

AFNI Start to Finish: fMRI Analysis with AFNI

The image displays a workflow for fMRI analysis using AFNI. It includes:

- uber_subject.py GUI:** A graphical interface for setting up subjects. It shows fields for 'subject ID' (FT) and 'group ID' (horses). Under 'EPI datasets', it lists three scans with their corresponding EPI datasets. Under 'stimulus timing files', it shows a table of stimuli:

index	label	basis func	stim timing file
1	vis	BLOCK(20,1)	AV1_vis.txt
2	aud	BLOCK(20,1)	AV2_aud.txt
- AFNI Message Board:** A screenshot of the AFNI website showing community updates and resources.
- Terminal Windows:**
 - `afni_proc.py` script: A shell script that automates the AFNI processing pipeline, including motion correction, registration, and regression.
 - `tchsh -xef proc.FT` command: A terminal window showing the execution of the `proc.FT` script, with progress updates for each job (e.g., 'Job #1 finished', 'Job #2 finished').

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**, **3dMVM**)
 - can use **gen_group_command.py** to run simple 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_25.2.13**)
 - 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?
 - review (at least) the afni_proc.py HTML QC for every subject
- ❖ 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.
- ◆ There were 2 types of stimulus conditions:



(1) **Auditory-Reliable**

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

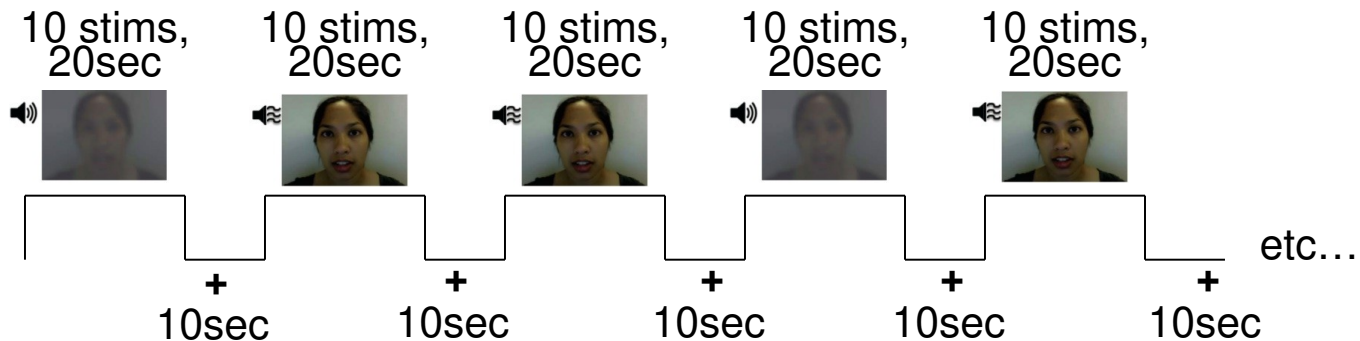


(2) **Visual-Reliable**

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 (*aud*) stimuli, and
 - 5 blocks contained Visual-Reliable (*vis*) stimuli.
- ◆ Each block contained 10 trials of *aud* stimuli OR 10 trials of *vis* 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 an APQC HTML report 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, but not modified (bad idea)
 - ❖ preferable to run **afni_proc.py** (generating proc script) per subject
 - rather than running one (modified?) proc script across all subjects
 - A graphical user interface exists (**uber_subject.py**), but it is currently out of date with afni_proc.py

Overview of remaining steps

- ❖ **cd AFNI_data7/task_demo_ap**
 - review directory contents and subject data under directory **sub-000**
 - review the **afni_proc.py** command in **do_21_ap_affine.tcsh**
- ❖ **tcsh do_21_ap_affine.tcsh**
 - runs **afni_proc.py** to generate proc script **proc.sub-000.affine**
 - executes **proc.sub-000.affine**
 - saving text output to **output.proc.sub-000.affine**
 - processed results are under **sub-000.affine.results** directory
- ❖ review proc script **proc.sub-000.affine** while viewing processed data
 - **cd sub-000.affine.results ; afni**
- ❖ review quality control HTML report
 - **open_apqc.py -infile sub*.results/QC_sub*/index.html**
- ❖ run group analysis (**3dttest++**, **3dMEMA** or **3dANOVA2**)
 - from the (older) **AFNI_data6/group_results** directory:
 - **tcsh s6.ttest.covary**

Note what is under **AFNI_data7/task_demo_ap**

do_01.gtkyd.tcsh

- script: getting to know your data

do_14_timing.tcsh

- script: TSV to AFNI timing conversion

do_20_ap_simple.tcsh

- script: ap_run_simple_rest.tcsh command

do_21_ap_affine.tcsh

- script: affine (faster) **afni_proc.py**

do_22_ap_nonlinear.tcsh

- script: better NL registration version

do_23_ap_surf.tcsh

- script: **afni_proc.py** for surface analysis

save.proc.sub-000.affine

- script: proc from **do_21_ap_affine.tcsh**

SSW

- data: NL alignment results from sswarper2

sub-000

- data: BIDS subject data directory tree

SUMA

- data: FreeSurfer surfaces imported into SUMA

under **sub-000/anat**

sub-000_T1w.nii.gz

- anatomical dataset

under **sub-000/func**

sub-000_task-av_run-01_bold.nii.gz

- run 1 EPI time series

sub-000_task-av_run-01_events.tsv

- run 1 event times, TSV format

sub-000_task-av_run-02_bold.nii.gz

- run 2 EPI time series

sub-000_task-av_run-02_events.tsv

- run 2 event times, TSV format

sub-000_task-av_run-03_bold.nii.gz

- run 3 EPI time series

sub-000_task-av_run-03_events.tsv

- run 3 event times, TSV format

times.aud.txt

- AFNI event timing: aud

times.vis.txt

- AFNI event timing: vis

Single Subject Analysis: FT

- ❖ change to analysis directory and review **afni_proc.py** command
 - **cd AFNI_data7/task_demo_ap**
 - **cat do_21_ap_affine.tcsh**

- ❖ execute that command, which also processes the data
 - **tcsh do_21_ap_affine.tcsh**

- ❖ review processing script and results
 - review the **proc.sub-000.affine** script
 - while looking at the results under **sub-000.affine.results**
 - **afni_open -e proc.sub-000.affine**
 - **cd sub-000.affine.results**
 - **ls**
 - **afni**

- ❖ review automatically generated quality control HTML report
 - **open_apqc -infile QC_sub-000.simple/index.html**
 - considered a **minimal** data review (run for every subject)

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
 - `tcsH 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