

Goal: run group analysis on single subject response magnitudes

- ❖ how do we get there?
 - create beta (response magnitude) maps for each subject
 - should be aligned, probably to a well known template
 - run group analysis program (e.g. **3dttest++**, **3dMEMA**, **3dANOVA***)
 - can use **uber_ttest.py** to run single group tests

- ❖ how do we create aligned beta maps?
 - write single subject processing script: pre-processing through regression
 - inputs: anat, EPI, stimulus timing
 - controls: processing decisions like blur size and alignment template
 - outputs: beta weights (and t-stats, contrasts, blur estimates, etc.)

- ❖ how do we write single subject processing scripts?
 - **afni_proc.py** can be used to generate processing scripts
 - an **afni_proc.py** command can be included in publication
 - ✓ along with the **AFNI** version (e.g. **AFNI_17.2.09**)
 - proc scripts are meant to be clear records of the processing

General suggestions

- ❖ picture this experiment as your own (i.e. feel responsibility)
 - decisions on processing were made by you (and your colleagues)
 - hopefully before acquiring any data
 - there is no single "correct" way to analyze data, just reasonable ways
- ❖ focus on understanding the processing steps
 - in light of your having chosen which steps to perform
- ❖ practice the good habit of reviewing results
 - do the initial images look good?
 - review each processing step along with data
 - are the EPI and anat well aligned by the end?
 - do the resulting statistical maps look reasonable?
- ❖ create scripts for any processing steps
 - they are records of how data was processed
 - they are easy to apply to any new subjects
 - they are easy to repeat
 - **expect to re-analyze everything (mistake, new decision, etc.)**
 - keep original data and all processing scripts

Review of stimulus conditions

- ◆ Speech Perception Task: Subjects were presented with audiovisual speech that was presented in a predominantly auditory or predominantly visual modality.
- ◆ A digital video system was used to capture auditory and visual speech from a female speaker.



(1) Auditory-Reliable

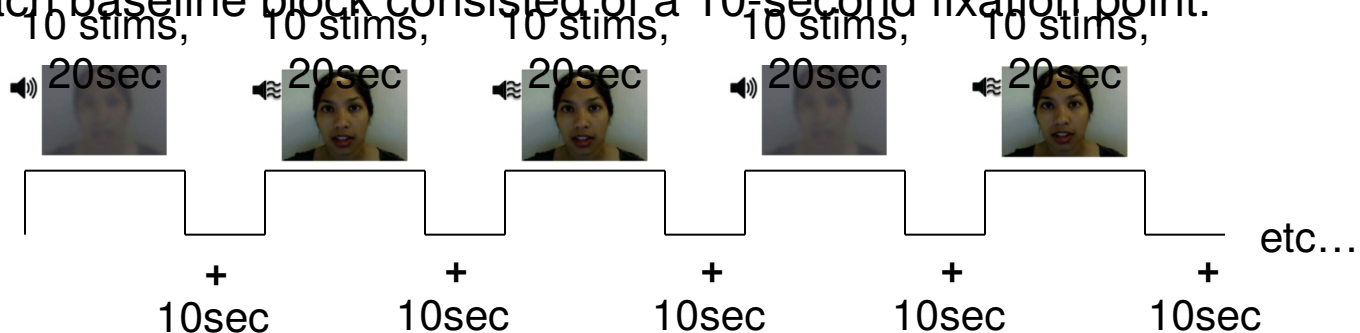
Example: Subjects can clearly *hear* the word “cat,” but the video of a woman mouthing the word is degraded.

(2) Visual-Reliable

Example: Subjects can clearly *see* the video of a woman mouthing the word “cat,” but the audio of the word is degraded.

❖ Experiment Design:

- ◆ There were 3 runs in a scanning session.
- ◆ Each run consisted of 10 blocked trials:
 - 5 blocks contained Auditory-Reliable (*Arel*) stimuli, and
 - 5 blocks contained Visual-Reliable (*Vrel*) stimuli.
- ◆ Each block contained 10 trials of *Arel* stimuli OR 10 trials of *Vrel* stimuli.
 - Each block lasted for 20 seconds (1 second for stimulus presentation, followed by a 1-second inter-stimulus interval).
- ◆ Each baseline block consisted of a 10-second fixation point.



❖ Data Collected:

- ◆ 2 Anatomical datasets for each subject, collected at 3 tesla.
 - 175 sagittal slices
 - voxel dimensions = 0.938 x 0.938 x 1.0 mm

- ◆ 3 Time Series (EPI) datasets for each subject.
 - 33 axial slices x 152 volumes = 5016 slices per run
 - TR = 2 sec; voxel dimensions = 2.75 x 2.75 x 3.0 mm

- ◆ Sample size, $n = 10$ (all right-handed subjects)

afni_proc.py

- What is **afni_proc.py**?
 - ❖ a program used to generate processing scripts for single subject analysis
 - a short command can generate a long processing script to:
 - copy inputs into new 'results' directory
 - process data (e.g. tshift/align/tlrc/volreg/blur/scale/regress)
 - leave results in place to allow review of processing
 - create **@ss_review_*** scripts, for quality control
 - many options for control over processing steps
 - many examples (in -help output) for getting started
 - ❖ generated scripts are in **tcsh** syntax
 - ❖ scripts are written to be easily read (good idea) and modified (bad idea)
 - ❖ preferable to run **afni_proc.py** (generating proc script) per subject
 - rather than running one (modified?) proc script across all subjects
 - graphical user interfaces exist (e.g. **uber_subject.py**), for those who prefer such things

Overview of remaining steps

❖ `cd AFNI_data6/FT_analysis`

- review directory contents and note subject data under directory **FT**
- review the **afni_proc.py** command in **s05.ap.uber**

❖ `tcsch s05.ap.uber`

- runs **afni_proc.py** to generate proc script **proc.FT**
- executes **proc.FT**, saving text output to **output.proc.FT**
- processed results are under **results.FT** directory

❖ review proc script **proc.FT** while viewing processed data

- `cd FT.results ; afni`

❖ run quality control review script, `@ss_review_driver`

- `tcsch @ss_review_driver`

❖ run group analysis (**3dttest++**, **3dMEMA** or **3dANOVA2**)

- from the **AFNI_data6/group_results** directory:
- `tcsch s6.ttest.covary`

Note what is under **AFNI_data6/FT_analysis**

FT

s01.ap.simple

s05.ap.uber

s09.cleanup

s11.proc.FT

s15.proc.FT.uber

- subject data directory
- basic **afni_proc.py** script
- more advanced script
- remove analysis results
- result of **s01.ap.simple**
- result of **s05.ap.uber**

under **FT**

AV1_vis.txt

AV2_aud.txt

FT_anat+orig.BRIK/HEAD

FT_epi_r1+orig.BRIK/HEAD

FT_epi_r2+orig.BRIK/HEAD

FT_epi_r3+orig.BRIK/HEAD

- visual reliable timing
- auditory reliable timing
- anatomical dataset
- EPI run 1
- EPI run 2
- EPI run 3

AV1_vis.txt:

60 90 120 180 240

120 150 180 210 270

0 60 120 150 240

Single Subject Analysis: FT

- ❖ change to analysis directory and review **afni_proc.py** command
 - **cd AFNI_data6/FT_analysis**
 - **cat s05.ap.uber**
- ❖ execute that command, which also processes the data
 - **tcsch s05.ap.uber**
- ❖ review processing script and results
 - review the **proc.FT** script while looking at the results under **FT.results**
 - **afni_open -e proc.FT**
 - **cd FT.results**
 - **ls**
 - **afni**
- ❖ run automatically generated quality control review script
 - **tcsch @ss_review_driver**
 - considered a **minimal** data review (run for every subject)
 - for each step in the review:
 - ✓ read prompt text in each black window and follow instructions
 - ✓ close any windows newly opened by the script
 - ✓ click “OK” to move on to the next step

Group Analysis: paired t-test (Vrel-Arel)

- ❖ **cd AFNI_data6/group_results**
- ❖ review the **3dttest++** script and possibly the covariates file
 - **cat s6.ttest.covary**
 - **cat covary.toe.gap.txt**
- ❖ execute the **3dttest++** command script
 - **tcsch s6.ttest.covary**
- ❖ view the results, in all their glory
 - **afni**
 - set OverLay to **stat.6.covary**
 - set Olay/Thr volumes to #0/#1, for Vrel-Arel and Tstat
 - threshold at $p < 0.005$ (right-click on T-t above threshold slider)
 - set color range scale to 1.0
 - Clusterize (with defaults) and open Rpt (cluster report) window
 - jump to CMass (center of mass) locations

AFNI Start to Finish (the horror continues...)

- To continue reviewing the data on your own, please see the corresponding tutorial that continues under the data directory:

➤ **AFNI_data6/FT_analysis/tutorial**

- Alternatively, this can be viewed from the AFNI web site:

http://afni.nimh.nih.gov/pub/dist/edu/data/CD.expanded/AFNI_data6/FT_analysis/tutorial

- or from the Help menu of **uber_subject.py**
 - ❖ Help --> Browse --> web: tutorial-single subject analysis