The Art and Pitfalls of fMRI Preprocessing

An SPM Perspective

Lars Kasper

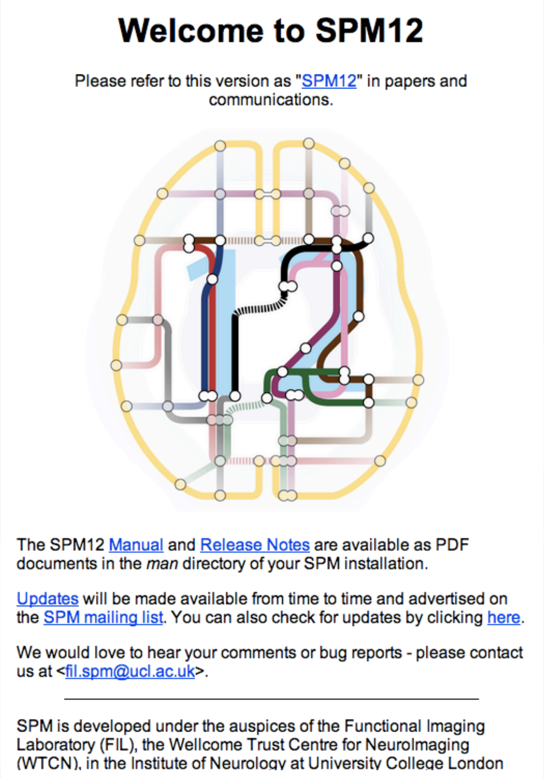
June 14th, 2015

MR-Technology Group & Translational Neuromodeling Unit

Institute for Biomedical Engineering
University of Zurich and ETH Zurich

Statistical Parametric Mapping

- What is SPM?
  - SPM = Statistical Parametric Mapping
  - Developed at the Functional Imaging Laboratory (FIL) at UCL, London
  - Much more than fMRI (E/MEG/PET)
- Philosophy
  - Free Software (GPL), Matlab-based
    - Script language, also for non-programmers
    - Simple debugging, interactive operation via command window, powerful editor
  - Easy access/modification of source code
    - Combination with self-written custom code



Welcome to SPM12

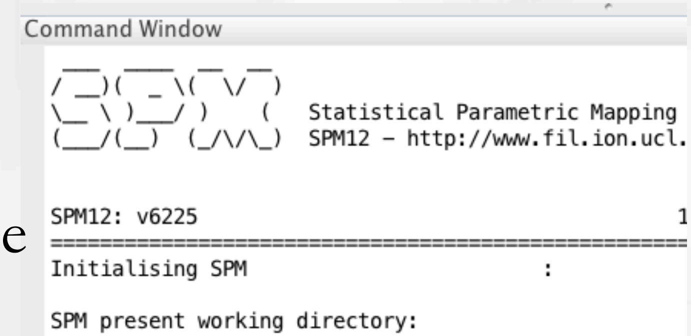
Please refer to this version as "SPM12" in papers and communications.

The SPM12 Manual and Release Notes are available as PDF documents in the *man* directory of your SPM installation.

Updates will be made available from time to time and advertised on the SPM mailing list. You can also check for updates by clicking here.

We would love to hear your comments or bug reports - please contact us at <fil.spm@ucl.ac.uk>.

SPM is developed under the auspices of the Functional Imaging Laboratory (FIL), the Wellcome Trust Centre for NeuroImaging (WTCN), in the Institute of Neurology at University College London



```
  ___  ____  __  __
 / __)(  _ \(  \/  )
 \__ \ )___/ )    (   Statistical Parametric Mapping
 (___/(__)  (_/\/\_)  SPM12 - http://www.fil.ion.ucl.


SPM12: v6225                                          1
=======================================================
Initialising SPM                            :

SPM present working directory:
```

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
- Monitoring and Comparing Pipelines
- Multi-subject Pipelines
- Integrating Own Code and Toolboxes: Physiological Noise Modeling and Evaluation

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

Overview of SPM for fMRI

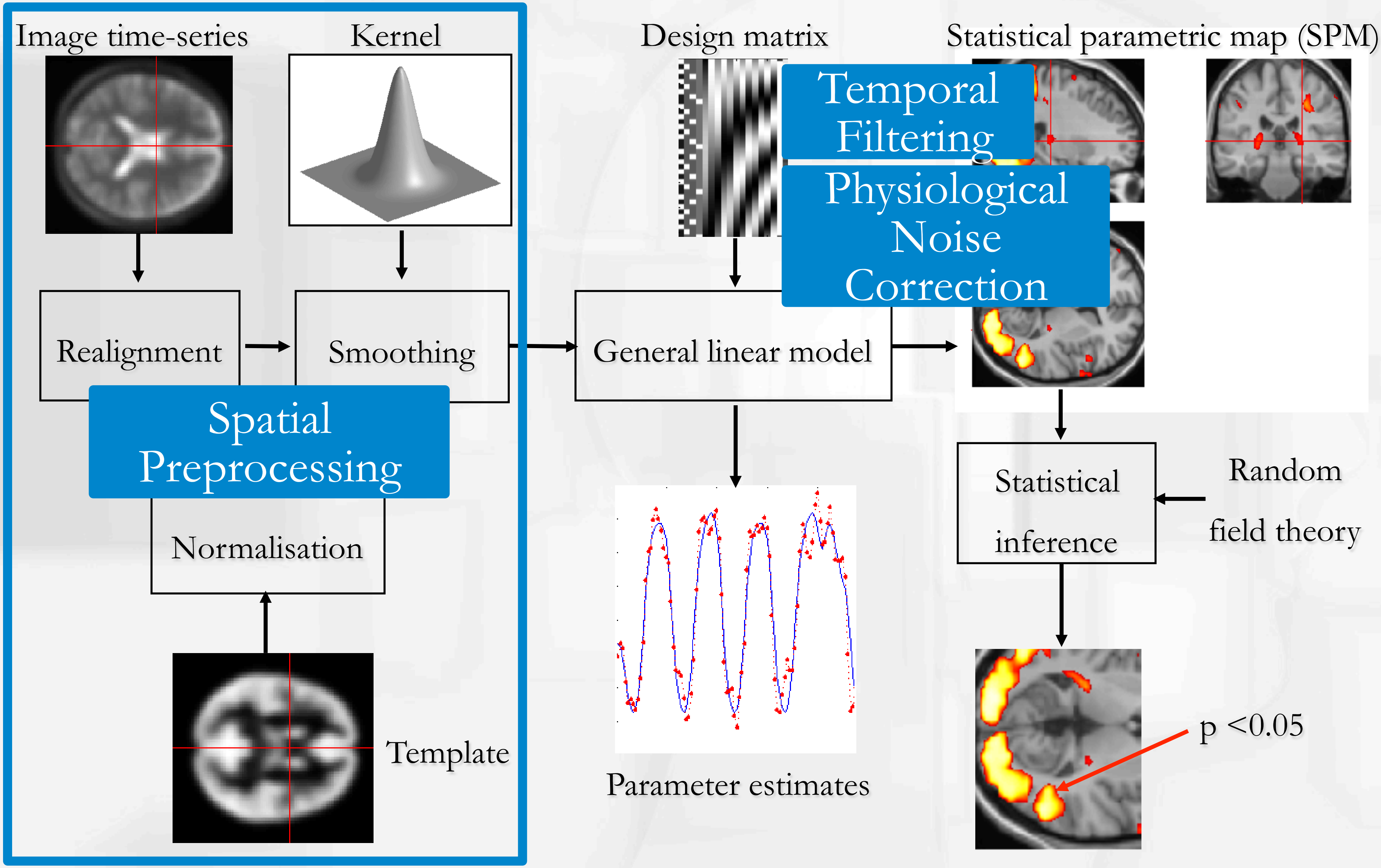


Image time-series

Kernel

Design matrix

Statistical parametric map (SPM)

Temporal Filtering

Physiological Noise Correction

Realignment

Smoothing

General linear model

Spatial Preprocessing

Normalisation

Template

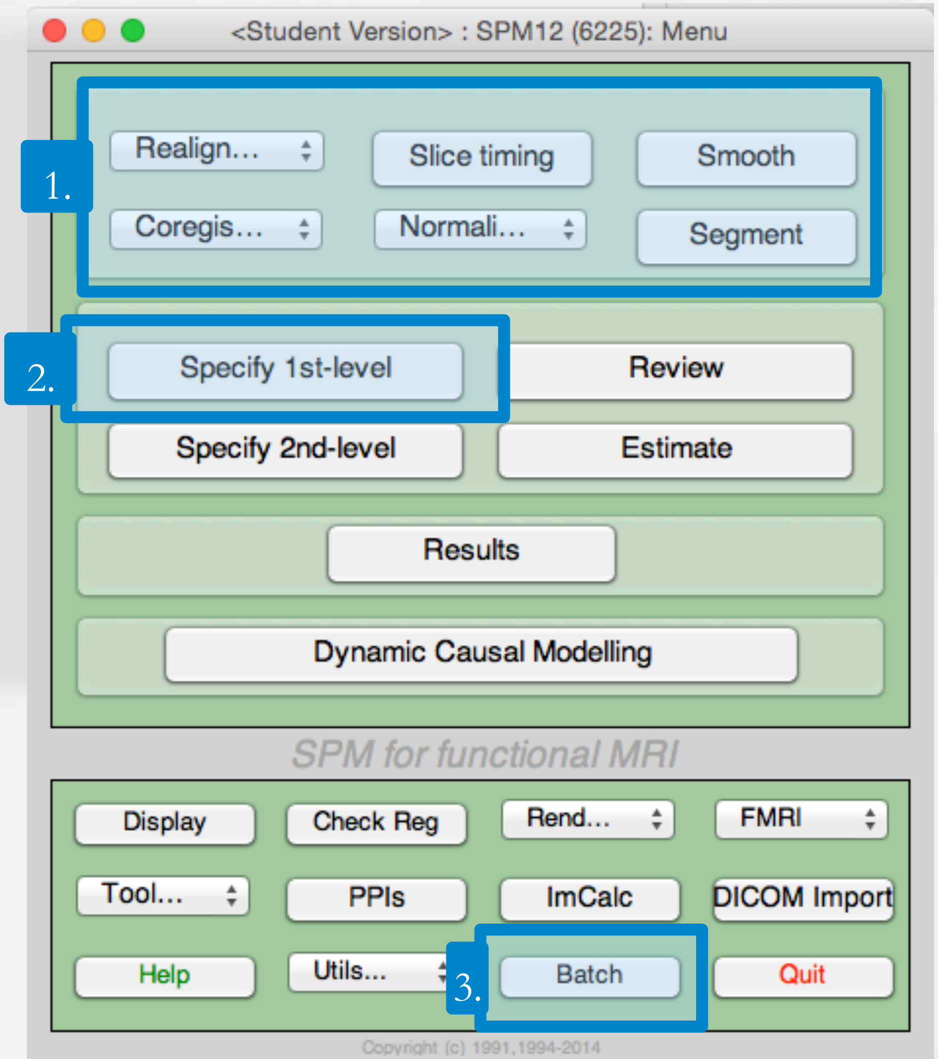
Parameter estimates

Statistical inference

Random field theory

$p < 0.05$

The SPM GUI



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

Spatial Preprocessing: SPM vocabulary

- SPM uses different names for different modalities of image registration
- depending on input images and allowed transformations

Realignment

- Intra-modal image registration
  - e.g. functional images
- rigid body transformations
  - translation/rotation

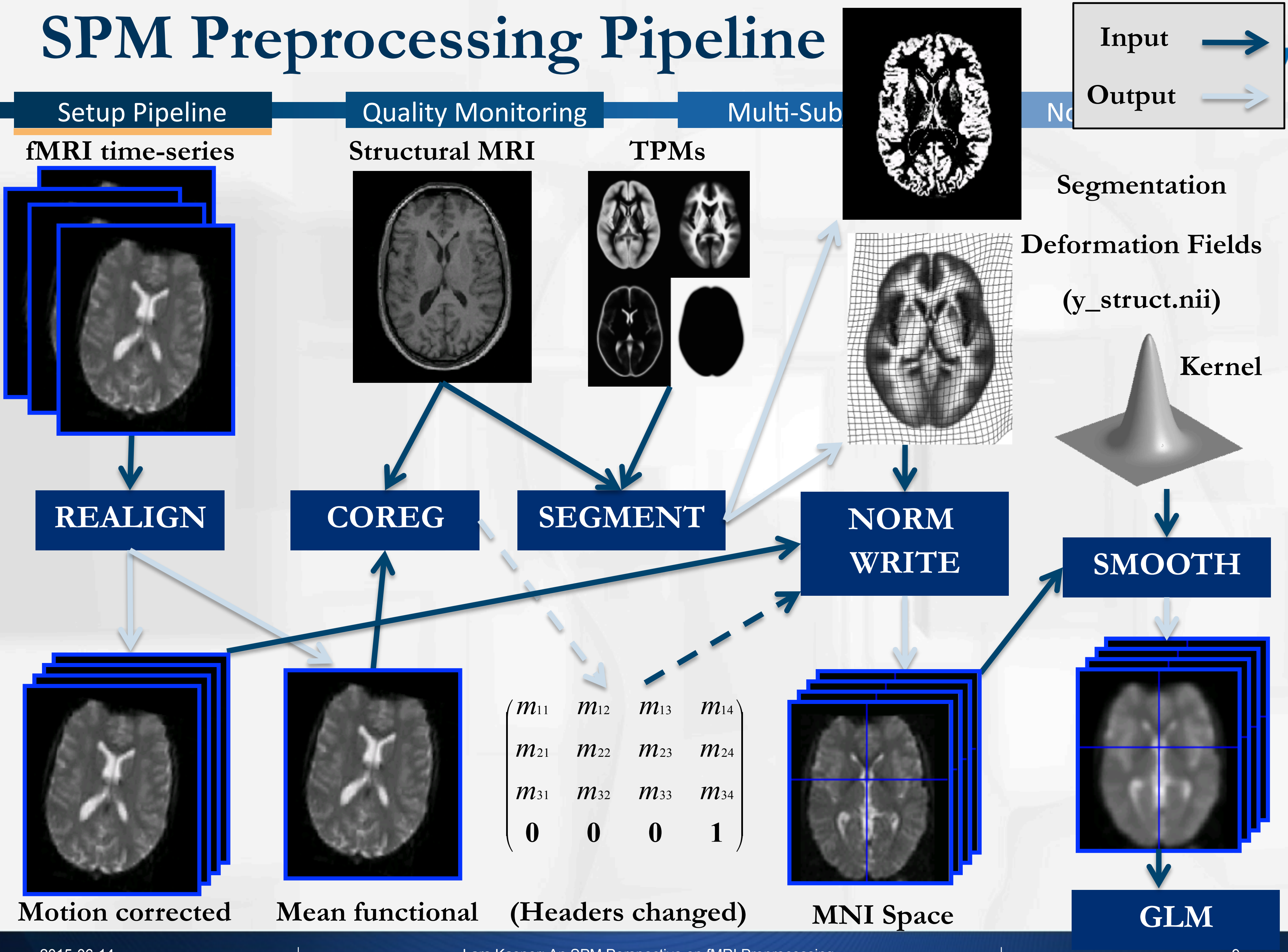
Co-Registration

- Inter-modal registration
  - e.g. T1/T2 contrast
  - functional to structural image
- affine transformations
  - rigid body
  - stretching/shearing

Normalisation

- Multi-modal registration
  - e.g. T1 and/or T2
  - structural image(s) to template
- non-linear transformations
  - voxel-wise mapping (deformation fields)

SPM Preprocessing Pipeline



Input →

Output →

fMRI time-series

Structural MRI

TPMs

Segmentation

Deformation Fields

(y_struct.nii)

Kernel

REALIGN

COREG

SEGMENT

NORM WRITE

SMOOTH

GLM

Motion corrected

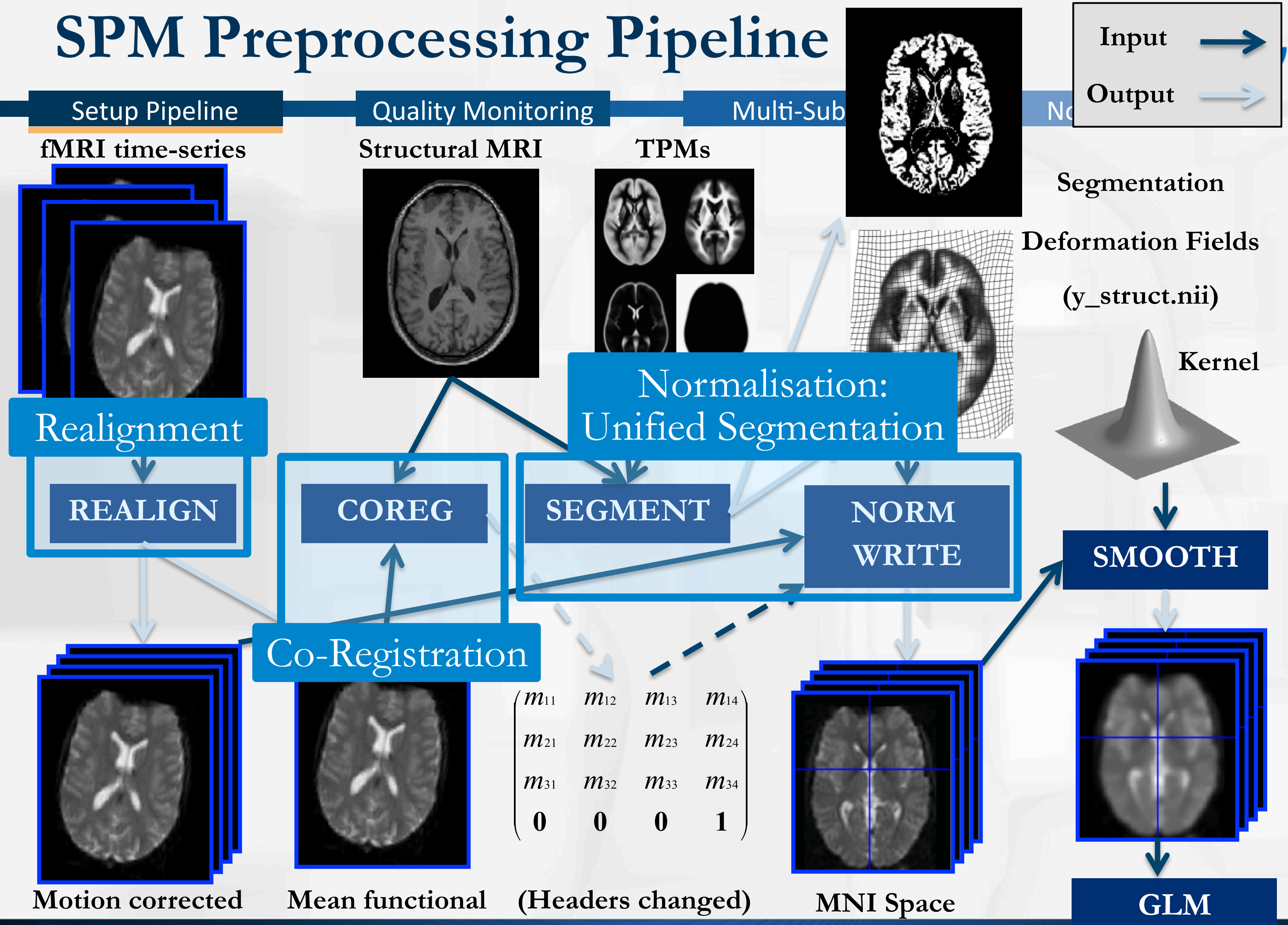
Mean functional

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

(Headers changed)

MNI Space

SPM Preprocessing Pipeline



Setup Pipeline

Quality Monitoring

Multi-Sub

No

Input

Output

fMRI time-series

Structural MRI

TPMs

Segmentation

Deformation Fields

(y_struct.nii)

Kernel

Realignment

REALIGN

COREG

Normalisation:
Unified Segmentation

SEGMENT

NORM WRITE

SMOOTH

Co-Registration

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

Motion corrected

Mean functional

(Headers changed)

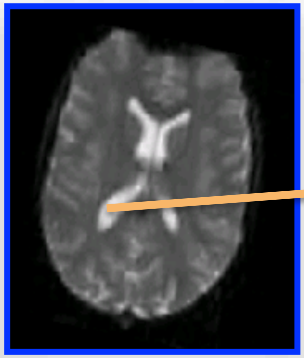
MNI Space

GLM

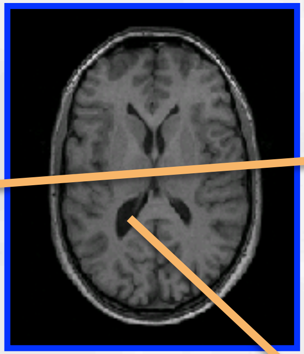
Theory: Co-Registration

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling

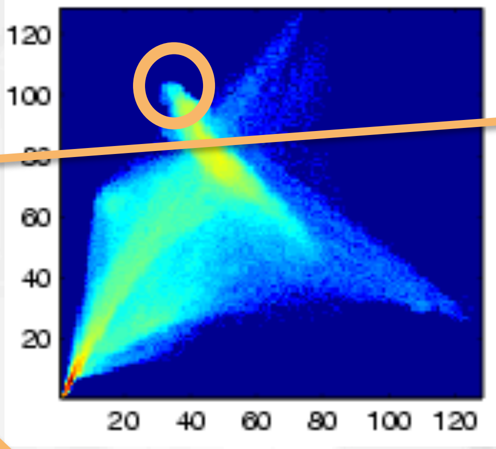
Mean functional



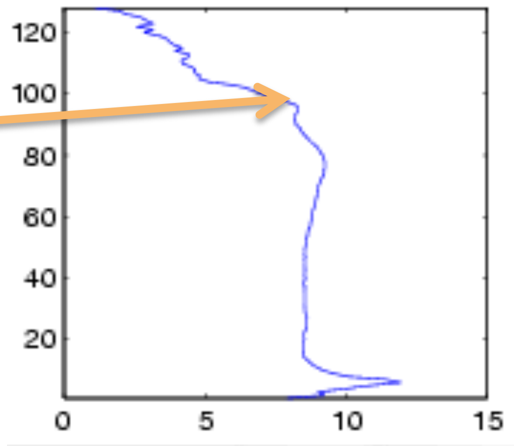
Anatomical MRI



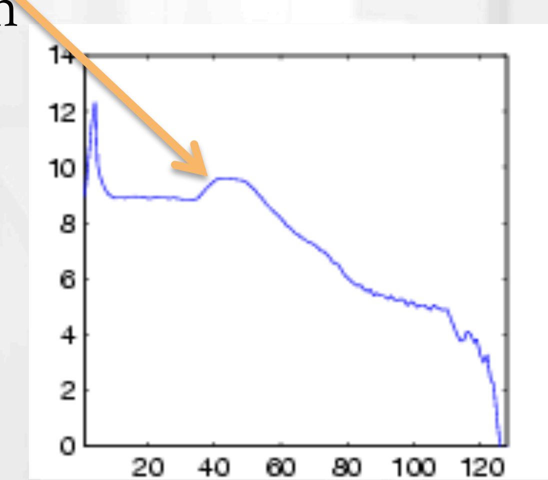
Joint Histogram



Marginal Histogram



- Joint and marginal Histogram
  - Quantify how well one image predicts the other
    - how much shared information
  - Joint probability distribution estimated from joint histogram
  - Aligned voxels = crisp peaks in joint histogram



intensity bins

structural

intensity bins

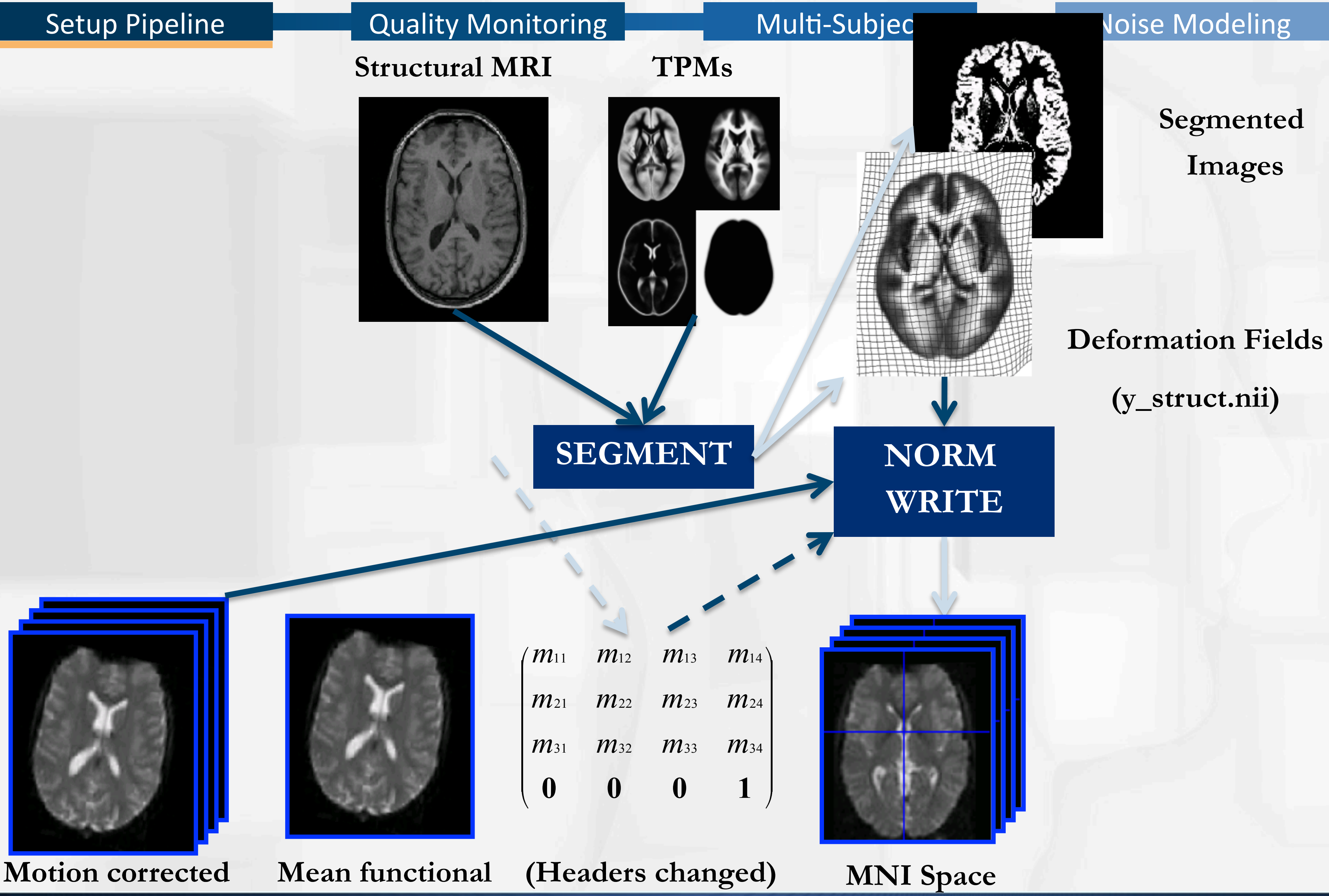
functional

Joint Histogram:

$h(i_f,i_s)$

Count of voxels who have intensity i_f in functional and i_s in structural image

Theory: Unified Segmentation



Structural MRI

TPMs

Segmented Images

Deformation Fields

(y_struct.nii)

SEGMENT

NORM WRITE

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

Motion corrected

Mean functional

(Headers changed)

MNI Space

Theory: Segmentation/Normalisation

- Why is normalisation difficult?
  - No simple similarity measure, a lot of possible transformations…
  - Different Imaging Sequences (Contrasts, geometry distortion)
  - Noise, artefacts, partial volume effects
  - Intensity inhomogeneity (bias field)
- **Normalisation** of segmented tissues is more robust and precise than of original image
- Tissue **segmentation** benefits from spatially aligned tissue probability maps (of prior segmentation data)

➡ Motivates a unified model of segmentation/normalisation

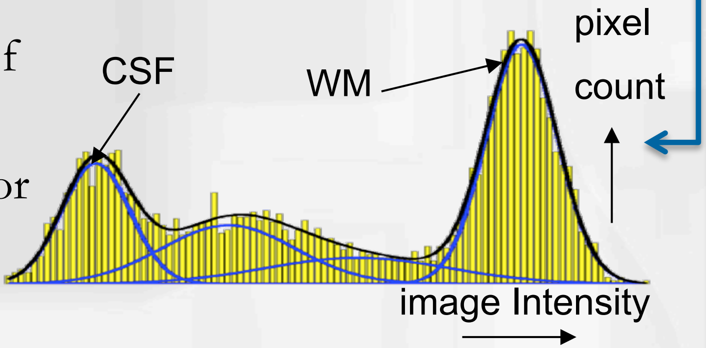
Theory: Unified Model Segmentation

- Bayesian generative model[1] of voxel intensities from tissue class probabilities, deformation fields and bias fields

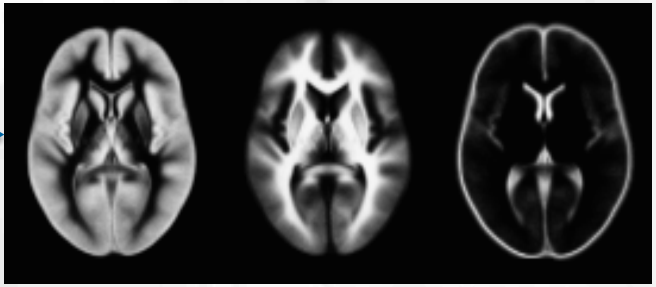
[1] Ashburner & Friston (2005), Neuroimage

Gaussian Mixture Model

probability of intensity in given voxel for tissue class



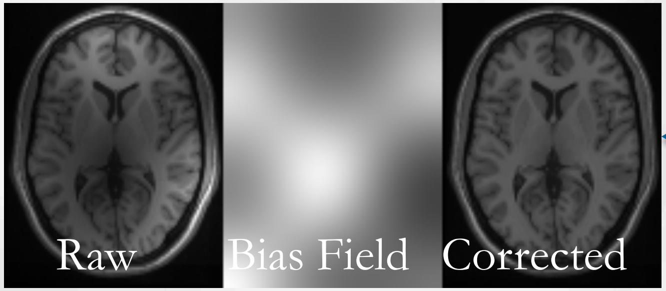
Prior: Tissue probability maps



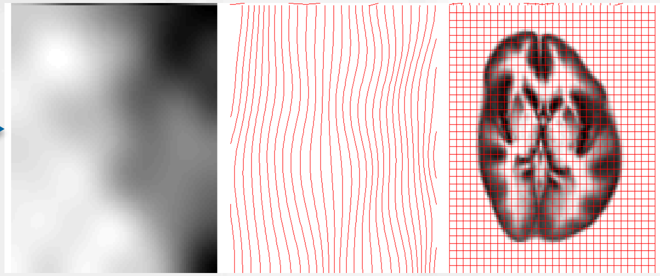
TPMs in MNI space

Bias Field

coil inhomo-geneities

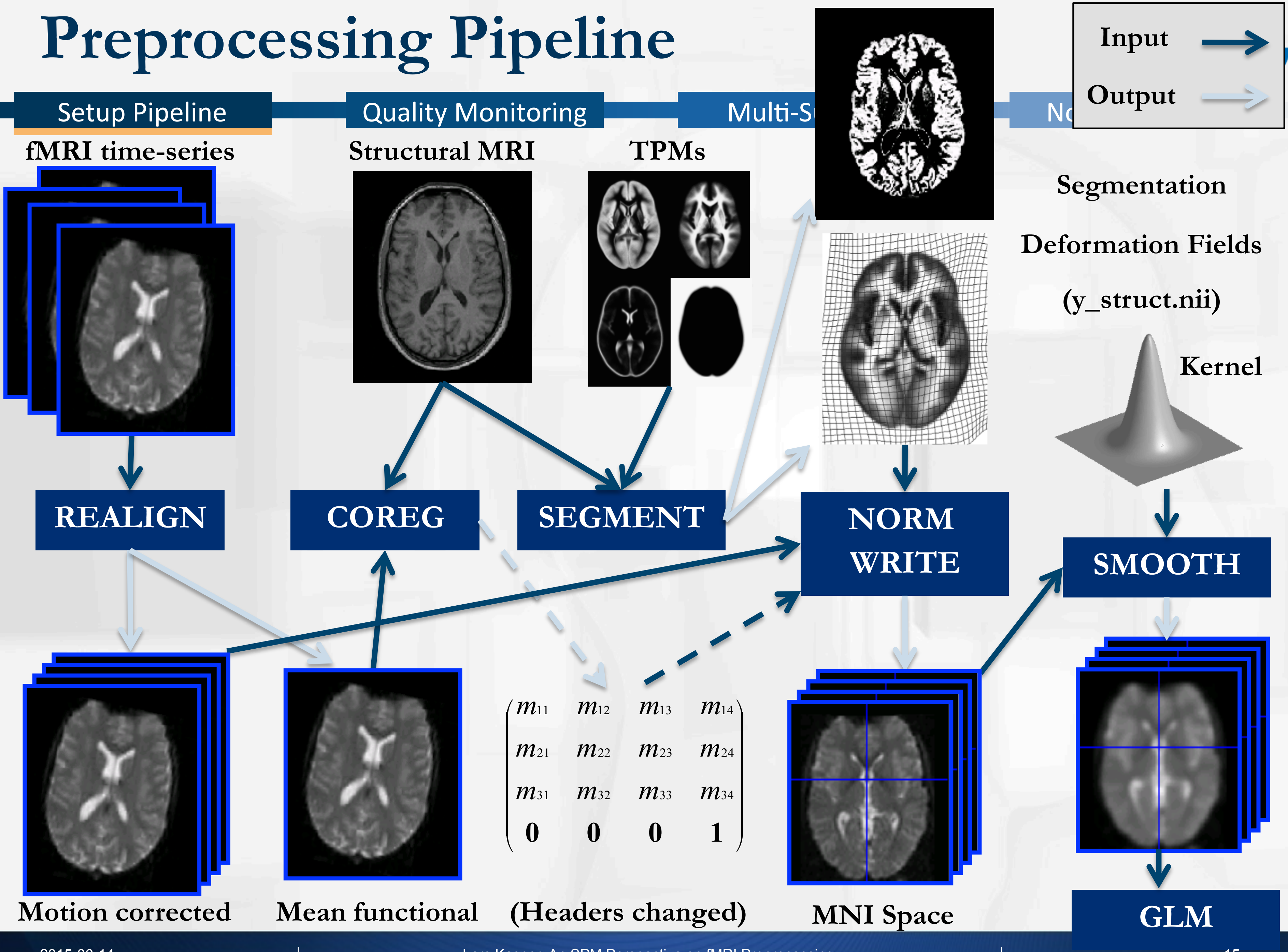


Deformation Fields



discrete cosine transforms

Preprocessing Pipeline



Setup Pipeline | Quality Monitoring | Multi-S | No

Input

Output

fMRI time-series

Structural MRI

TPMs

Segmentation

Deformation Fields

(y_struct.nii)

Kernel

REALIGN

COREG

SEGMENT

NORM WRITE

SMOOTH

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

Motion corrected

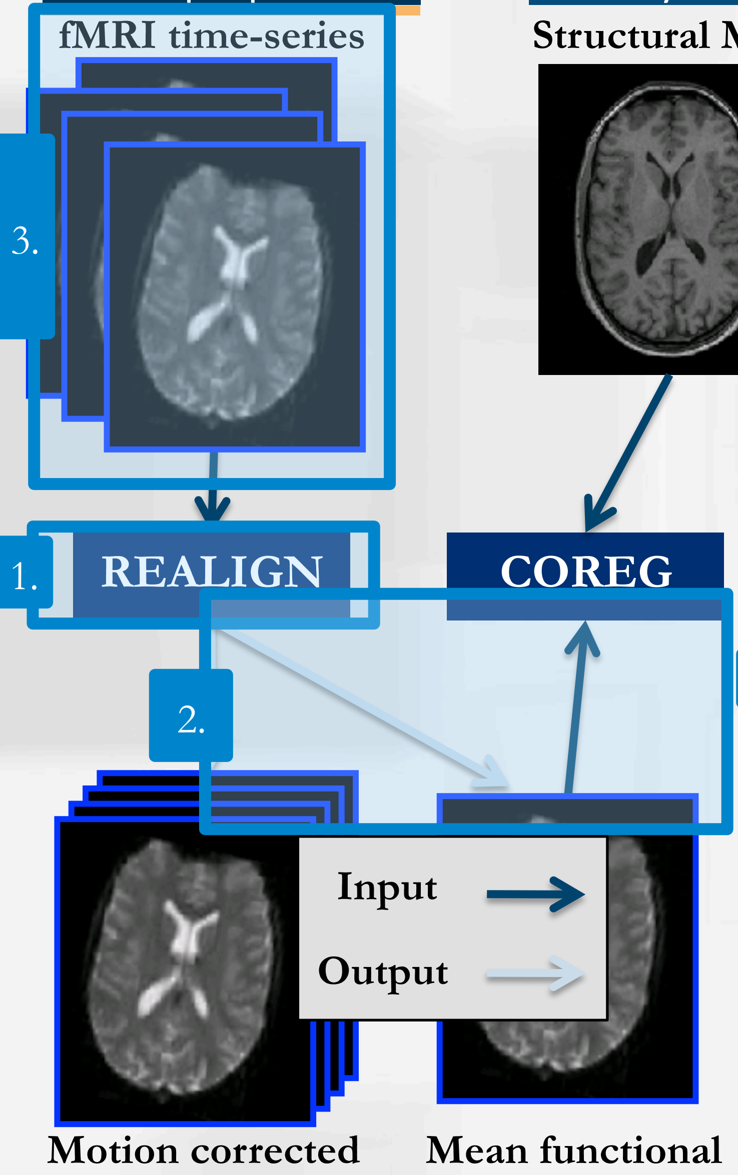
Mean functional

(Headers changed)

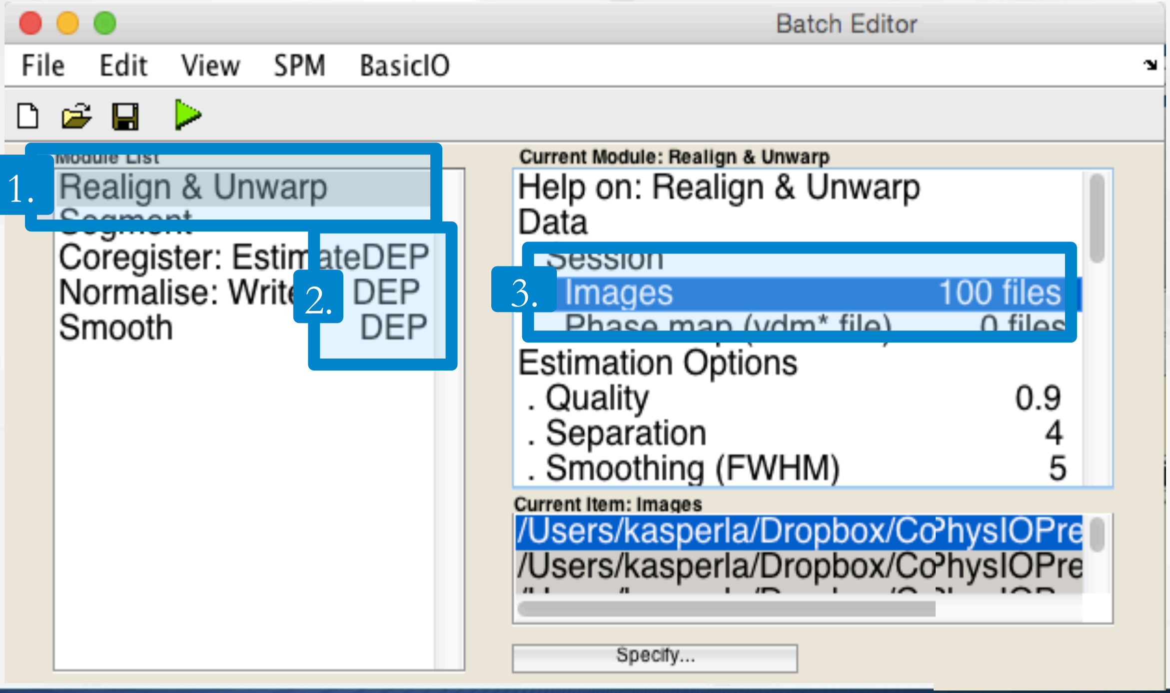
MNI Space

GLM

From Pipelines to Batches



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



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

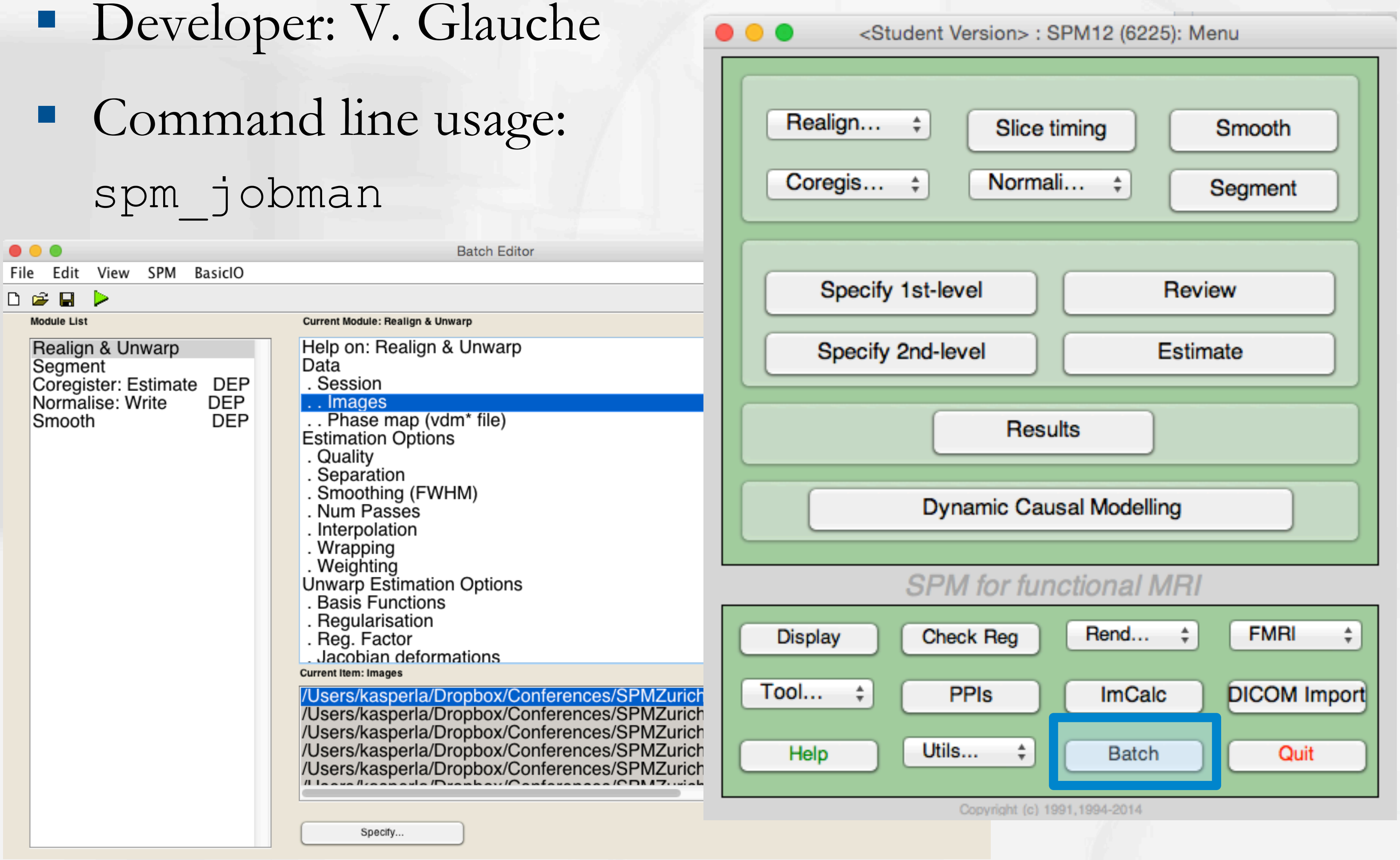
Current Item: Images

/Users/kasperla/Dropbox/PhysIOPre
/Users/kasperla/Dropbox/PhysIOPre

The Batch Editor in SPM

- Developer: V. Glauche
- Command line usage:

  `spm_jobman`



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
. . Phase map (vdm* file)
Estimation Options
. Quality
. Separation
. Smoothing (FWHM)
. Num Passes
. Interpolation
. Wrapping
. Weighting
Unwarp Estimation Options
. Basis Functions
. Regularisation
. Reg. Factor
. Jacobian deformations

Current Item: Images

/Users/kasperla/Dropbox/Conferences/SPMZurich
/Users/kasperla/Dropbox/Conferences/SPMZurich
/Users/kasperla/Dropbox/Conferences/SPMZurich
/Users/kasperla/Dropbox/Conferences/SPMZurich
/Users/kasperla/Dropbox/Conferences/SPMZurich

Specify...

<Student Version> : SPM12 (6225): Menu

Demo 1

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling

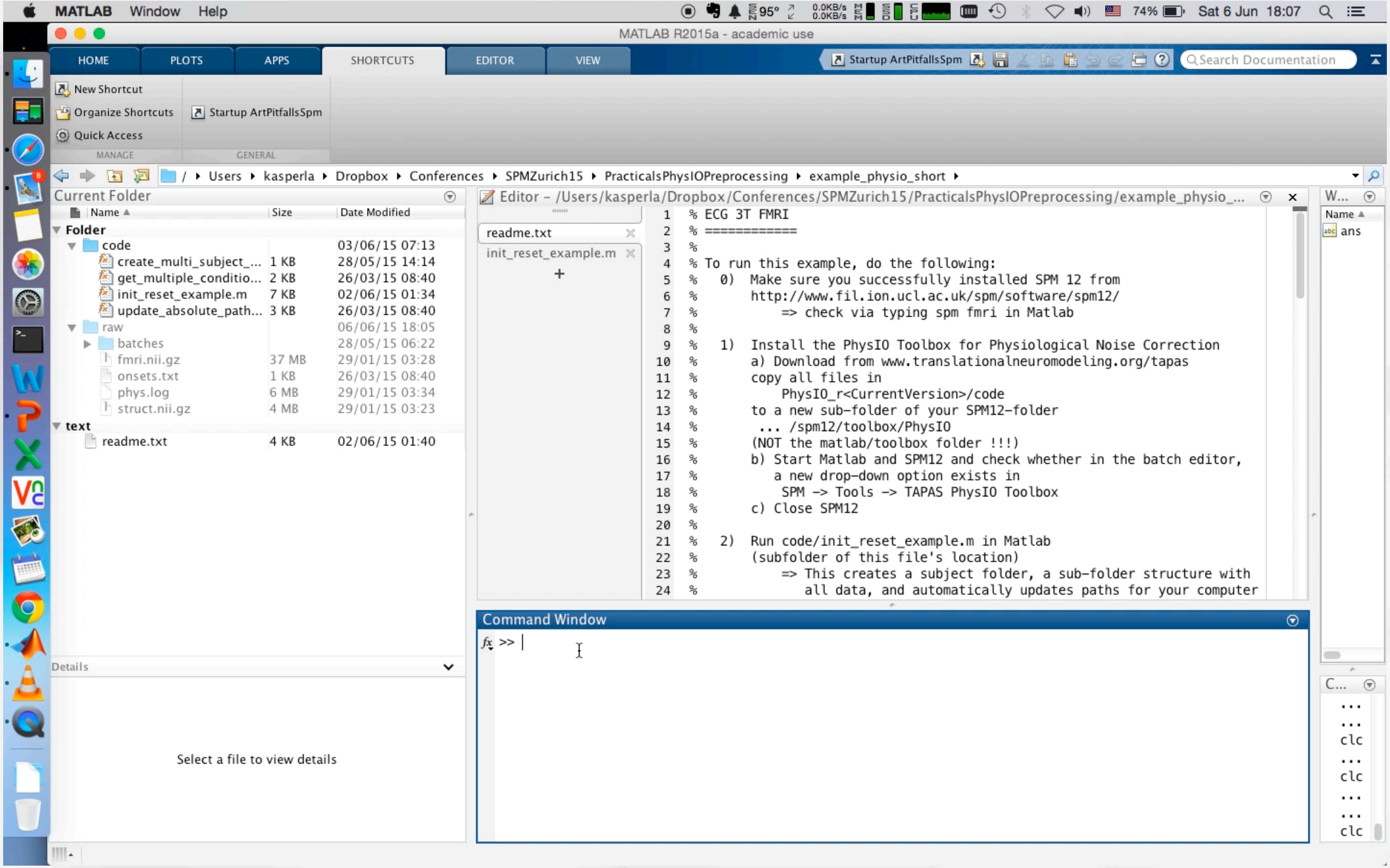
- Simple preprocessing pipeline for fMRI (G. Ridgway)
  - Based on `spm12/batches/preproc_fmri_simplified.m`
  - Data: Social Learning (A. Diaconescu[1], TNU Zurich, Philips 3T)
- After download:

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. 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' );`
4. Run Batch (Press Play)

[1] Diaconescu (2014), PLoS CB

Demo 1 – Hands-On Movie

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling



Name ▲	Size	Date Modified
Folder		
code		03/06/15 07:13
create_multi_subject_...	1 KB	28/05/15 14:14
get_multiple_conditio...	2 KB	26/03/15 08:40
init_reset_example.m	7 KB	02/06/15 01:34
update_absolute_path...	3 KB	26/03/15 08:40
raw		06/06/15 18:05
batches		28/05/15 06:22
fmri.nii.gz	37 MB	29/01/15 03:28
onsets.txt	1 KB	26/03/15 08:40
phys.log	6 MB	29/01/15 03:34
struct.nii.gz	4 MB	29/01/15 03:23
text		
readme.txt	4 KB	02/06/15 01:40

```
% ECG 3T FMRI
% ============
%
% To run this example, do the following:
%  0)  Make sure you successfully installed SPM 12 from
%      http://www.fil.ion.ucl.ac.uk/spm/software/spm12/
%          => check via typing spm fmri in Matlab
%
%  1)  Install the PhysIO Toolbox for Physiological Noise Correction
%      a) Download from www.translationalneuromodeling.org/tapas
%      copy all files in
%          PhysIO_r<CurrentVersion>/code
%      to a new sub-folder of your SPM12-folder
%       ... /spm12/toolbox/PhysIO
%      (NOT the matlab/toolbox folder !!!)
%      b) Start Matlab and SPM12 and check whether in the batch editor,
%         a new drop-down option exists in
%          SPM -> Tools -> TAPAS PhysIO Toolbox
%      c) Close SPM12
%
%  2)  Run code/init_reset_example.m in Matlab
%      (subfolder of this file's location)
%          => This creates a subject folder, a sub-folder structure with
%             all data, and automatically updates paths for your computer
```

```
>> |
```

Demo 1 - Results

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling

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

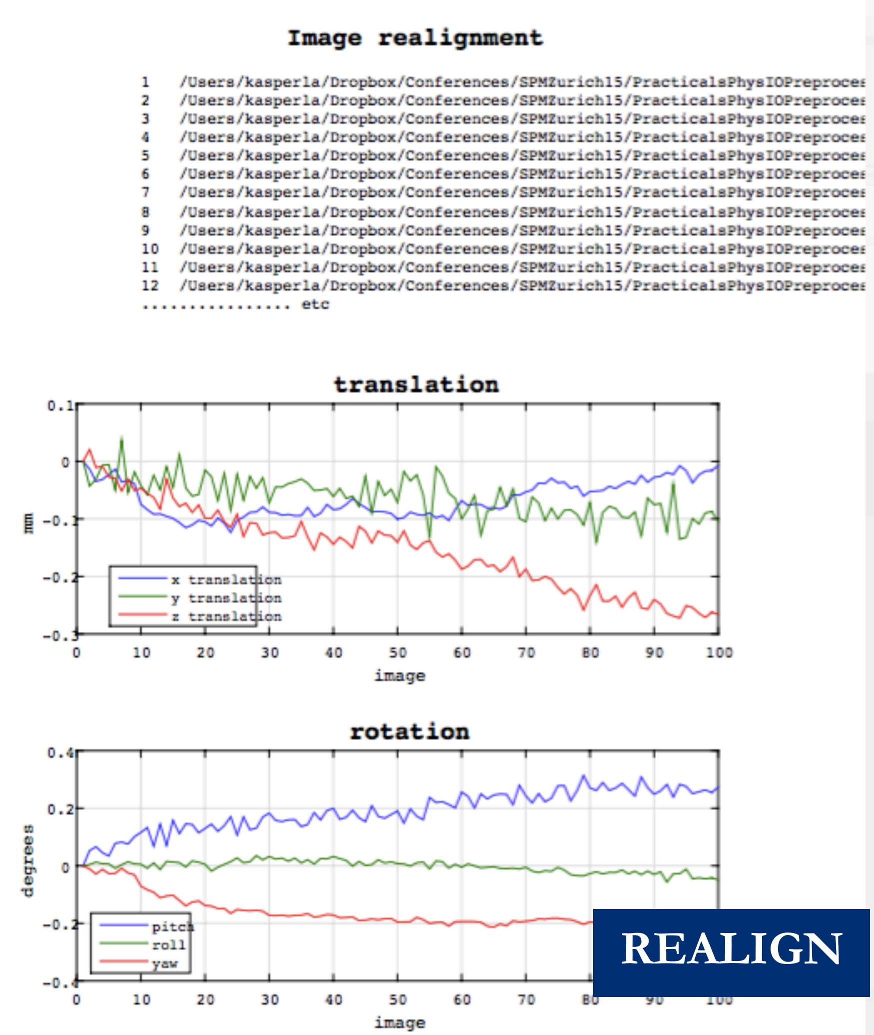
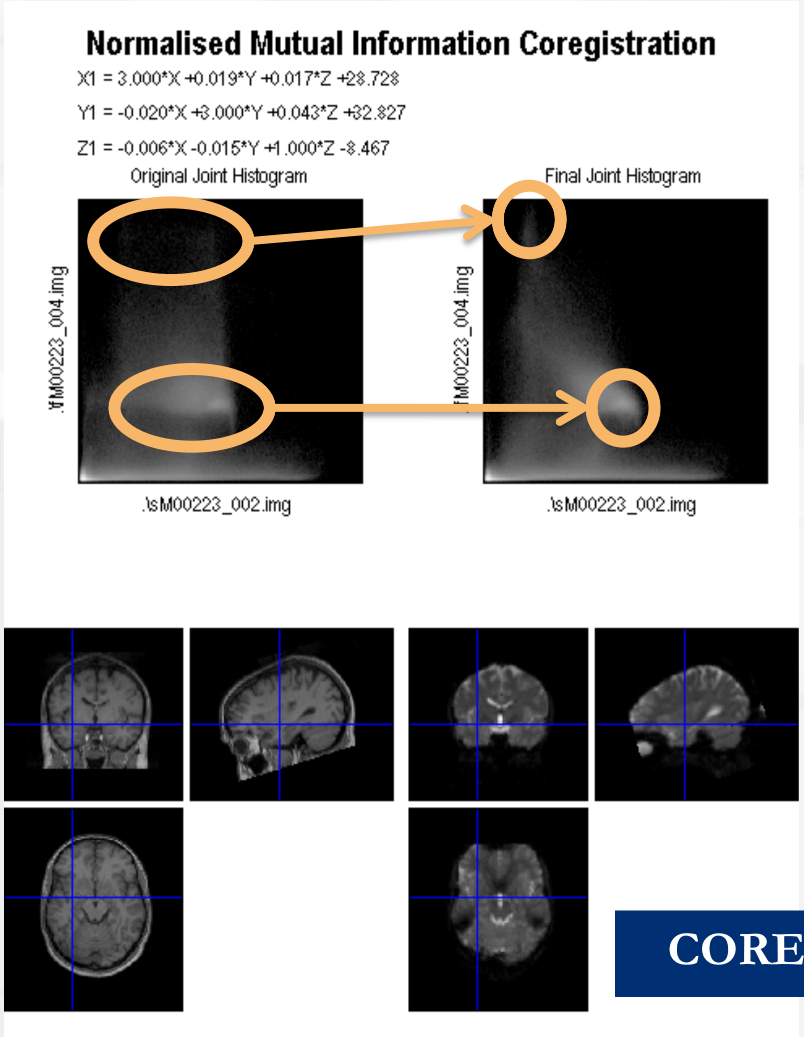


Image realignment

```
1    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
2    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
3    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
4    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
5    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
6    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
7    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
8    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
9    /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
10   /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
11   /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
12   /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreproces
................ etc
```

REALIGN



Normalised Mutual Information Coregistration

X1 = 3.000*X +0.019*Y +0.017*Z +28.728

Y1 = -0.020*X +3.000*Y +0.043*Z +32.827

Z1 = -0.006*X -0.015*Y +1.000*Z -8.467

COREG

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

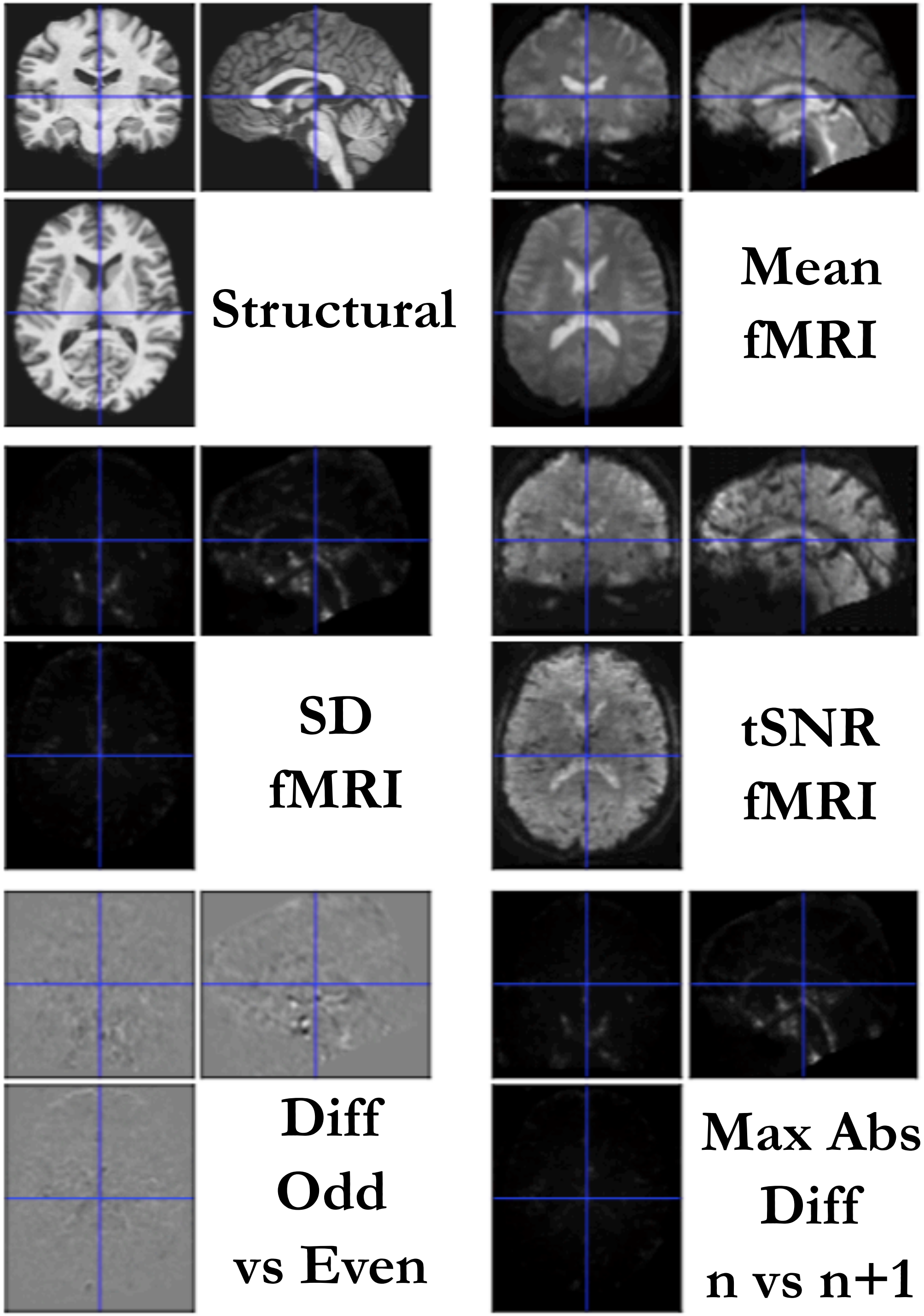
Pipeline Monitoring

- 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

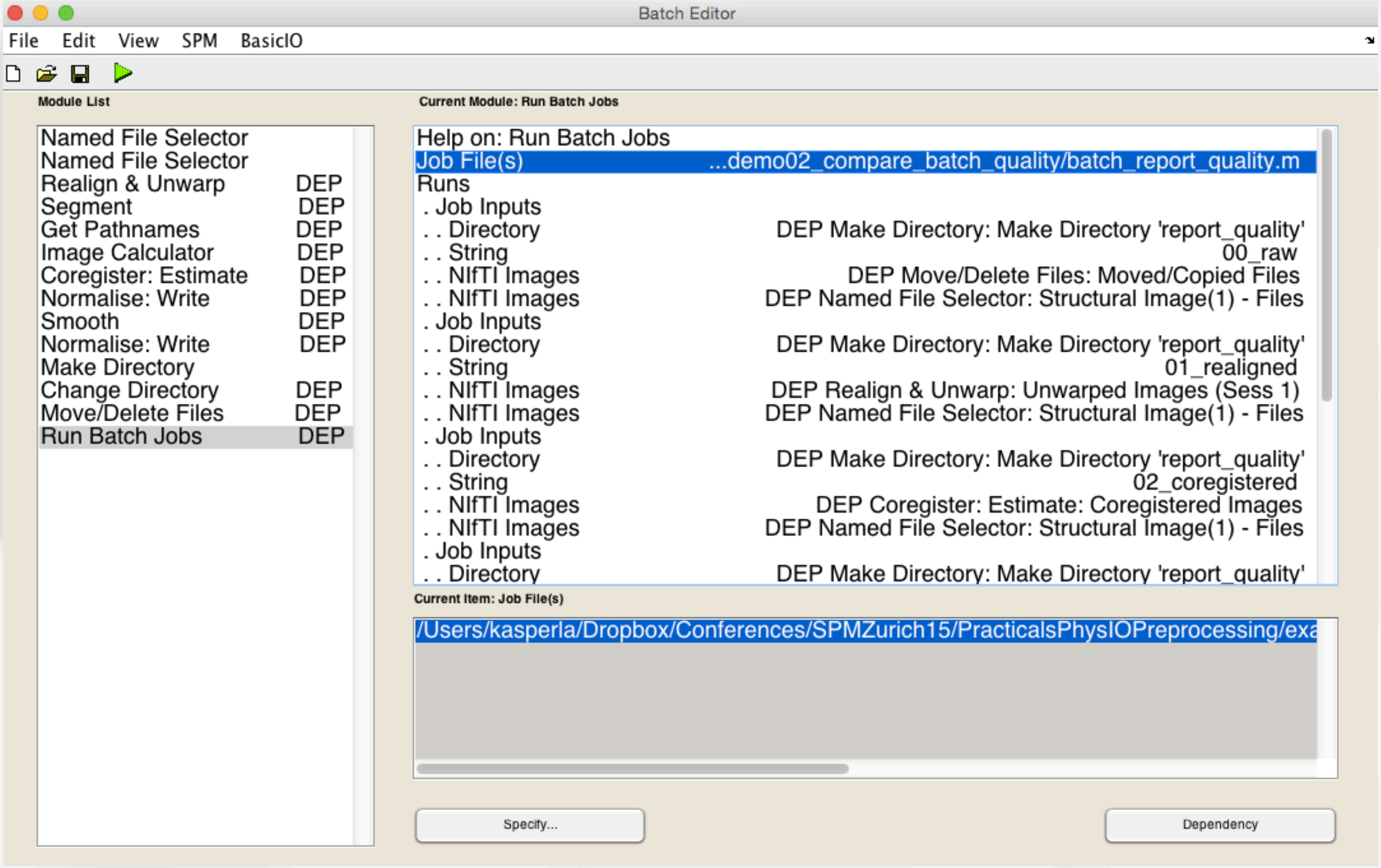
- Structural Image:
  visual inspection geometry/ contrast/noise/SNR
- Functional Time Series:
  - Mean => Artifact levels (localization)
  - SD => Fluctuation levels
  - tSNR = Mean/SD
    => sensitivity for BOLD signal changes
  - Diff = max(abs(diff)) or odd – even
    => outlier detection, image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



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



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

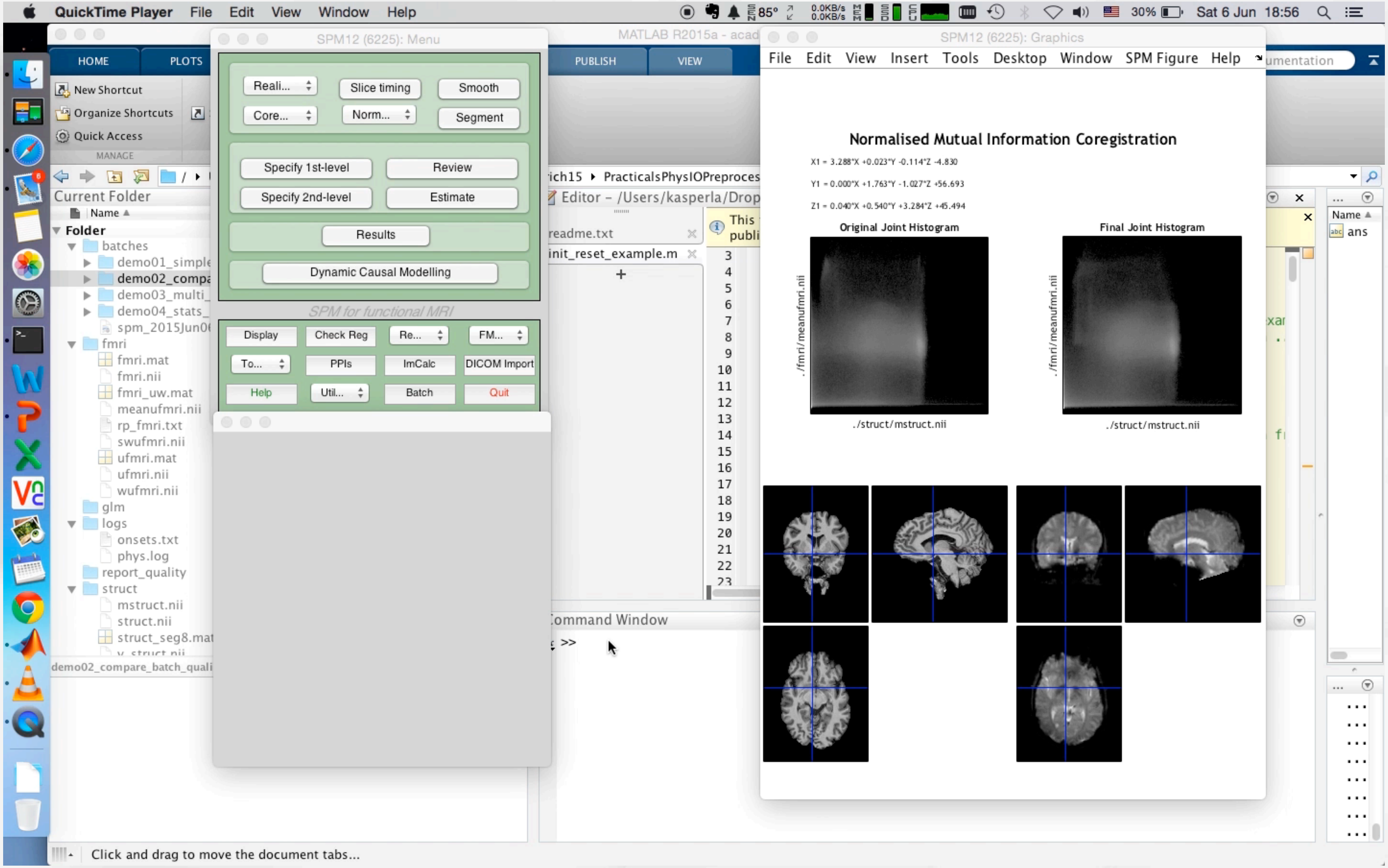
Help on: Run Batch Jobs	
Job File(s)	...demo02_compare_batch_quality/batch_report_quality.m
Runs	
. Job Inputs	
. . Directory	DEP Make Directory: Make Directory 'report_quality'
. . String	00_raw
. . NIfTI Images	DEP Move/Delete Files: Moved/Copied Files
. . NIfTI Images	DEP Named File Selector: Structural 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

Demo 2 - Video

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling



Normalised Mutual Information Coregistration

```
X1 = 3.288*X +0.023*Y -0.114*Z -4.830
Y1 = 0.000*X +1.763*Y -1.027*Z +56.693
Z1 = 0.040*X +0.540*Y +3.284*Z +45.494
```

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

SPM Plotting: CheckReg-Magic

- Recommended Plotting: Check Reg
  - via Batch Editor: SPM => Util => Check Reg.
  - Matlab command line:

    ```
    spm_check_registration('img1.nii',
    'img2.nii', …)
    ```

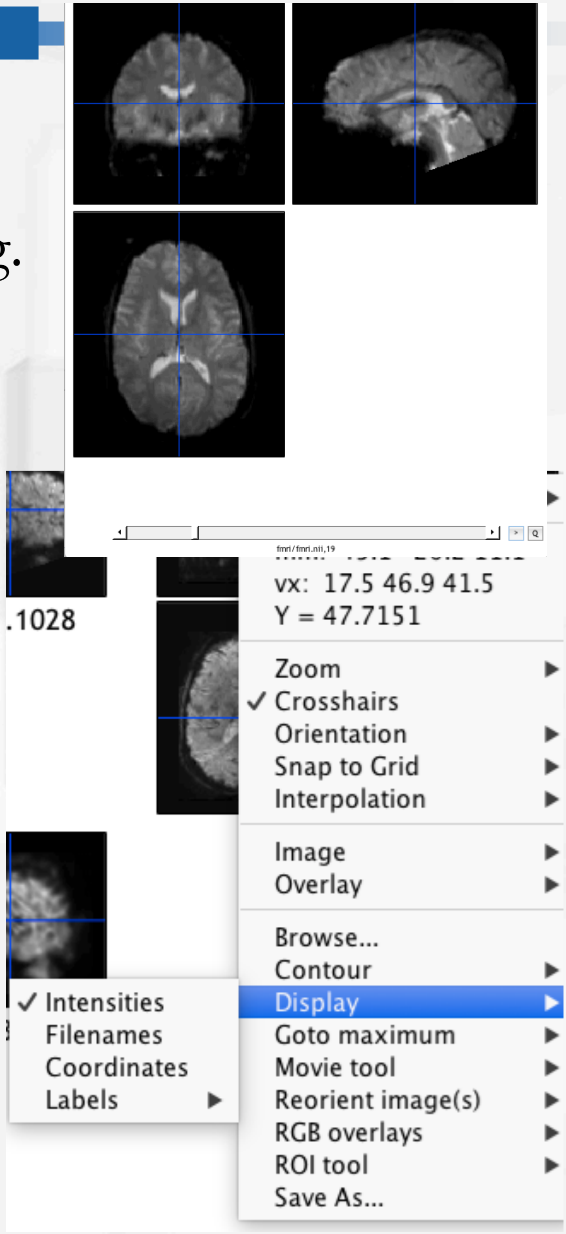
  - 4D NIFTI files:
    - show individual image tiles:

      ```
      spm_check_registration('fmri.nii,1',
      'fmri.nii,2', …)
      ```

    - show movie (**NEW**! SPM12):

      ```
      spm_check_registration('fmri.nii')
      ```

  - **Right click** reveals neat features (add blobs, edges, anatomic labels, header info, contrast)



vx: 17.5 46.9 41.5
Y = 47.7151

Plotting Example

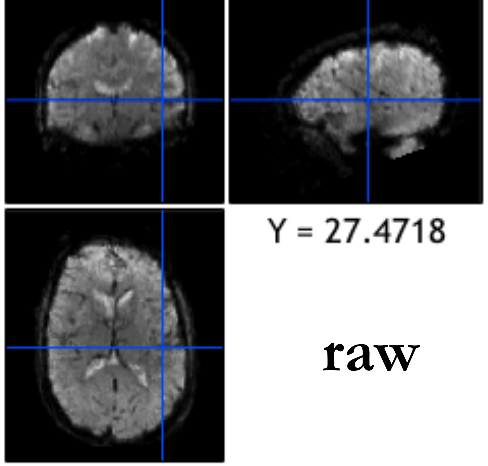
- Example: Comparing temporal SNR throughout preprocessing
  - Temporal SNR per pixel in functional image time series after each preprocessing step
  - Command line code:

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

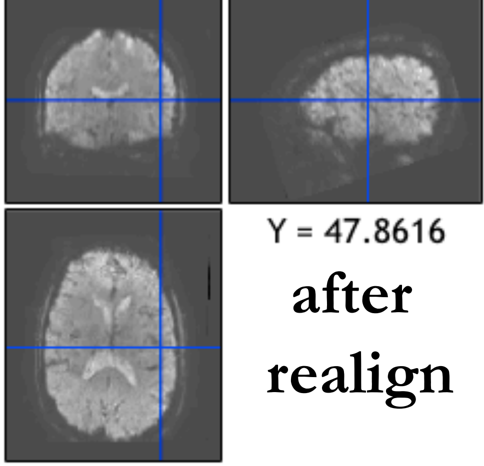
Plotting Example

Setup Pipeline | Quality Monitoring

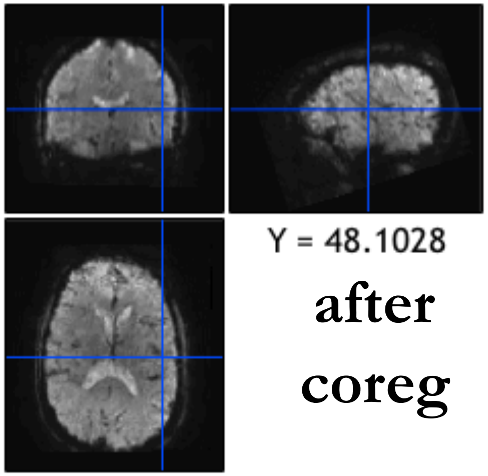
- Example: Temporal SNR per pixel in functional image time series after each preprocessing step
- Right click reveals amazing features
- Result: Increased SNR through realignment and smoothing



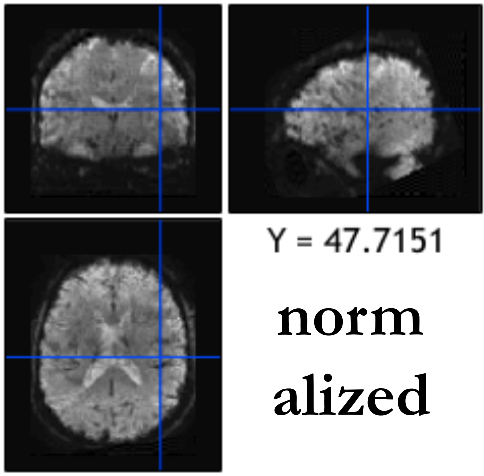
raw



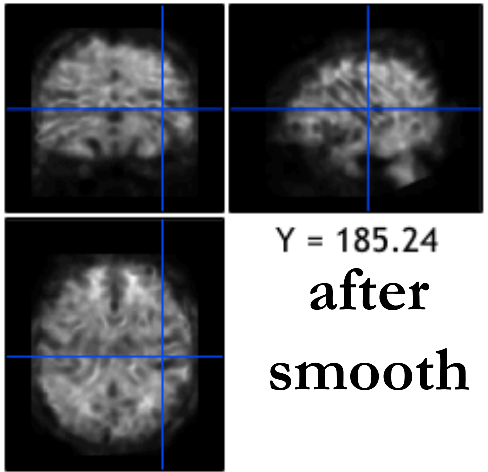
after realign



after coreg



norm alized

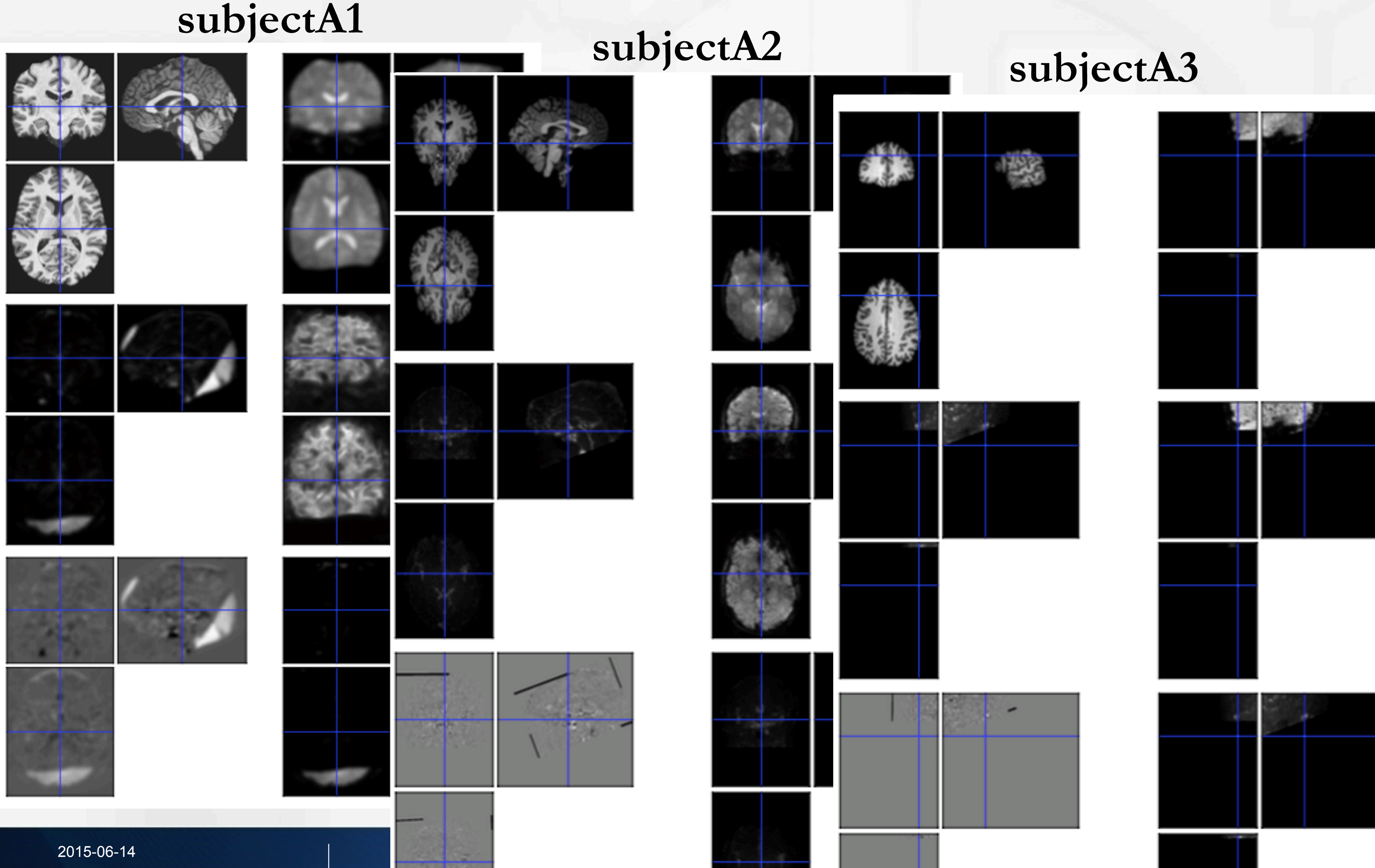


after smooth

Spotting Common Pitfalls

- 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

Spotting Common Pitfalls



Spotting Common Pitfalls

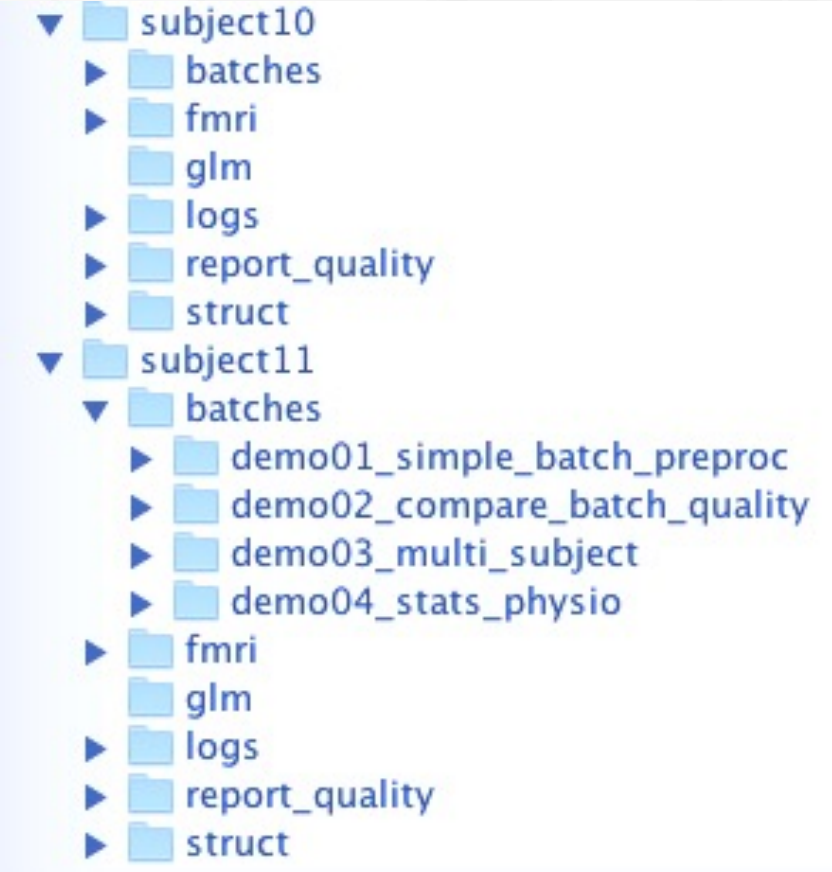
- Task: Spotting unusual quality report images
- What went wrong here?
    - subjectA1
        - realignment failed
        - one volume rotated (30 degrees around x-axis)
    - subjectA2
        - segmentation failed
        - structural mirrored compared to template
    - subjectA3
        - co-registration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

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

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

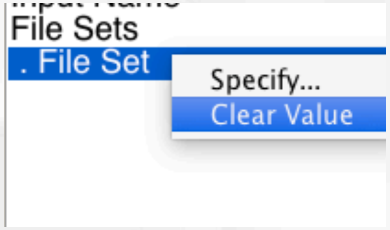


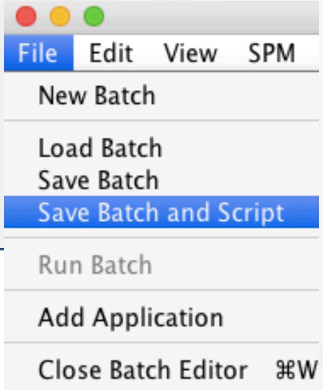
- subject10
  - batches
  - fmri
  - glm
  - logs
  - report_quality
  - struct
- subject11
  - batches
    - demo01_simple_batch_preproc
    - demo02_compare_batch_quality
    - demo03_multi_subject
    - demo04_stats_physio
  - fmri
  - glm
  - logs
  - report_quality
  - struct

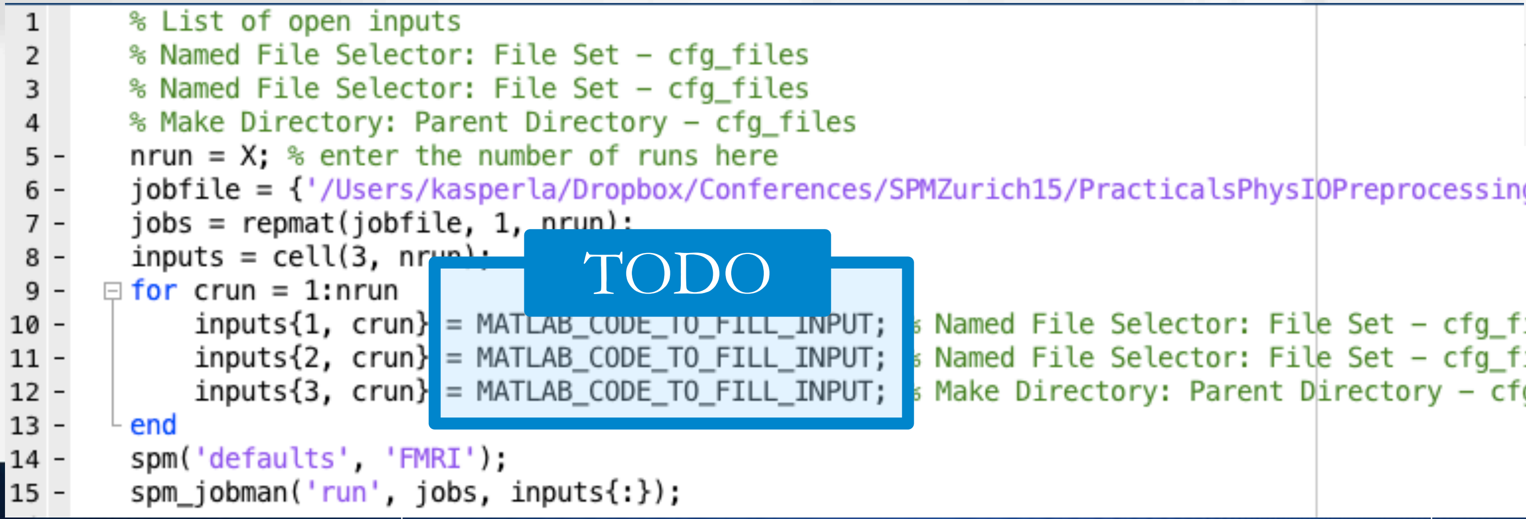
- batches/ => subject batch with data spec
- fmri/ => functional data per session
- struct/ => structural data
- glm/ => statistical analyses
- logs/ => behavioural response files, physiological recordings

Looping over subjects

1. Test and optimise batch for 1 subject
2. Clear subject-specific data inputs from batch (right click), save as template job
3. Save Batch and Script
   - creates _job.m-file: batch w/o data
   - creates .m-file: Matlab script w/ multi-subject loop



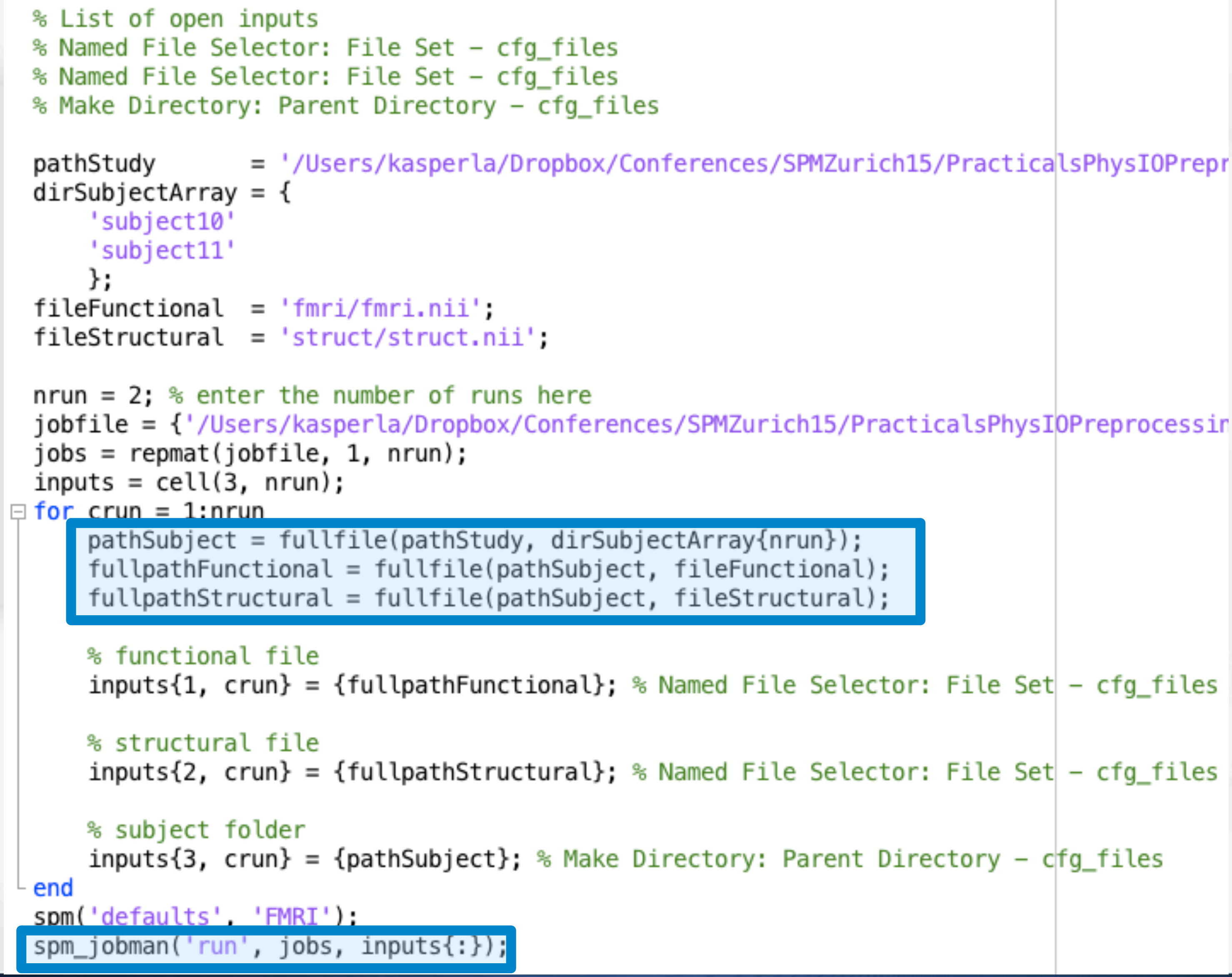




```
% List of open inputs
% Named File Selector: File Set - cfg_files
% Named File Selector: File Set - cfg_files
% Make Directory: Parent Directory - cfg_files
nrun = X; % enter the number of runs here
jobfile = {'/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreprocessing/example_physio_
jobs = repmat(jobfile, 1, nrun);
inputs = cell(3, nrun);
for crun = 1:nrun
    inputs{1, crun} = MATLAB_CODE_TO_FILL_INPUT; % Named File Selector: File Set - cfg_files
    inputs{2, crun} = MATLAB_CODE_TO_FILL_INPUT; % Named File Selector: File Set - cfg_files
    inputs{3, crun} = MATLAB_CODE_TO_FILL_INPUT; % Make Directory: Parent Directory - cfg_files
end
spm('defaults', 'FMRI');
spm_jobman('run', jobs, inputs{:});
```

TODO

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/PracticalsPhysIOPreprocessin
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{:});
```

Multi-subject Quality Monitoring

- Concatenate `spm_<date>.ps` report files
- Use spm_check_registration to plot same statistics for different subjects
  - same way as for quality monitoring, after different preprocessing steps
  - e.g. did SNR image before smoothing change from subject to subject => possible coil failure

Demo 3

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling

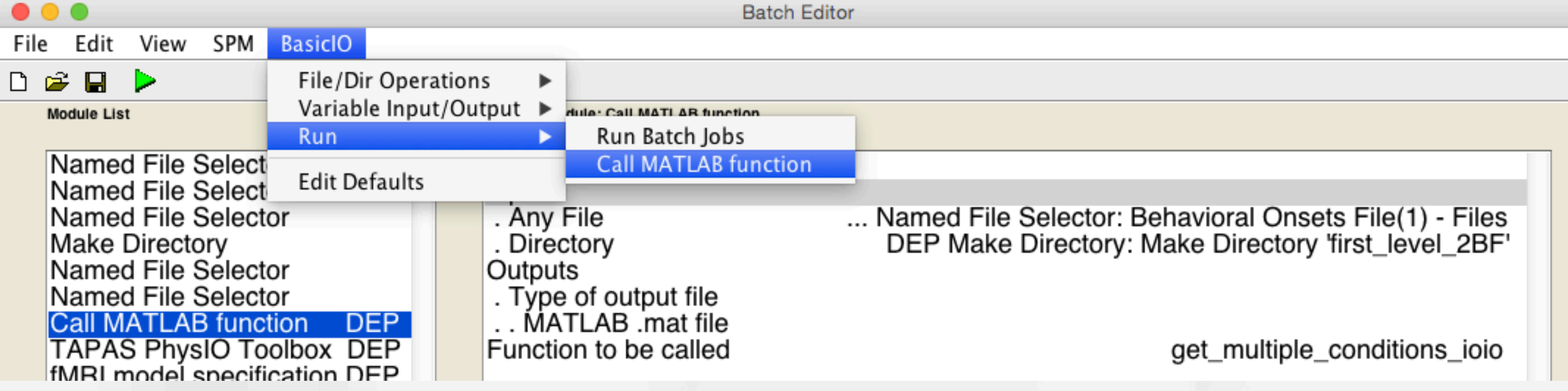
- Performs Demo 2 (complete preproc + quality report) for multiple subjects (subject10 and subject11)
  - create subject folders: Run `code/create_multi_subject_data.m`
- Run via
  1. Load "job": `subject10/batches/demo03_multi_subject/batch_preproc_fmri_report_quality_job.m` - Inspect!
  2. Load & Run `batch_preproc_fmri_report_quality_gui.m`
     - in same folder; use *Batch* Editor to inspect
  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

- Call Matlab script/function within Batch Editor
  - BasicIO => Call Matlab Function
  - Specify number and type of input parameters (also dependencies)
- Use, if own code must run at certain point in pipeline
  - e.g. extra realignment parameters 12/24
  - Otherwise: Matlab Editor/Command Window more efficient



Named File Select
Named File Select
Named File Selector
Make Directory
Named File Selector
Named File Selector
Call MATLAB function DEP
TAPAS PhysIO Toolbox DEP
fMRI model specification DEP

. Any File ... Named File Selector: Behavioral Onsets File(1) - Files
. Directory DEP Make Directory: Make Directory 'first_level_2BF'
Outputs
. Type of output file
. . MATLAB .mat file
Function to be called get_multiple_conditions_ioio

Physiological Noise Correction

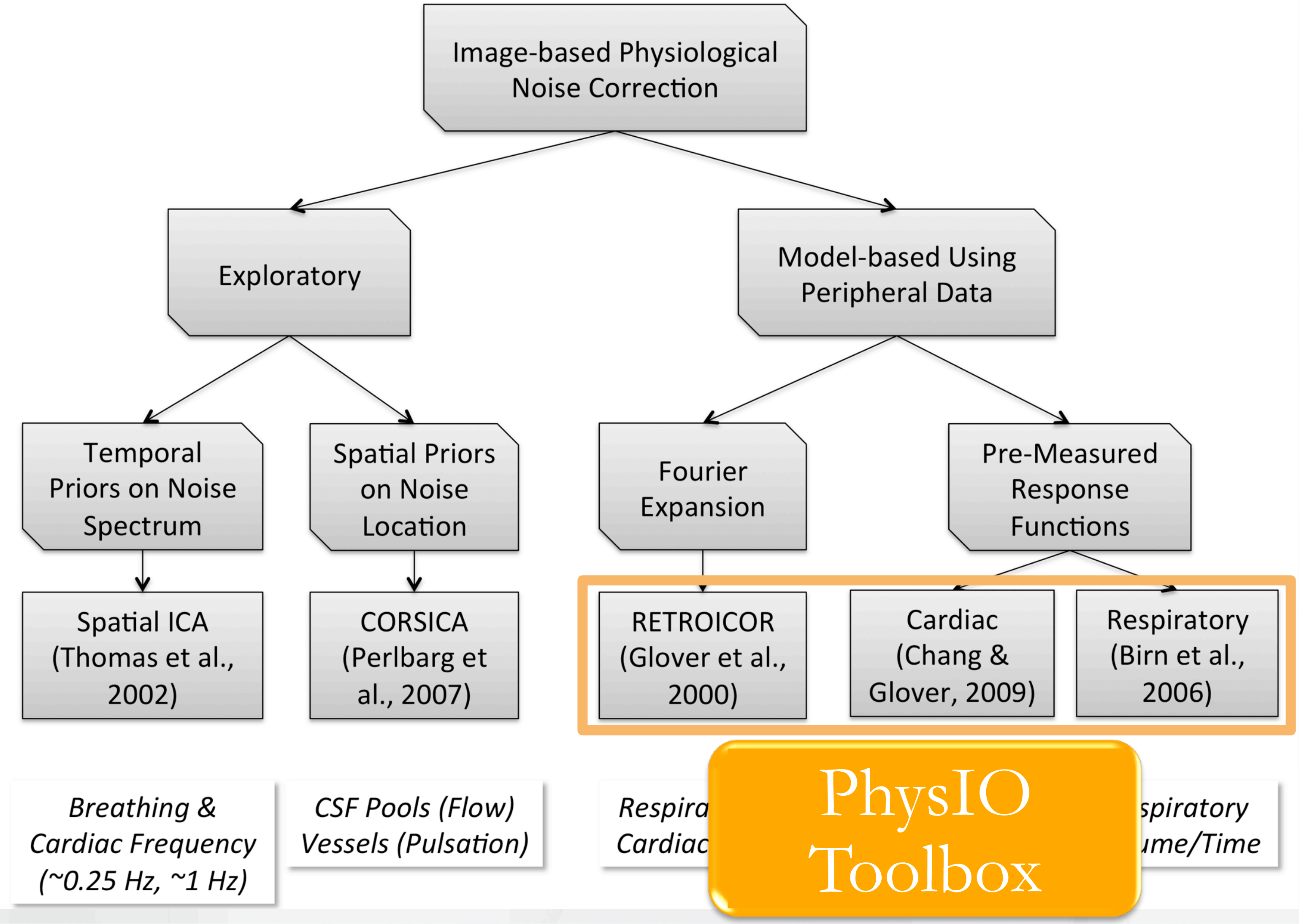


Image-based Physiological Noise Correction

Exploratory

Model-based Using Peripheral Data

Temporal Priors on Noise Spectrum

Spatial Priors on Noise Location

Fourier Expansion

Pre-Measured Response Functions

Spatial ICA (Thomas et al., 2002)

CORSICA (Perlbarg et al., 2007)

RETROICOR (Glover et al., 2000)

Cardiac (Chang & Glover, 2009)

Respiratory (Birn et al., 2006)

Breathing & Cardiac Frequency (~0.25 Hz, ~1 Hz)

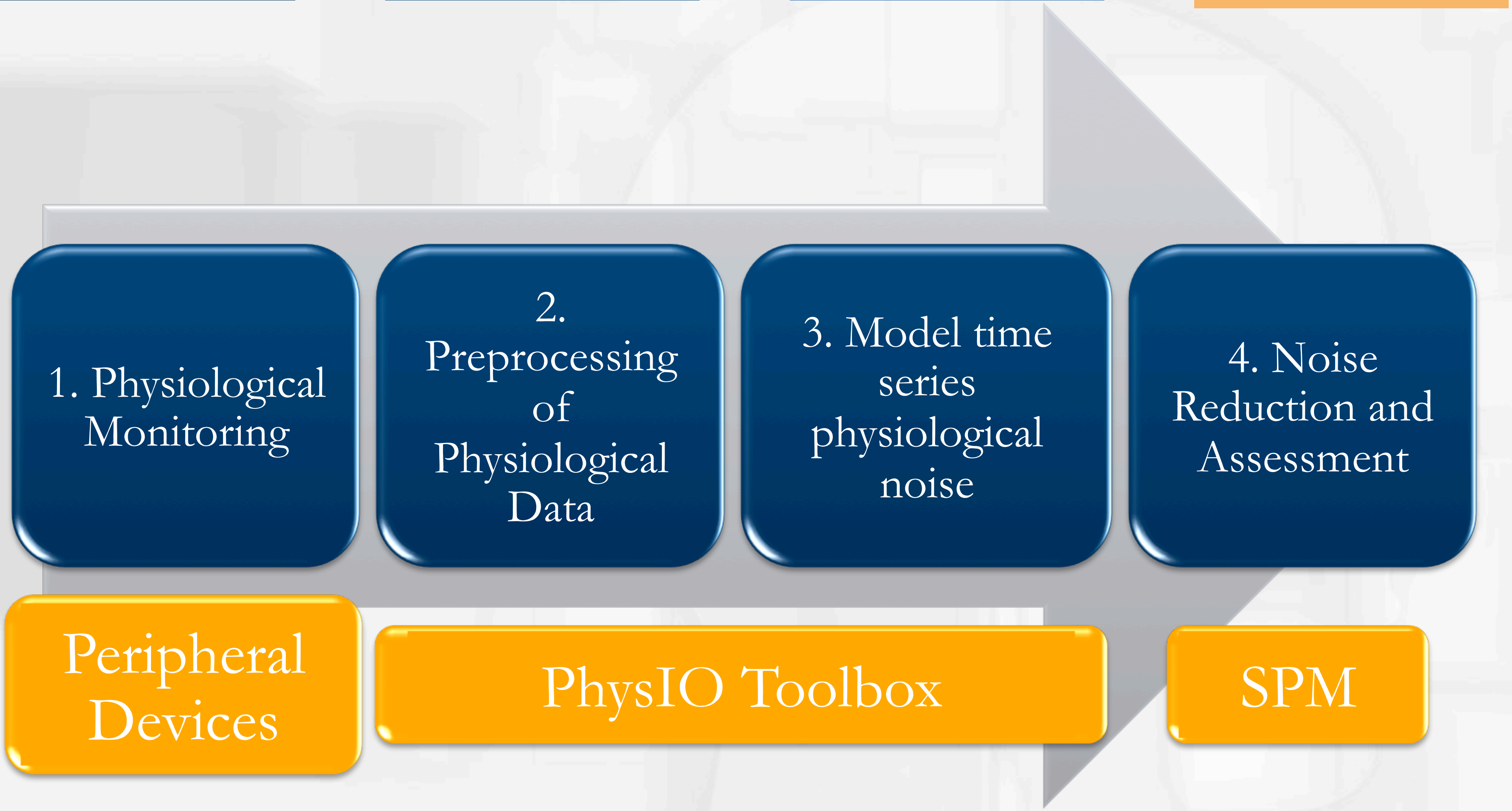
CSF Pools (Flow) Vessels (Pulsation)

Respira Cardiac

spiratory ume/Time

PhysIO Toolbox

TAPAS PhysIO Toolbox



1. Physiological Monitoring

2. Preprocessing of Physiological Data

3. Model time series physiological noise

4. Noise Reduction and Assessment

Peripheral Devices

PhysIO Toolbox

SPM

ECG, PPU → Cardiac cycle

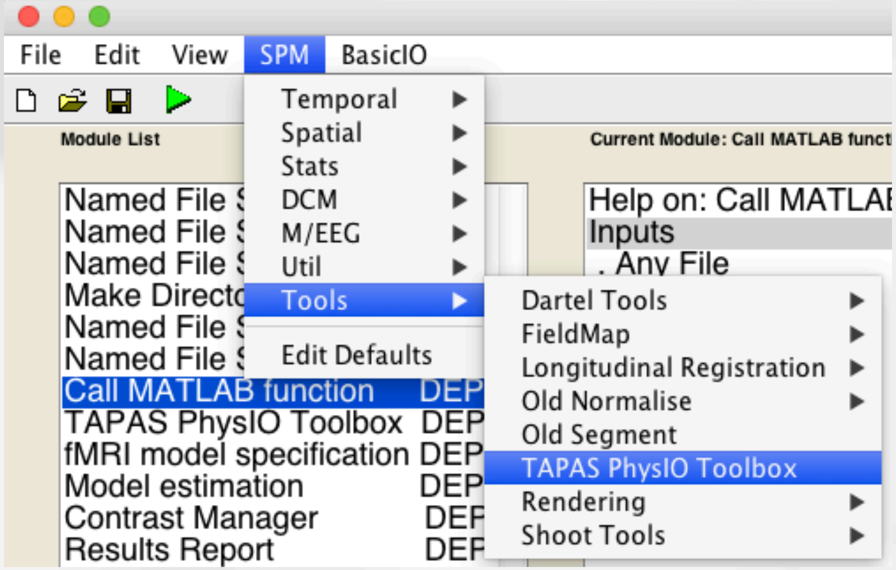
Breathing belt → Respiratory cycle

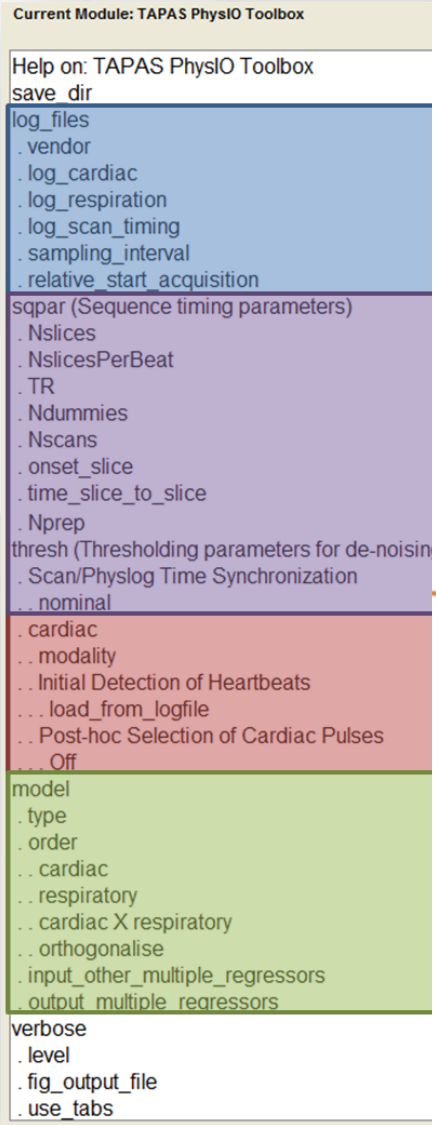
Confound regressors

PhysIO: SPM Batch Interface

Setup Pipeline | Quality Monitoring | Multi-Subject

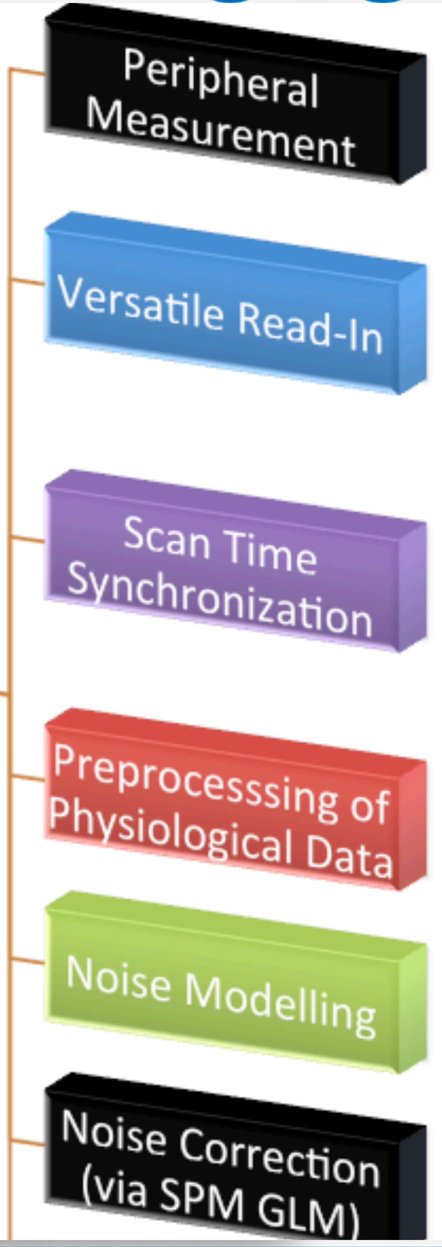
- Copy `PhysIO/code/*.m` to `spm12/toolbox/physio` to appear in Batch Editor
  - Menu SPM => Tools
  - works with dependencies etc.





Current Module: TAPAS PhysIO Toolbox

```
Help on: TAPAS PhysIO Toolbox
save_dir
log_files
. vendor
. log_cardiac
. log_respiration
. log_scan_timing
. sampling_interval
. relative_start_acquisition
sqpar (Sequence timing parameters)
. Nslices
. NslicesPerBeat
. TR
. Ndummies
. Nscans
. onset_slice
. time_slice_to_slice
. Nprep
thresh (Thresholding parameters for de-noisin
. Scan/Physlog Time Synchronization
. . nominal
. cardiac
. . modality
. . Initial Detection of Heartbeats
. . . load_from_logfile
. . Post-hoc Selection of Cardiac Pulses
. . . Off
model
. type
. order
. . cardiac
. . respiratory
. . cardiac X respiratory
. . orthogonalise
. input_other_multiple_regressors
. output_multiple_regressors
verbose
. level
. fig_output_file
. use_tabs
```



Peripheral Measurement

Versatile Read-In

Scan Time Synchronization

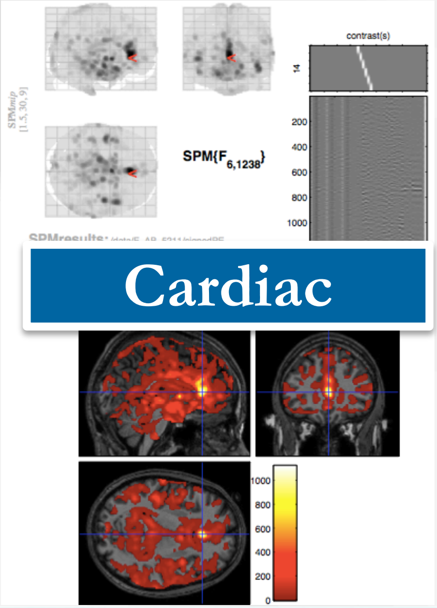
Preprocessing of Physiological Data

Noise Modelling

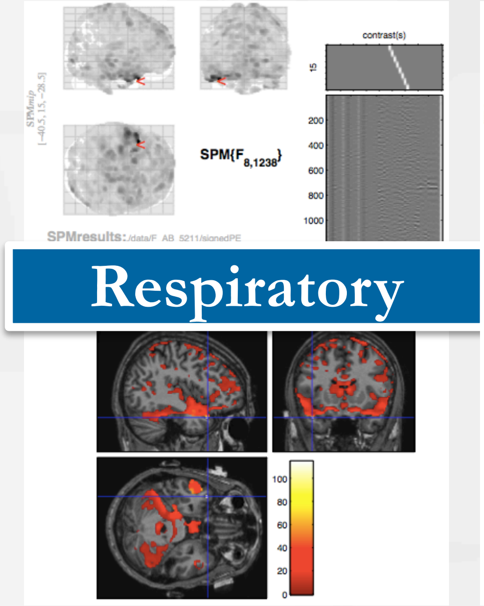
Noise Correction (via SPM GLM)

Pipeline Noise Modeling/Report

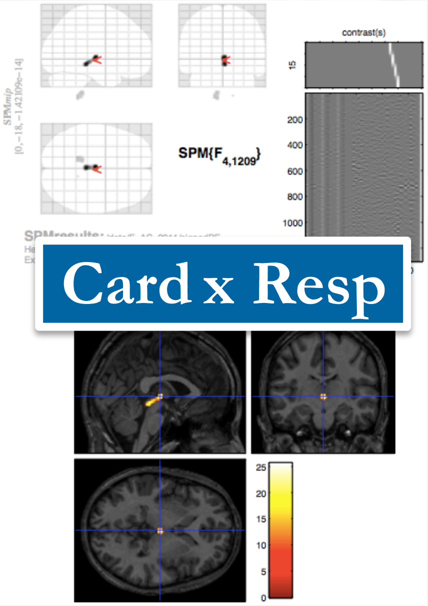
- Model physiological noise as nuisance regressors
- Check noise correction via explained variance (F-contrasts)
  - Automatic contrast creation and report (glass brain): Batch Editor SPM => Stats => Contrast Manager/Results Report
  - With underlays: `tapas_physio_report_contrasts`



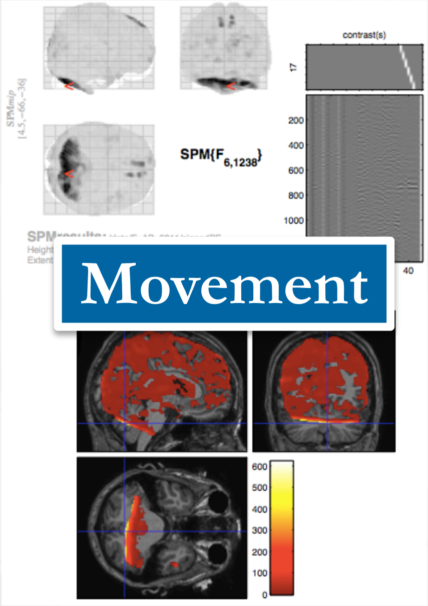
Cardiac



Respiratory



Card x Resp



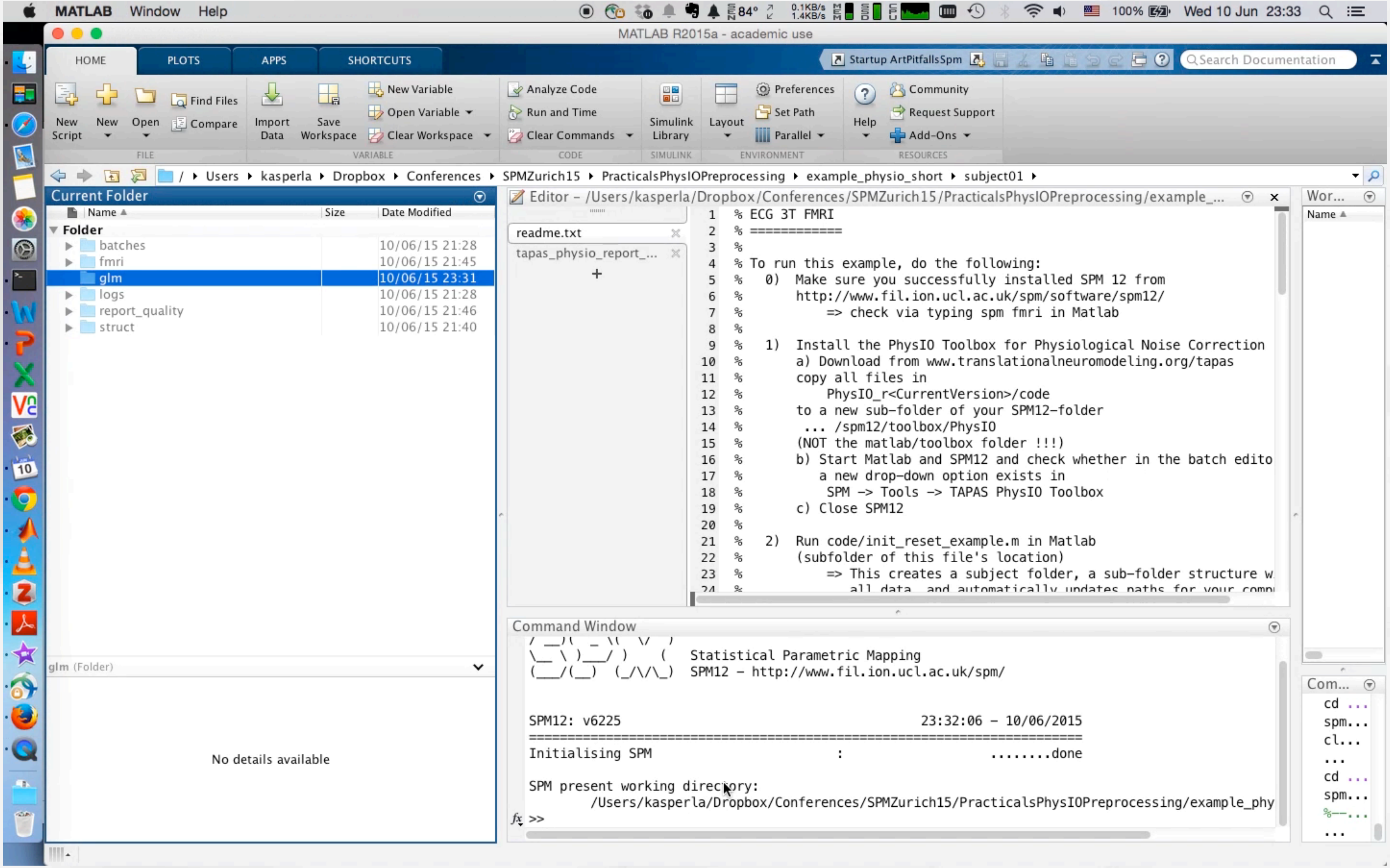
Movement

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 -Video

Setup Pipeline | Quality Monitoring | Multi-Subject | **Noise Modeling**



Name	Size	Date Modified
Folder		
batches		10/06/15 21:28
fmri		10/06/15 21:45
glm		10/06/15 23:31
logs		10/06/15 21:28
report_quality		10/06/15 21:46
struct		10/06/15 21:40

```
% ECG 3T FMRI
% ============
%
% To run this example, do the following:
%    0)  Make sure you successfully installed SPM 12 from
%        http://www.fil.ion.ucl.ac.uk/spm/software/spm12/
%            => check via typing spm fmri in Matlab
%
%    1)  Install the PhysIO Toolbox for Physiological Noise Correction
%        a) Download from www.translationalneuromodeling.org/tapas
%        copy all files in
%            PhysIO_r<CurrentVersion>/code
%        to a new sub-folder of your SPM12-folder
%         ... /spm12/toolbox/PhysIO
%        (NOT the matlab/toolbox folder !!!)
%        b) Start Matlab and SPM12 and check whether in the batch edito
%           a new drop-down option exists in
%            SPM -> Tools -> TAPAS PhysIO Toolbox
%        c) Close SPM12
%
%    2)  Run code/init_reset_example.m in Matlab
%        (subfolder of this file's location)
%            => This creates a subject folder, a sub-folder structure w
```

```
 / __)(  _ \(  \/  )
 \__ \ )___/ )    (   Statistical Parametric Mapping
 (___/(__)  (_/\/\_)  SPM12 - http://www.fil.ion.ucl.ac.uk/spm/


SPM12: v6225                                  23:32:06 - 10/06/2015
========================================================================
Initialising SPM                     :                        ........done

SPM present working directory:
        /Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreprocessing/example_phy
>>
```

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 and Matlab script (subj.-loop)

- Integrating Own Code & Physiological Noise Correction

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

Thank you for your attention!

- …and these people for their support in the preparation:
  - Ged Ridgway
  - Guillaume Flandin
  - Klaas Enno Stephan
  - John Ashburner
  - Andreea Diaconescu
  - Steffen Bollmann
- Questions:
  - Now ☺
  - Poster #3736 (PhysIO): Wed, 12:45pm
  - `kasper@biomed.ee.ethz.ch`

Outline

- **Setting up a Preprocessing Pipeline in SPM: The Batch Editor**
  - SPM Vocabulary
    - What is SPM? GUI
    - What is Realign/Co-Registration/Normalisation via Unified Segmentation
    - Dealing with Headers (Estimate vs Write/Reslice), 3D vs 4D frames
  - Preprocessing Theory (SPM-specific)
    - Co-Registration Joint Histogram
    - Unified Segmentation: Algorithm, TPMs, Why does it "normalise"?
  - Batch Editor Philosophy: Modules/Dependencies/Data
    - History, V. Glauche, spm/batches (Ged), `preproc_fmri_simplified.m`
    - Finally: Process Diagram and corresponding pipeline, run!

Outline

- **Monitoring and Comparing Pipelines**
  - Performance Measures: Mean/SD/SNR/Diff Images
    - Rationale from time series, partial volume effects
    - mean = artifact, SD = sensitivity loss, SNR = combined, mean(abs(diff))
  - SPM Plotting Routines and Automatic Reporting
    - spm_print, spm_check_registration
  - Spotting Failed Pipelines
    - fmri.mat with 1 vol (or spike in FFT?), anatomy mirrored for coreg/segment
  - Comparing Alternative Pipelines
    - slice timing correction no/before/after realign: `preproc_fmri.m`
    - deep or broad folder tree for different pipelines?

Outline

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling

- **Multi-subject Pipelines**
  - Staying on Top: Organisation of Data
    - Same folder structure for different subjects, suggested naming
  - Looping Pipelines over Subjects' Data
    - Unfilled job dependencies (<-X) as input parameters
    - Skipping subjects
    - No GUI for cluster (spm_get_defaults('cmdline',1)) => howto output?
  - Tips for Efficient Performance Monitoring
    - Compiling large PS files with abundant plotting
    - Scripts for re-loading quality report images (`spm_check_registration`)

Outline

- **Integrating Own Code and Toolboxes:**

  Physiological Noise Modeling and Evaluation
  - Executing Custom Matlab Code within the Pipeline
    - e.g. creation of multiple_conditions, extra motion regressors (12/24)
  - The TAPAS PhysIO Toolbox
    - Overview which noise models implemented
    - Focus on robustness, multi-vendor compatibility
    - Modules/Parameters colored and in Batch Editor
  - Automatic (Noise) Modeling and Contrast Reporting
    - Contrast Report => Glass Brain & Table as .ps
    - tapas_physio_report_contrasts => F-contrast, showing typical distributions and structural underlay

Preprocessing: Image Registration

Setup Pipeline | Quality Monitoring | Multi-Subject | Noise Modeling

B. Allowed Transformations

Rigid-Body	Affine	Non-linear	
REALIGN	**COREG**	**SEGMENT**	**NORM WRITE**

C. Similarity Measure

Mean-squared Difference	Mutual Information	Tissue Class Probability

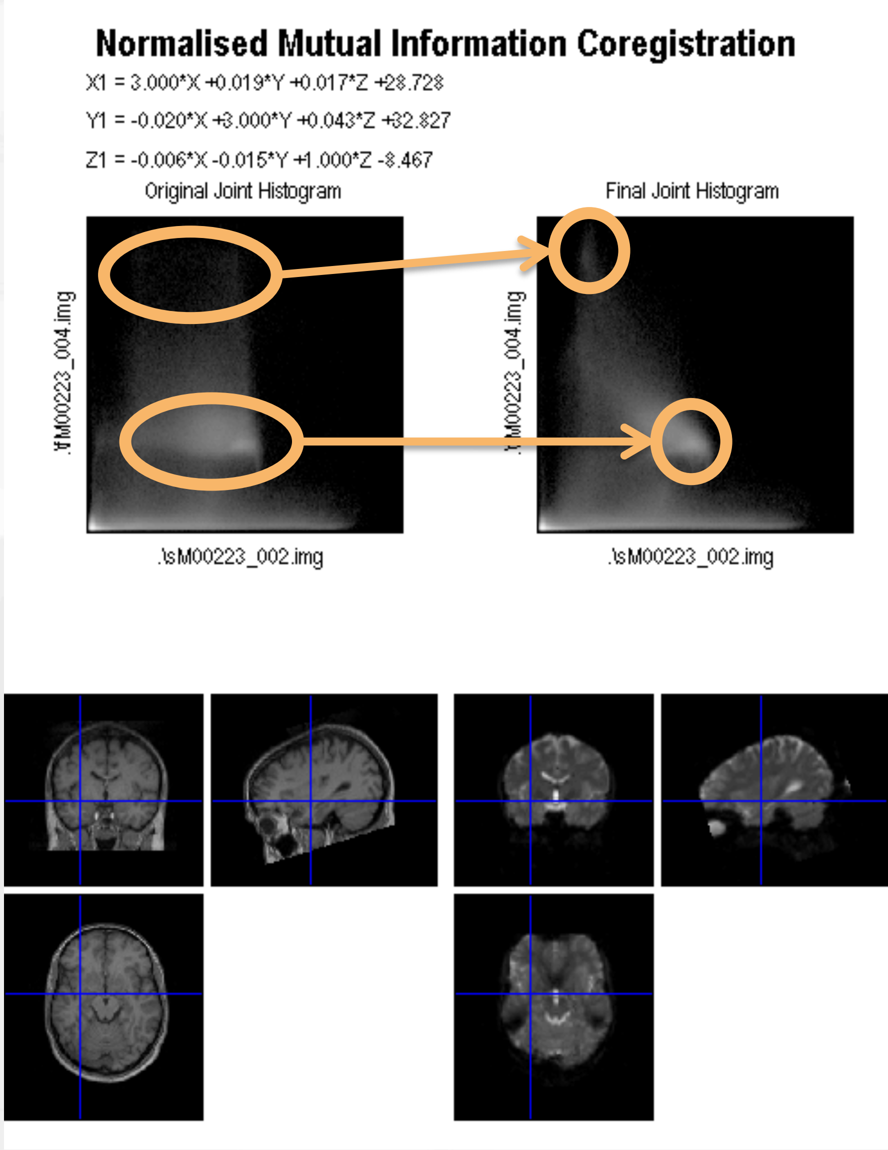
D. Optimisation

Exact Linearized Solution	Conjugate Direction Line Search	Iterated Conditional Modes (EM/Levenberg-Marquardt)

Theory: Good Co-Registration Output

- Voxels of same tissue have same intensity in MR-contrast
- In a 2nd MR contrast, this intensity may be different, but still the same among all voxels of same tissue type

➡ Aligned voxels in 2 images = crisp peaks in joint histogram



Normalised Mutual Information Coregistration

```
X1 = 3.000*X +0.019*Y +0.017*Z +28.728
Y1 = -0.020*X +3.000*Y +0.043*Z +32.827
Z1 = -0.006*X -0.015*Y +1.000*Z -8.467
```

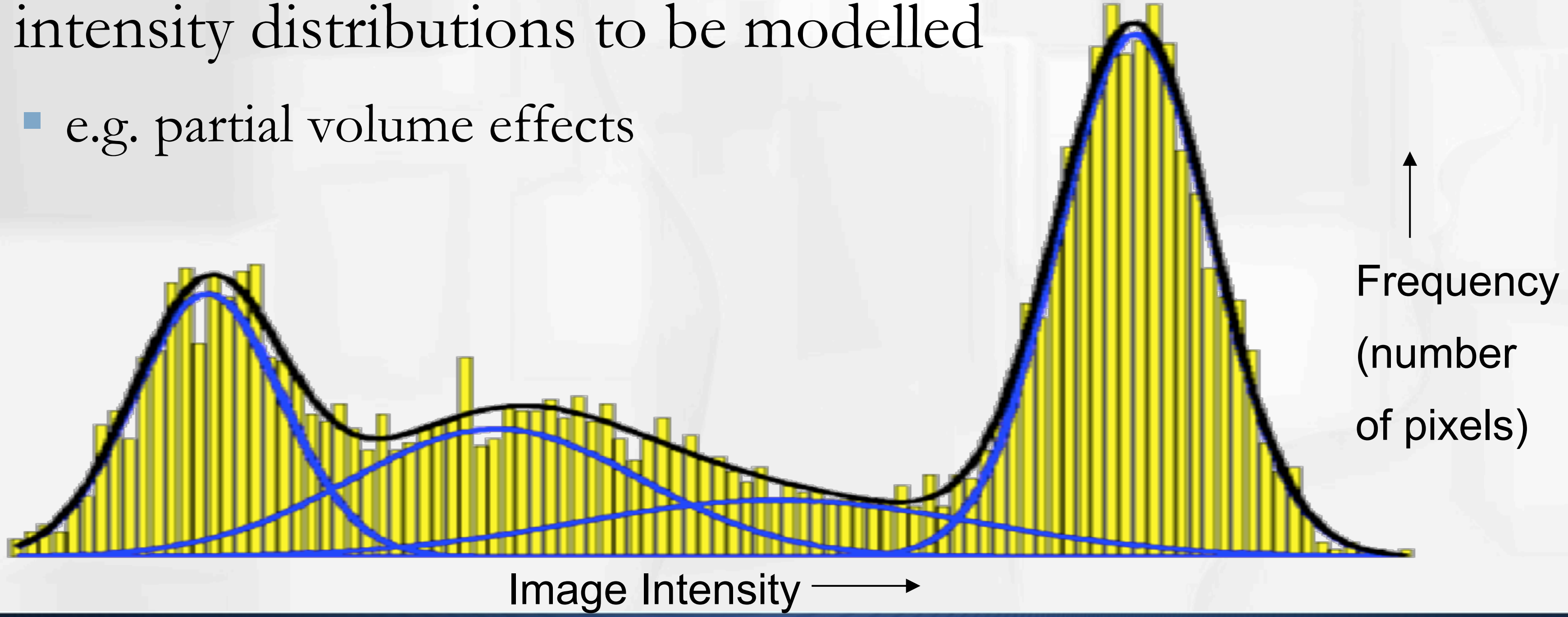
Theory: Unified Model Segmentation

- SPM12 implements a generative model[1] of voxel intensity from tissue class probabilities
    - Principled Bayesian probabilistic formulation
    - Segmentation by inverting a Gaussian mixture model
- Deformations of prior tissue probability maps (TPMs, priors) part of the model
    - The inverse transformation that aligns the TPMs normalises the structural image(s)
    - Non-linear deformations are constrained by regularisation factors
- Bias correction (coil inhomogeneity) within the model

[1] Ashburner & Friston (2005), Neuroimage

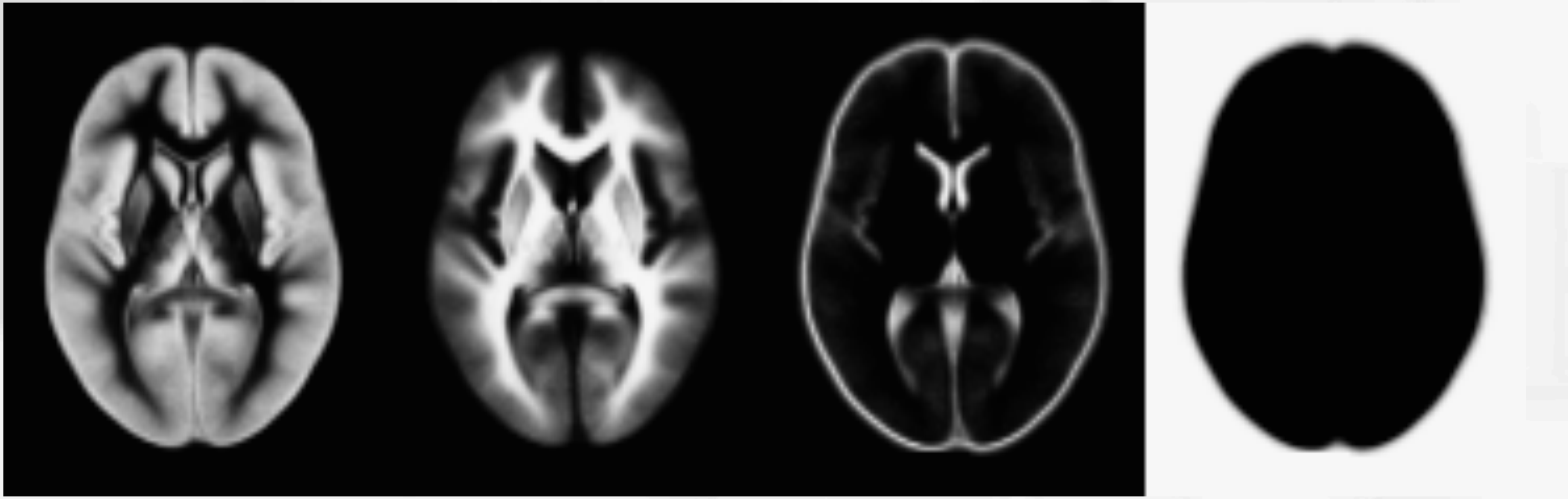
Mixture of Gaussians

- Classification is based on a Mixture of Gaussians model, which represents the intensity probability density by a number of Gaussian distributions.
- Multiple Gaussians per tissue class allow non-Gaussian intensity distributions to be modelled
  - e.g. partial volume effects



Tissue Probability Maps

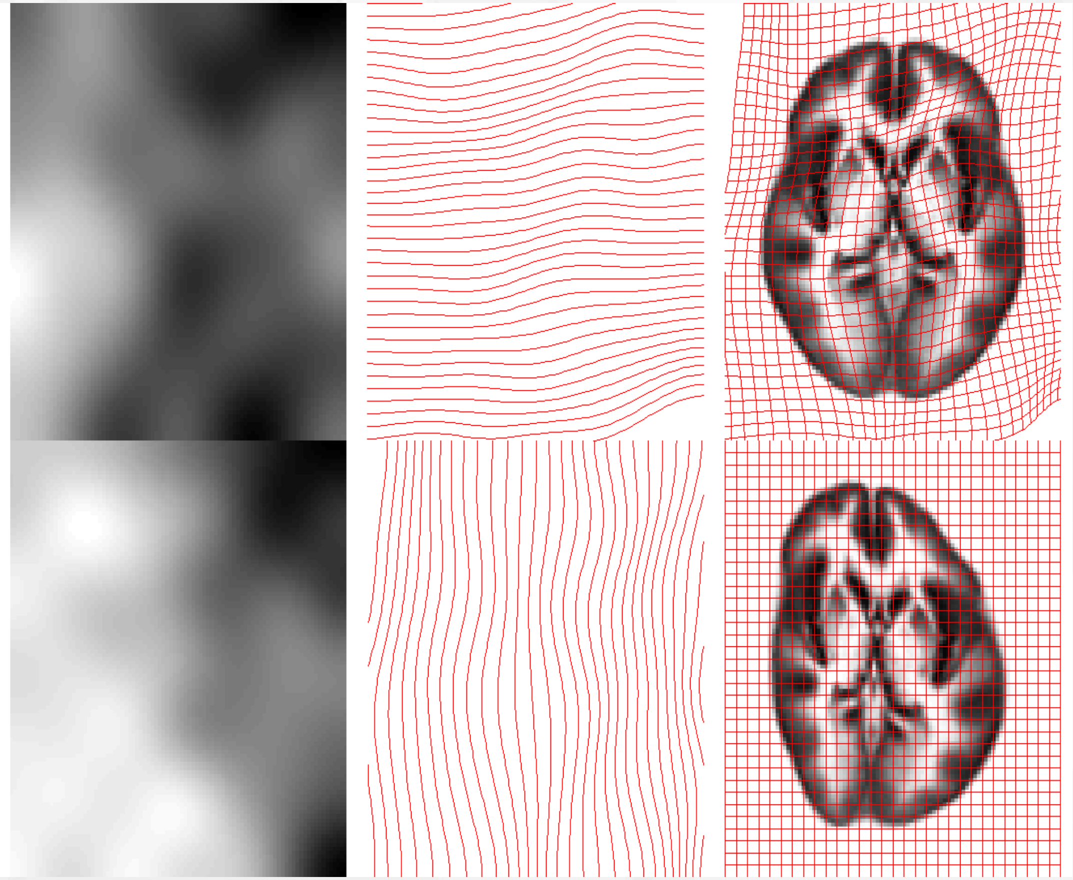
- Tissue probability maps (TPMs) are used as the prior, instead of the proportion of voxels in each class

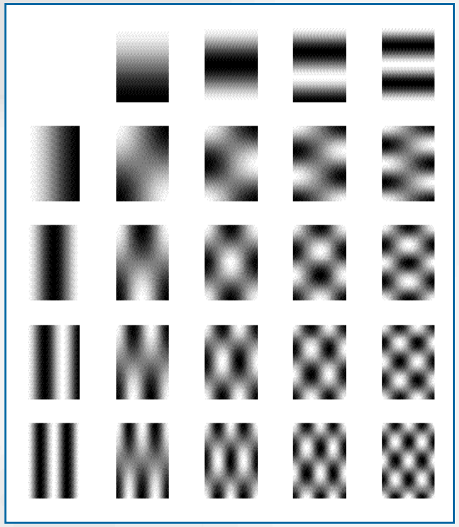


ICBM Tissue Probabilistic Atlases. These tissue probability maps were kindly provided by the **International Consortium for Brain Mapping**

Deforming the Tissue Probability Maps

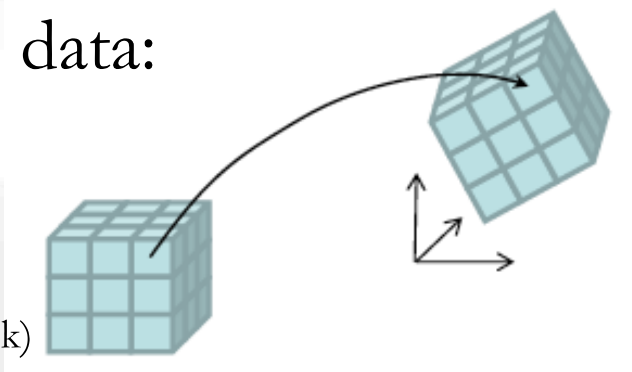
- Tissue probability map templates are warped to match the subject
- The inverse transform warps to the TPMs





Headers (Map Voxel-to-World)

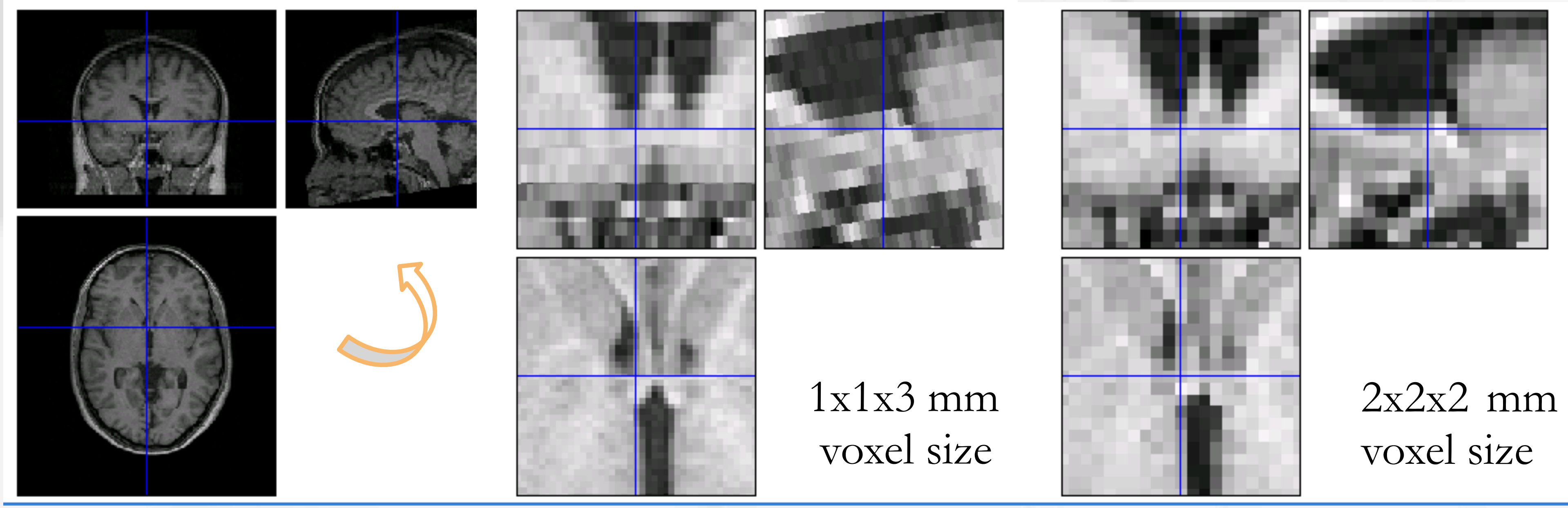
- Data:
  - 3D images are made up of voxels.
  - Voxel intensities are stored on disk as lists of numbers.
- Header: Meta-information about the data:
  - image dimensions (list to 3D)
  - "voxel-to-world mapping" (affine)
    - from: data coordinates (voxel column i, row j, slice k)
    - to: a real-world position (x,y,z mm) in a coordinate system (e.g. Scanner or MNI)



- SPM changes headers only where possible
  - to avoid interpolation in realign and co-register (estimate)
  - enforce rewriting the data (by interpolation) by"reslice"/"write"

E. Reslicing/Interpolation

- Images have to be saved as voxel intensity list on disk
- After applying transformation parameters, data is re-sampled onto same grid of voxels as reference image



Reoriented

Resliced

1x1x3 mm voxel size

2x2x2 mm voxel size

SPM 3D vs 4D

- Always take care whether you specify a nifti-file (containing a 4D dataset, including different volumes) or a nifti-frame (one 3D volume)
  - file: `subject01/fmri/fmri.nii`
  - frame: `subject01/fmri/fmri.nii,1 (or ,15 etc.)`
- The Batch Editor is not always consistent in what it needs, i.e. just the file name or a list of *all* individual frames
- This is one the most common error sources using preprocessing pipelines