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:
  - Section: Talk and Lecture Materials

- SPM12 (Statistical Parametric Mapping)
  - [http://www.fil.ion.ucl.ac.uk/spm/software/spm12/](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/](http://www.translationalneuromodeling.org/tapas/)
  - Documentation & Example Data (Philips/Siemens/GE):
    - [http://www.translationalneuromodeling.org/software/documentation/](http://www.translationalneuromodeling.org/software/documentation/)
Outline

- Setting up a Preprocessing Pipeline in SPM: The Batch Editor
- Monitoring and Comparing Pipelines: Quality Assurance
- Multi-subject Pipelines: Automatisation
- Integrating Own Code and Toolboxes: Physiological Noise Modeling and Evaluation
The SPM GUI

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

### 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

**Setup Pipeline**
- fMRI time-series

**Quality Monitoring**
- Structural MRI
- TPMs
- Normalisation: Unified Segmentation
  - Coregistration
  - Realignment

**Multi-Subject Noise Modeling**
- Mean functional
- Motion corrected
- Headers changed

**Output**
- Segmentation
- Deformation Fields (y_struct.nii)
- MNI Space
- GLM

**Input**

```
\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}
```
From Pipelines to Batches

- Elements of a Pipeline
  1. Modules
  2. Dependencies
  3. Data (Subject-specific)

- Philosophy: Batch = Materialised Pipeline

**fMRI time-series**

**Structural MRI**

---

**Elements of a Pipeline**

1. Modules
2. Dependencies
3. Data (Subject-specific)

---

**Philosophy: Batch = Materialised Pipeline**

---

**Pipeline Components**

1. Modules
2. Dependencies
3. Data (Subject-specific)

---

**fMRI time-series**

**Structural MRI**

- **REALIGN**
  - Input: Motion corrected
  - Output: Mean functional

- **COREG**
  - Input: Motion corrected
  - Output: Mean functional

---

**Motion corrected**

**Mean functional**

---

**Lars Kasper: An SPM Perspective on fMRI Preprocessing**

---

**Batch Editor**

- Current Module: Realign & Unwarp
- Help on: Realign & Unwarp
- Data
- Session
- Images: 100 files
- Estimation Options
  - Quality
  - Separation
  - Smoothing (FWHM)
- Specification/Paths:
  - /Users/kasperla/Dropbox/CcPhysIOPre
  - /Users/kasperla/Dropbox/CcPhysIOPre

---

2015-06-14
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:
    \[
    \text{spm\_check\_registration('img1.nii', 'img2.nii', ...)}
    \]
- 4D NIFTI files:
  - show individual image tiles:
    \[
    \text{spm\_check\_registration('fmri.nii,1', 'fmri.nii,2', ...)}
    \]
  - show movie (NEW! SPM12):
    \[
    \text{spm\_check\_registration('fmri.nii')}\]
- 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)

Demo 1 – GUI Batch Editor

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

### Module List

- **Realign & Unwarp**
  - Segment
  - Coregister: Estimate
  - Normalise: Write
  - 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

- File path: 
  - `/Users/kasperla/Dropbox/Conferences/SPMZurich15/PracticalsPhysIOPreprocessing`

### Registration Options

- Register to first
  - 2nd Degree B-spline
- No wrap
- 0 files

### Smoothing Options

- FWHM: 5
- 0.9

### Other Parameters

- No wrap
- Medium
- No
Demo 1 - Dependencies

Setup Pipeline  Quality Monitoring  Multi-Subject  Noise Modeling

Batch Editor
File  Edit  View  SPM  BasicIO

Module List
- Realign & Unwarp
- Segment
- Coregister: Estimate
- DEP
- Normalise: Write
- DEP
- Smooth
- DEP

Help on: Coregister: Estimate
Reference Image
DEP Segment: Bias Corrected (1)
Source Image
DEP Realign & Unwarp: Unwarped Mean Image
Other Images
- ... Realign & Unwarp: Unwarped Images (Sess 1)
Estimation Options
- Objective Function
- Separation
- Tolerances
- Histogram Smoothing

Current Module: Coregister: Estimate

Reference from
- Realign & Unwarp: Unwarped Mean Image

Current Item: Source Image

Specify...
Dependency
Demo 1 - Results

- Automatic status plots saved in \texttt{spm_<date>.ps}

**Image realignment**

- Translation plots:
  - \( x \) translations
  - \( y \) translations
  - \( z \) translations

- Rotation plots:
  - \( p \times \) translations
  - \( r \times \) translations
  - \( y \times \) translations

**Normalised Mutual Information Coregistration**

\[ X_1 = 3.000^3\times +0.019^2\times +0.017^2\times +28.728 \]
\[ Y_1 = -0.020^3\times +0.000^2\times +0.043^2\times +0.827 \]
\[ Z_1 = -0.006^3\times -0.016^2\times +0.000^2\times -3.367 \]

**REALIGN**

**COREG**

Original Joint Histogram

Final Joint Histogram

Images before and after realignment.
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

- **Setup Pipeline**
- **Quality Monitoring**

- **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

2015-09-25 Lars Kasper: fMRI Preprocessing & Noise Modeling
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

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

File | Edit | View | SPM | BasicIO

Batch Editor

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: DEP
- 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
    - . String
      - . NIFTI Images
        - . NIFTI Images
          - . NIFTI Images

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/`
  - `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
Example: Comparing temporal SNR throughout preprocessing

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

```matlab
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
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…

- 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

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

- **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/PracticalsPhysIOPreprocessi

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/PracticalsPhysIOPreprocessi

jobs = repmat(jobfile, 1, nrun);
inputs = cell(3, nrun);

for crun = 1:nrun

    pathSubject = fullfile(pathStudy, dirSubjectArray{crun});
    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 1\textsuperscript{st} 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

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

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

log_files
  . vendor
  . log_cardiac
  . log_respiration
  . log_scan_timing
  . sampling_interval
  . relative_start_acquisition
  . align_scan

sqpar (Sequence timing parameters)
  . Nslices
  . NslicesPerBeat
  . TR
  . Ndummies
  . Nscans
  . onset_slice
  . time_slice_to_slice
  . Nprep

thresh (Thresholding parameters for de-noising and timing)

Current item: save_dir

Reference from
Make Directory: Make Directory 'first_level_2BF'

Specify...
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
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"