A Pypelined Approach to Neuroimaging Data Analysis

Satrajit Ghosh and Susan Whitfield-Gabrieli

Massachusetts Institute of Technology
Acknowledgements

**People**

Alfonso Nieto-Castanon, PhD  
Jason A. Tourville, PhD  
Oliver Hinds, PhD  
Mark Pearrow

**Institutions**

**Massachusetts Institute of Technology**  
*Speech communication group, Research Laboratory of Electronics*  
*Athinoula A. Martinos Imaging Center, McGovern Institute for Brain Research*

**University of California, Berkeley**

**Boston University**  
*SpeechLab, Department of Cognitive and Neural Systems*

**Massachusetts General Hospital**  
*Athinoula A. Martinos Center for Biomedical Imaging*

**NIPy team**

*University of California, Berkeley*  
Dav Clark  
Matthew Brett, MD  
Christopher Burns  
Mark D’Esposito  
Cindee Madison  
Jarrod Millman  
Fernando Perez, PhD

*Neurospin, Sarclay, France*  
Jean-Baptiste Poline, PhD  
Alexis Roche, PhD  
Bertrand Thirion, PhD  
Gael Varoquaux, PhD

*Stanford University*  
Jonathon Taylor, PhD

*University of Magdeburg, Germany*  
Michael Hanke

*Rutgers University Newark, USA*  
Yaroslav O. Halchenko

and the list goes on.

**Funding**

National Institute for Biomedical Imaging and Bio-engineering  
R03 EB008673  
PIs: Satrajit Ghosh, Susan Whitfield-Gabrieli, MIT
RAPIDART
Why neuroimaging pipelines?

Why in Python?
Why another batch scripting system?

Architecture

Data Interoperability
Region-of-interest Analysis of Parcellated Imaging Data
Reducing inter-subject anatomical variability

Artifact detection Tools
Quality assurance

MATLAB-based
SPM-focused
Problem 1. Data formats

**Neuroimaging Pypelines**

<table>
<thead>
<tr>
<th>SPM</th>
<th>FSL</th>
<th>FreeSurfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>nii (3D/4D)</td>
<td>nii.gz/nii  (4D)</td>
<td>mgz</td>
</tr>
<tr>
<td>img/hdr (analyze)</td>
<td></td>
<td>nii</td>
</tr>
<tr>
<td>MAT (matlab)</td>
<td>C++</td>
<td>C++</td>
</tr>
<tr>
<td>MATLAB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Neuroimaging Pipelines

Wishlist

Motion Correction
Coregistration
Normalization
Smoothing
Model Specification
Model Estimation
Statistical Inference
Neuroimaging Pipelines

Optimized processing

- Motion Correction
- Coregistration
- Normalization
- Smoothing
- Model Specification
- Model Estimation
- Statistical Inference

Wishlist

Compare algorithms

- Motion Correction
- Coregistration
- Normalization

Smoothing
- Smoothing SPM
- Smoothing FreeSurfer

Normalization
- Normalization ANTS

Coregistration
- Coregistration SPM

Model Specification
- Model Specification NIPY

Model Estimation
- Model Estimation FMRISTAT

Model Estimation
- Model Estimation

Statistical Inference

Thursday, June 18, 2009
Neuroimaging Pipelines

Optimized processing:
- Motion Correction
- Coregistration
- Normalization
- Smoothing
- Model Specification
- Model Estimation
- Statistical Inference

Compare algorithms:
- Motion Correction
- Coregistration
- Normalization
- Smoothing
- Model Specification
- Model Estimation
- Statistical Inference

Custom Software:
- Preprocessing
- Model Specification
- Model Estimation
- Statistical Inference

Thursday, June 18, 2009
Neuroimaging Pypelines

• High-level language
• Well formed programming language
• Cross-platform
• Free
• Extensive community, commercial support and use
• Strong numerical and robust scientific computing packages (numpy, scipy)
• Integrates well with other software (PyR2)
• Interactive-command line
• Automatic doc generation

Image courtesy
Dr. Arno Klein

Thursday, June 18, 2009
• Interoperability!
• Extensible framework based on a high-level language
• Integrates different neuroimaging software (FSL, SPM, etc.)
• Lightweight
• Provides embarrassing and finer levels of parallelism
• Minimizes data redundancy
• Supports evaluation of the impact of different parameters
<table>
<thead>
<tr>
<th>Neuroimaging Pypelines</th>
<th>Approach</th>
</tr>
</thead>
</table>

Target three levels of users:
- those who like to drag and drop
- those who like to type config files
- those who like to code
Target three levels of users:
- those who like to drag and drop
- those who like to type config files
- those who like to code
• Processing pipeline represented as a directed acyclic graph (DAG)
• Leverages graph algorithms for scheduling
• Each process defines its own inputs and outputs
• Provides wrappers/interfaces to call external software (SPM, FSL)
• Can iterate pipeline-segment over parameterizations of modules
• Separates data specification and code execution
• Generalized framework for neuroimaging analysis
• Integrates several neuroimaging software
• Python-based (easy to learn, easy to use) – really
• Pedagogical (you should know your parameters)
• Engages the community
• Go to Matthew Brett’s talk tomorrow
Neuroimaging Pypelines

Open questions

Problem 1. Data Formats
Dicom, Nifti, Analyze, MATLAB, Gifti, MGZ, VTK

Problem 2. Metadata
What metadata should be carried along with the data?
What is essential?

Problem 3. Ontology
What is our vocabulary?
How are all the relations stored?