monitoring and Comparing Pipelines
- Multi-subject Pipelines
- **Integrating Own Code and Toolboxes: Physiological Noise Modeling and Evaluation**
  - Executing Custom Matlab Code within the Pipeline
  - The TAPAS PhysIO Toolbox
  - Automatic (Noise) Modeling and Contrast Reporting

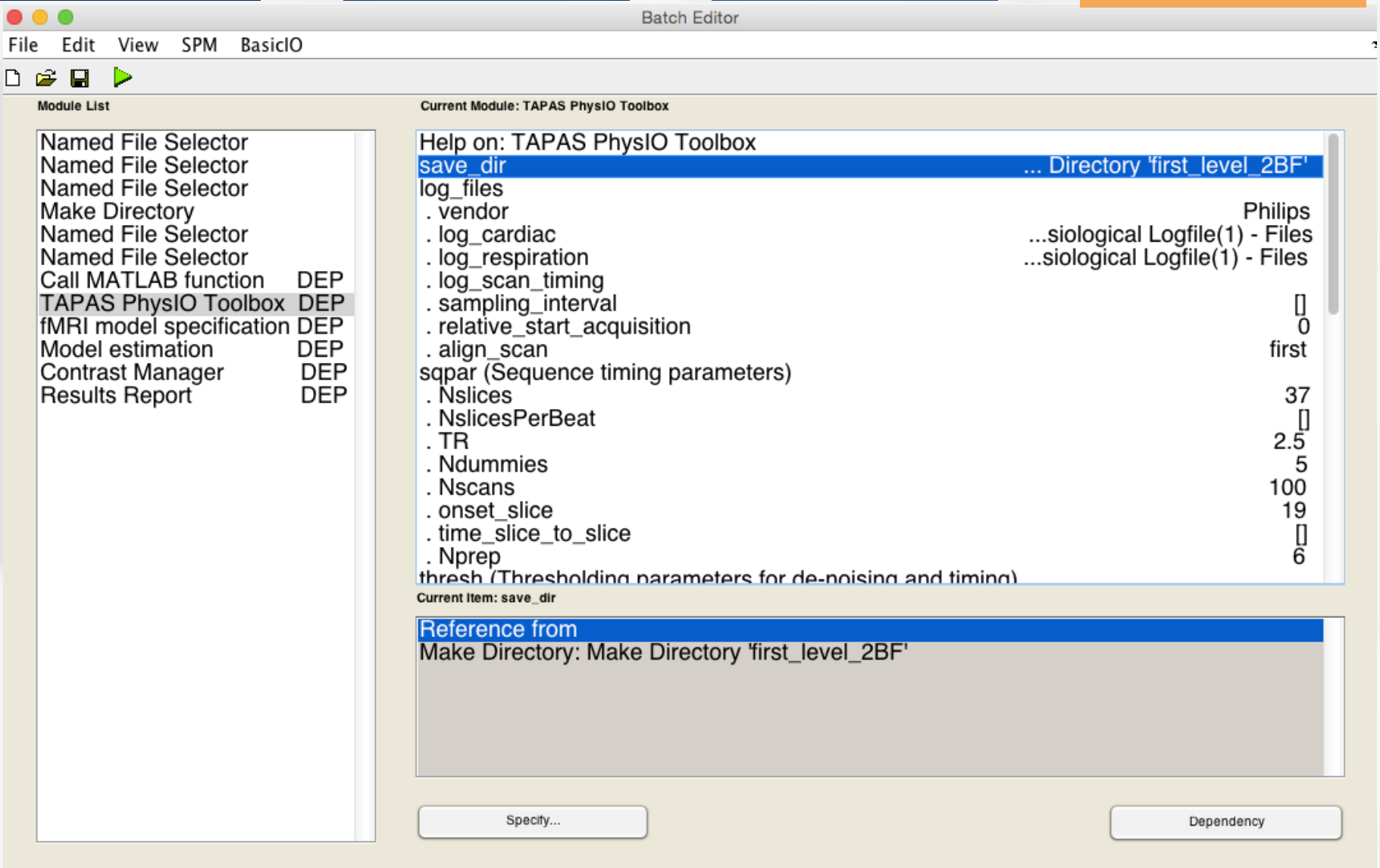
# Demo 4



- Performs whole single-subject analysis except preprocessing
  - creates behavioral regressors (visual stimuli/button responses)
    - `multiple_conditions.mat`
  - creates nuisance regressors, including RETROICOR via the TAPAS PhysIO Toolbox
    - `multiple_regressors.mat`
  - Sets up 1st level analysis GLM & estimates it
  - Estimates F-contrasts of interests and plots them to .ps file
- Run via
  1. `spm_jobman('interactive', subject01/batches/demo04_stats_physio/batch_physio_glm_contrasts.m)`

# Demo 4

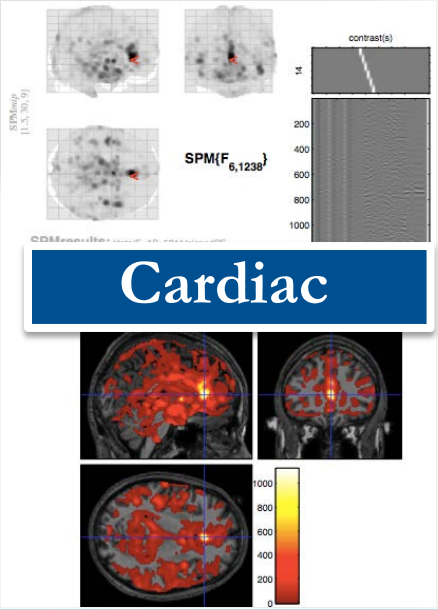


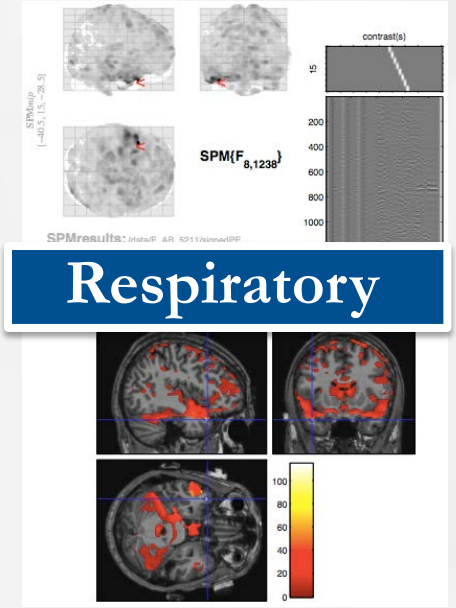

Batch Editor
File Edit View SPM BasicIO
Module List
Named File Selector
Named File Selector
Named File Selector
Make Directory
Named File Selector
Named File Selector
Call MATLAB function DEP
TAPAS PhysIO Toolbox DEP
fMRI model specification DEP
Model estimation DEP
Contrast Manager DEP
Results Report DEP
Current Module: TAPAS PhysIO Toolbox
Help on: TAPAS PhysIO Toolbox
save_dir ... Directory 'first_level_2BF'
log_files
. vendor Philips
. log_cardiac ...siological Logfile(1) - Files
. log_respiration ...siological Logfile(1) - Files
. log_scan_timing
. sampling_interval []
. relative_start_acquisition 0
. align_scan first
sqpar (Sequence timing parameters)
. Nslices 37
. NslicesPerBeat []
. TR 2.5
. Ndummies 5
. Nscans 100
. onset_slice 19
. time_slice_to_slice []
. Nprep 6
thresh (Thresholding parameters for de-noising and timing)
Current Item: save_dir
Reference from
Make Directory: Make Directory 'first_level_2BF'
Specify...
Dependency

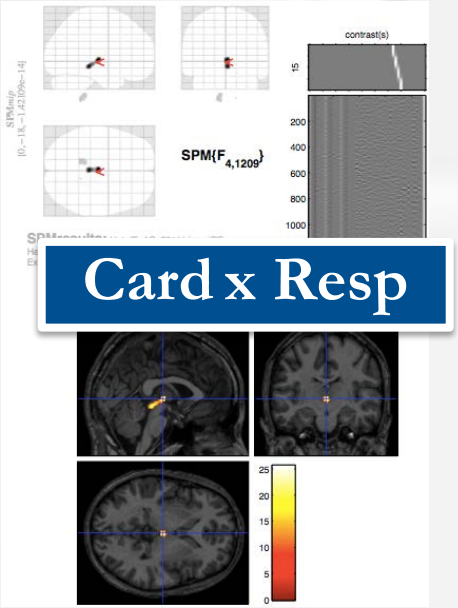
# Pipeline Noise Modeling/Report

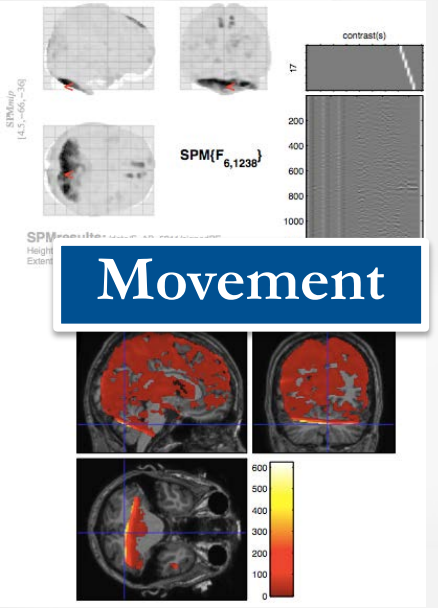


- Model physiological noise as nuisance regressors
- Check noise correction via explained variance (F-contrasts)
  - Automatic contrast creation and report (glass brain): Batch Editor SPM => Stats => Contrast Manager/Results Report
  - With underlays: `tapas_physio_report_contrasts`



# Conclusion



- Setting up a Preprocessing Pipeline in SPM:

  The Batch Editor

  *…best way for reproducible, documented preprocessing*

- Monitoring and Comparing Pipelines

  *…within Batch Editor, CheckReg Tool, Stat Images per step*

- Multi-subject Pipelines

  *…via template batch, same sub-folders & Matlab script (subj.-loop)*

- Integrating Own Code & Physiological Noise Correction

  *…by calling Matlab functions within Batch Editor, e.g. PhysIO Toolbox*