3dDeconvolve

New-ish Things — April 2007

Analysis of Functional NeuroImages

by

Robert W Cox, PhD

Released under the GNU General Public License Version 2 (GPL)

Ikuko Mukai

AFNI User

AFNI is a research tool.

Clinical uses are not supported or advised.

http://afni.nimh.nih.gov/pub/dist/doc/misc/Decon/DeconSummer2004.html

http://afni.nimh.nih.gov/pub/dist/doc/misc/Decon/DeconSpring2007.html

Summary

- Direct input of stimulus timing with `-stim_times` option
  - ★ With specification of a model for the BOLD response expected after each stimulus
  - ★ Amplitude modulated BOLD response option with `-stim_times_AM2`
- Master script to carry out entire single-subject analysis
  - ★ Smoothing, registration, masking, scaling, statistics
- Generation of fitted BOLD response sub-models with the new `3dSynthesize` program
- Smaller changes:
  - ★ More extensive checking for potential errors or problems
  - ★ `-float` option to output floating point format datasets
  - ★ `-CENSORTR` option to censor out individual time points
- Program `3dBlurToFWHM` for controlled & masked blurring

Analysis Using Stimulus Timing

```
3dDeconvolve -input rall_vr+orig -concat '1D: 0 108 216 324'
 -num_stimts 4
 -stim_times 1 '1D: 17.5 | 185.0 227.5 | 60.0 142.5 | 227.5'
               'BLOCK(20,1)'
 -stim_times 2 '1D: 100.0 | 17.5 | 185.0 227.5 | 17.5 100.0'
               'BLOCK(20,1)'
 -stim_times 3 '1D: 60.0 227.5 | 60.0 | 17.5 | 142.5 185.0'
               'BLOCK(20,1)'
 -stim_times 4 '1D: 142.5 185.0 | 100.0 142.5 | 100.0 | 60.0'
               'BLOCK(20,1)'
 -stim_label 1 Actions  -stim_label 2 Tools
 -stim_label 3 HighC    -stim_label 4 LowC
 -gltsym 'SYM: Actions -Tools'                -glt_label 1 AvsT
 -gltsym 'SYM: HighC   -LowC'                 -glt_label 2 HvsL
 -gltsym 'SYM: Actions Tools -HighC -LowC'    -glt_label 3 ATvsHL
 -fout -tout
 -bucket func_rall -fitts fitts_rall
 -xjpeg xmat_rall.jpg -x1D xmat_rall.x1D -cbucket coef_rall
```

- Run start TR indexes
- τ's on command line instead of file:
  - | indicates a new "line" (1 line of stimulus start times per run)
  - Indicates BOLD response model used after each stimulus time
- generate **Actions–Tools** statistical map

Stimulus Timing Input

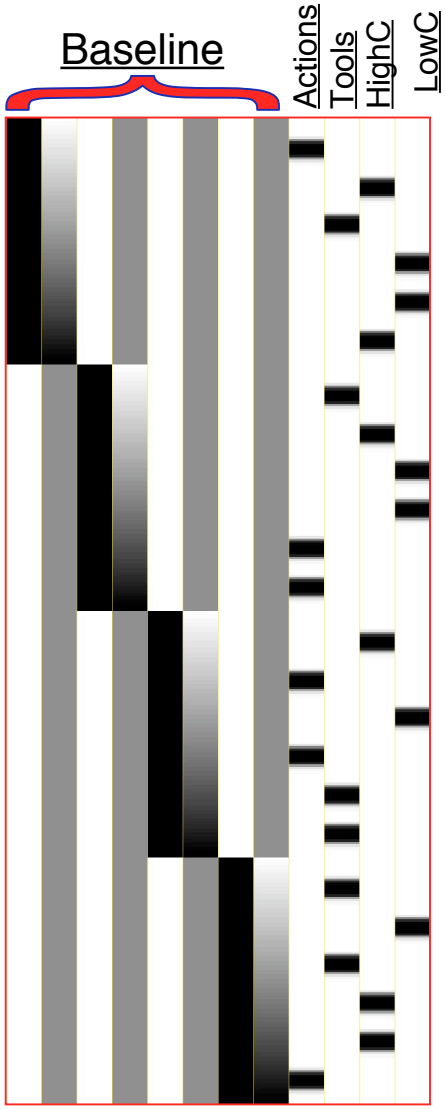
```
-num_stimts 4
```

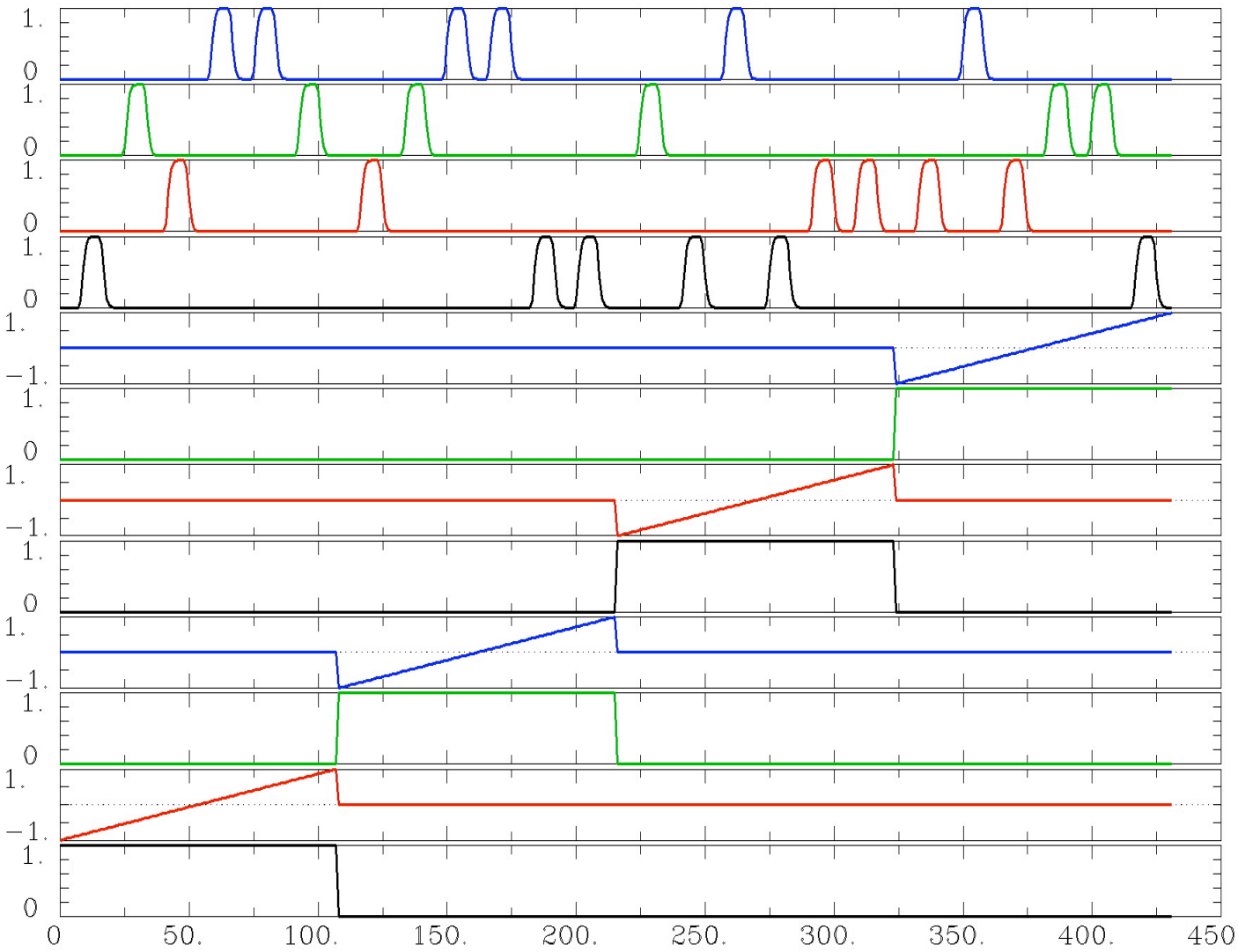
- We have 4 stimulus classes, so need 4 `-stim_times` options

```
-stim_times 1
'1D: 17.5 | 185.0 227.5 | 60.0 142.5 | 227.5'
'BLOCK(20,1)'
```

- "File" with 4 lines, each line specifying the start time in seconds for the stimuli within the corresponding imaging run, with the time measured relative to the start of the imaging run itself
- HRF for each block stimulus (stimulus duration = 20 s) is specified to go to maximum value of 1 (cf. graphs on next slide)
  - ★ This feature is useful when converting FMRI response magnitude to be in units of percent of the mean baseline

Regressor Matrix for This Script





via **-xjpeg**

via **-x1D** and **1dplot -sep_scl xmat_rall.x1D**

Two Possible Formats for `-stim_times`

- A single column of numbers (GLOBAL times)
  - ★ One stimulus time per row
  - ★ Times are relative to first image in dataset being at ***t*=0**
  - ★ May not be simplest to use if multiple runs are catenated

```
 4.7
 9.6
11.8
19.4
```

- One row for each run within a catenated dataset (LOCAL times)
  - ★ Each time in j^{th} row is relative to start of run #*j* being ***t*=0**
  - ★ If some run has NO stimuli in the given class, just put a single "`*`" in that row as a filler
    - o Different numbers of stimuli per run are OK
    - o At least one row must have more than 1 time (so that the LOCAL type of timing file can be told from the GLOBAL)

```
4.7 9.6 11.8 19.4
*
8.3 10.6
```

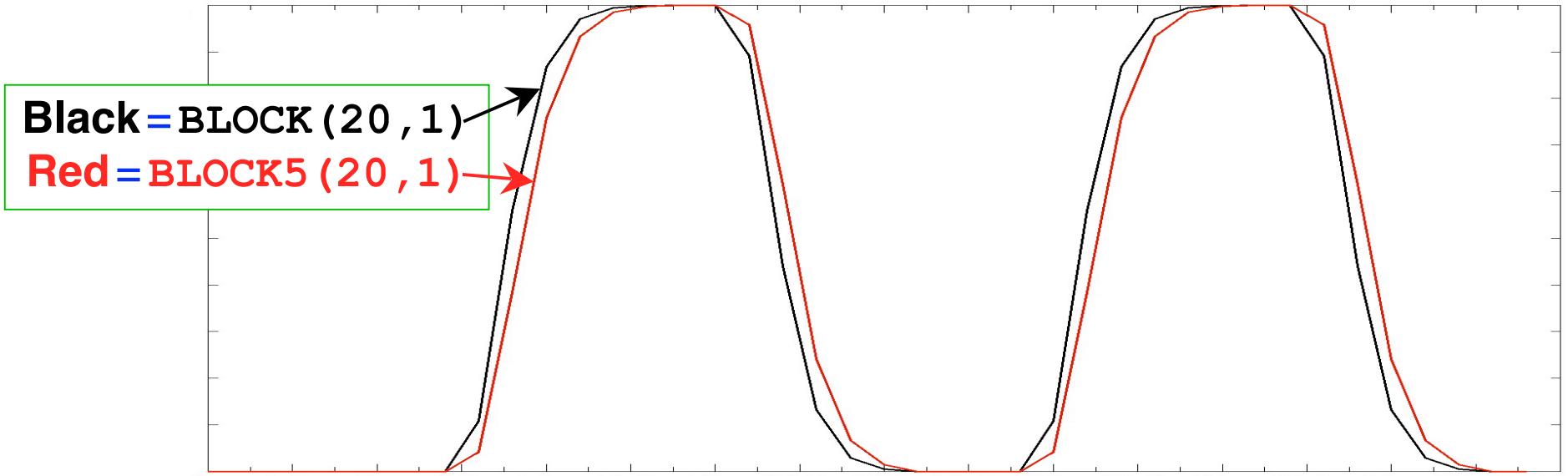
- Two methods are available because of users' diverse desires
  - ★ **N.B.**: if you chop first few images off the start of each run, the inputs to `-stim_times` must be adjusted accordingly

The 'BLOCK()' HRF Model

- **BLOCK(L)** is convolution of square wave of duration **L** with "gamma variate function" $t^4 e^{-t} / [4^4 e^{-4}]$ (peak value=1 at t=4):

$$h(t) = \int_0^{\min(t,L)} s^4 e^{-s} / [4^4 e^{-4}] ds$$

- "Hidden" option: **BLOCK5** replaces "4" with "5" in the above
  - Slightly more delayed rise and fall times
- **BLOCK(L,1)** makes peak amplitude of *block* response = 1



AM Regression - 1

- **AM** = **A**mplitude **M**odulated (or Modulation)
  - ★ Have some extra data measured about each response to a stimulus, and maybe the BOLD response amplitude is modulated by this
  - ★ Reaction time; Galvanic skin response; Pain level perception; Emotional valence (happy or sad or angry face?)
- Want to see if some brain activations vary proportionally to this **ABI** (**A**uxiliary **B**ehaviorial **I**nformation)

- Discrete levels (2 or maybe 3) of ABI:
  - ★ Separate the stimuli into sub-classes that are determined by the ABI ("on" and "off", maybe?)
  - ★ Use a GLT to test if there is a difference between the FMRI responses in the sub-classes

```
3dDeconvolve ...                                                    \
 -stim_times 1 regressor_on.1D  'BLOCK(2,1)' -stim_label 1 'On'   \
 -stim_times 2 regressor_off.1D 'BLOCK(2,1)' -stim_label 2 'Off' \
 -gltsym 'SYM: +On | +Off' -glt_label 1 'On+Off'                   \
 -gltsym 'SYM: +On -Off'   -glt_label 2 'On-Off' ...
```

  - "`On+Off`" tests for any activation in *either* the "on" or "off" conditions
  - "`On-Off`" tests for differences in activation *between* "on" and "off" conditions
  - Can use `3dcalc` to threshold on ***both*** statistics at once to find a **conjunction**

AM Regression - 2

- Continuous (or several finely graded) ABI levels
  - ★ Want to find active voxels whose activation level also depends on ABI
  - ★ **`3dDeconvolve`** is a linear program, so must make the assumption that the change in FMRI signal as ABI changes is linearly proportional to the changes in the ABI values
- Need to make 2 separate regressors
  - ★ One to find the mean FMRI response (the usual **`-stim_times`** analysis)
  - ★ One to find the variations in the FMRI response as the ABI data varies
- The second regressor should have the form

$$r_{\mathrm{AM2}}(t) = \sum_{k=1}^{K} h(t-\tau_k)\cdot(a_k - \bar{a})$$

  - ★ Where a_k = value of k^{th} ABI value, and $\bar{a}$ is the average ABI value
- Response (β) for first regressor is standard activation map
- Statistics and β for second regressor make activation map of places whose BOLD response changes with changes in ABI
  - ★ Using 2 regressors allows separation of voxels that are active but are *not* detectably modulated by the ABI from voxels that *are* ABI-sensitive

AM Regression - 3

- New feature of **3dDeconvolve**: **-stim_times_AM2**
- Use is very similar to standard **-stim_times**
  - ★ **-stim_times_AM2 1 times_ABI.1D 'BLOCK(2,1)'**
  - ★ The **times_ABI.1D** file has time entries that are "married" to ABI values:

```
10*5 23*4 27*2 39*5
17*2 32*5
*
16*2 24*3 37*5 41*4
```

  - ★ Such files can be created from 2 standard ASCII .1D files using the new **1dMarry** program
    - o The **-divorce** option can be used to split them up
- **3dDeconvolve** automatically creates the two regressors (unmodulated and amplitude modulated)
  - ★ Use **-fout** option to get statistics for activation of the pair of regressors (i.e., testing null hypothesis that *both* β weights are zero: that there is no ABI-independent *or* ABI-proportional signal change)
  - ★ Use **-tout** option to test each β weight separately
  - ★ Can **1dplot** $\mathbf{X}$ matrix columns to see each regressor

AM Regression - 4

- The AM feature is new, and so needs some practical user experiences before it can be considered "standard practice"
  - ★ In particular: don't know how much data or how many events are needed to get good ABI-dependent statistics
- If you want, **`-stim_times_AM1`** is also available
  - ★ It only builds the regressor proportional to ABI data directly, with no mean removed:

$$r_{\mathrm{AM1}}(t) = \sum_{k=1}^{K} h(t-\tau_k)\cdot a_k$$

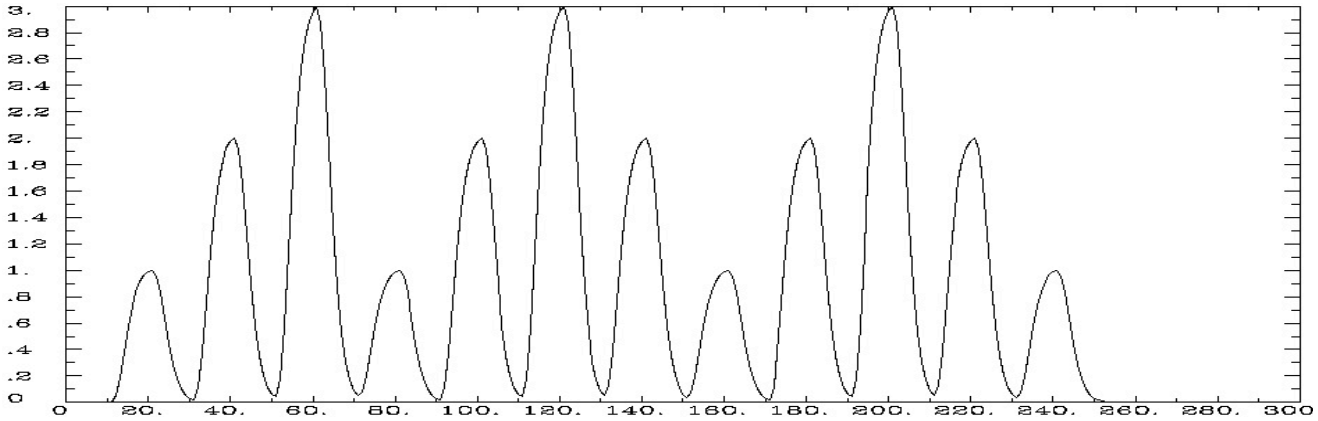
  - ★ Can't imagine what value this option has, but you never know … (if you can think of a good use, let me know)
- Future directions:
  - ★ Allow more than one amplitude to be married to each stimulus time (insert obligatory polygamy/polyandry joke here)
    - o How many ABI types at once is too many? I don't know.
  - ★ How to deal with unknown nonlinearities in the BOLD response to ABI values? I don't know. (Regress each event separately, then compute MI?)
  - ★ Deconvolution with amplitude modulation? Requires more thought.

AM Regression - 5

Timing: **AM.1D** = 10*1 30*2 50*3 70*1 90*2 110*3 130*2 150*1 170*2 190*3 210*2 230*1

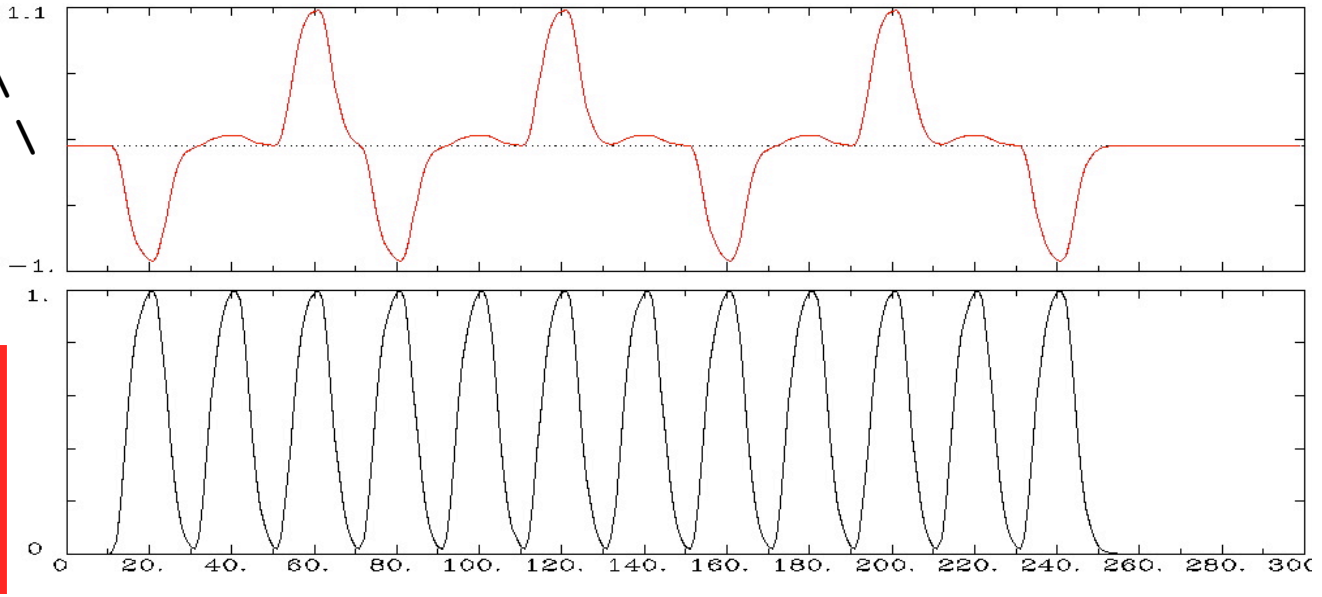
- **3dDeconvolve -nodata 300 1.0 -num_stimts 1 **
  -stim_times_AM1 1 AM.1D 'BLOCK(10,1)' -x1D AM1.x1D
- **1dplot AM1.x1D'[2]'**

AM1 model of signal (modulation = ABI)



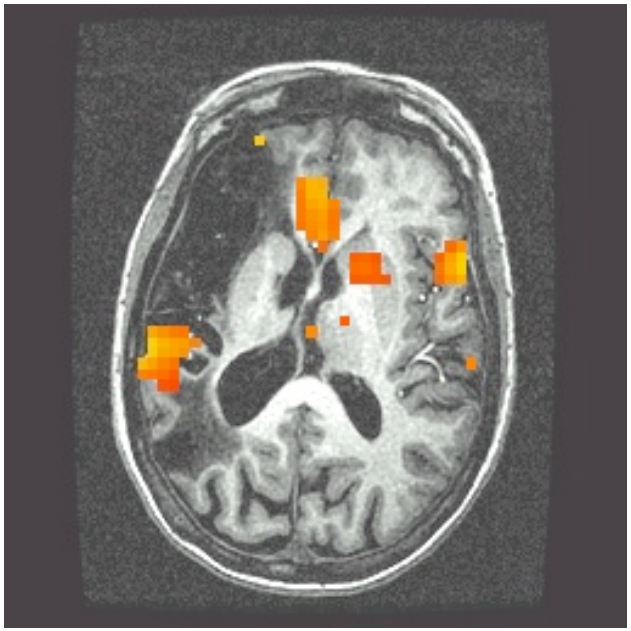
- **3dDeconvolve -nodata 300 1.0 **
  **-num_stimts 1 **
  **-stim_times_AM2 1 **
  **AM.1D 'BLOCK(10,1)' **
  -x1D AM2.x1D
- **1dplot -sepscl **
  AM2.x1D'[2,3]'

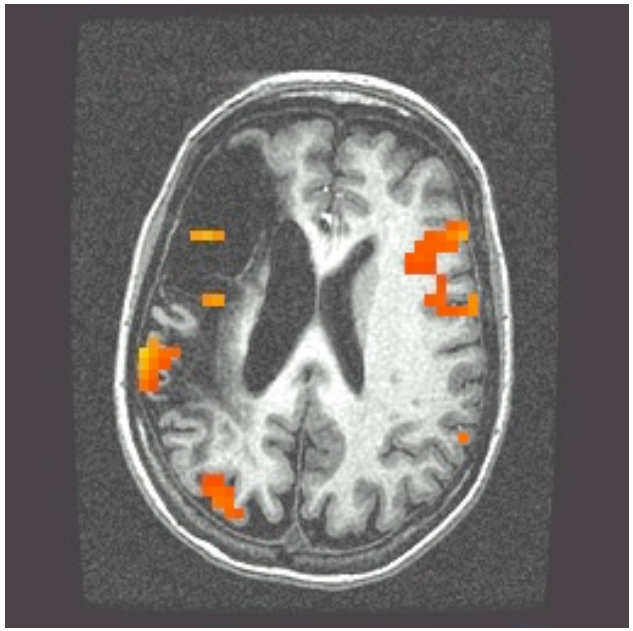
AM2 model of signal: is 2D sub-space spanned by these 2 time series



AM Regression - 6

- First actual user: Whitney Postman (NIDCD; PI=Al Braun)
- Image naming stimulus in stroke (partially aphasic) patient
- ABI data = number of common alternative names for each image (e.g., “balcony”, “porch”, “veranda”), from 1 to 9
  - 9 imaging runs, 144 stimulus events
- 2 slices showing activation map for BOLD responses proportional to ABI





Deconvolution Signal Models

- Simple or Fixed-shape regression:
  - ★ We fix the shape of the HRF — amplitude varies
  - ★ Use `-stim_times` to generate the signal model from the stimulus timing
  - ★ Find the amplitude of the signal model in each voxel — solution to the set of linear equations = β weights
- Deconvolution or Variable-shape regression:
  - ★ We allow the shape of the HRF to vary in each voxel, for each stimulus class
  - ★ Appropriate when you don't want to over-constrain the solution by assuming an HRF shape
  - ★ ***Caveat***: need to have enough time points during the HRF in order to resolve its shape (e.g., TR $\leq$ 3 s)

Deconvolution: Pros & Cons (+ & –)

+ Letting HRF shape varies allows for subject and regional variability in hemodynamics
+ Can test HRF estimate for different shapes (e.g., are later time points more "active" than earlier?)
– Need to estimate more parameters for each stimulus class than a fixed-shape model (e.g., 8 shape parameters *vs*. 1 parameter=amplitude of HRF)
– Which means you need more data to get the same statistical power (assuming that the fixed-shape model you would otherwise use was in fact "correct")
– Freedom to get any shape in HRF results can give weird shapes that are difficult to interpret

Expressing HRF via Regression Unknowns

- The tool for expressing an unknown function as a finite set of numbers that can be fit via linear regression is an **expansion in basis functions**

$$h(t) = \beta_0 \psi_0(t) + \beta_1 \psi_1(t) + \beta_2 \psi_2(t) + \cdots = \sum_{q=0}^{q=p} \beta_q \psi_q(t)$$

  - ★ The basis functions $\psi_q(t)$ & expansion order p are known
    - o Larger $p \Rightarrow$ more complex shapes & more parameters
  - ★ The unknowns to be found (in each voxel) comprises the set of weights β_q for each $\psi_q(t)$
- β weights appear only by multiplying known values, and HRF only appears in signal model by linear convolution (addition) with known stimulus timing
  - Resulting signal model still solvable by linear regression

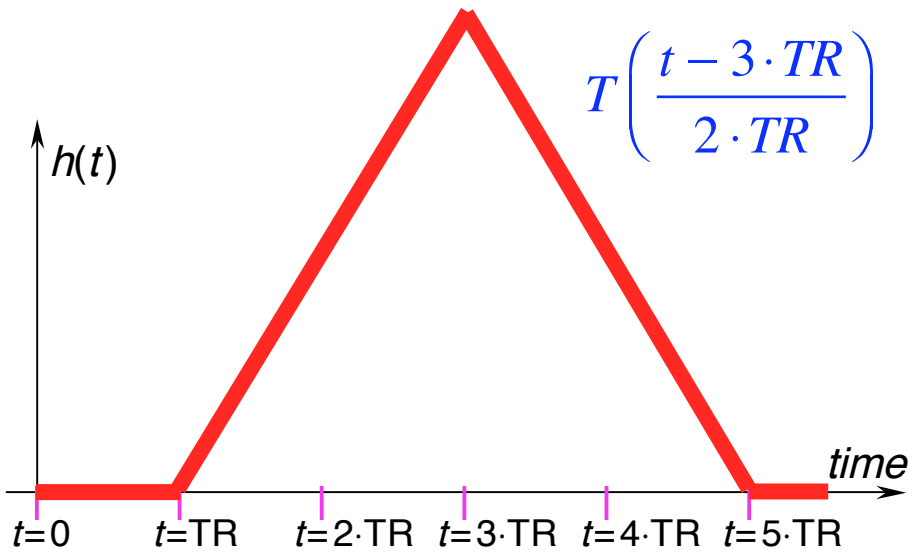
`3dDeconvolve` with "Tent Functions"

- Need to describe HRF shape and magnitude with a finite number of parameters
  - ★ And allow for calculation of $h(t)$ at any arbitrary point in time after the stimulus times:

$$r_n = \sum_{k=1}^{K} h(t_n - \tau_k) = \text{sum of HRF copies}$$

- Simplest set of such functions are **tent functions**
  - ★ Also known as "piecewise linear splines"

$$T(x) = \begin{cases} 1-|x| & \text{for } -1 < x < 1 \\ 0 & \text{for } |x| > 1 \end{cases}$$

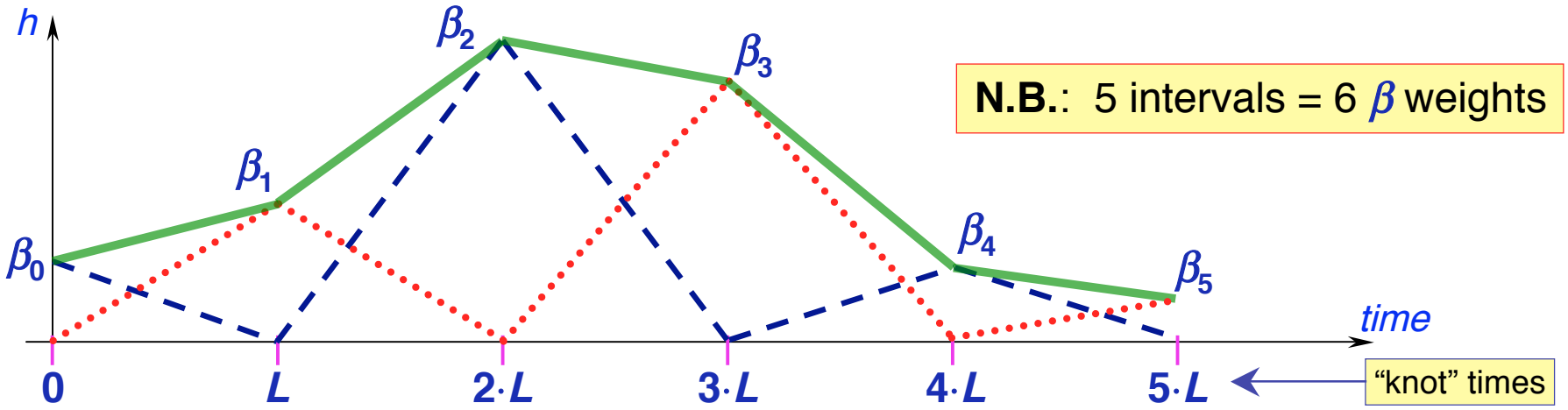


Tent Functions = Linear Interpolation

A

- Expansion of HRF in a set of spaced-apart tent functions is the same as linear interpolation between "knots"

$$h(t) = \beta_0 \cdot T\left(\frac{t}{L}\right) + \beta_1 \cdot T\left(\frac{t-L}{L}\right) + \beta_2 \cdot T\left(\frac{t-2\cdot L}{L}\right) + \beta_3 \cdot T\left(\frac{t-3\cdot L}{L}\right) + \cdots$$



- Tent function parameters are also easily interpreted as function values (e.g., β_2 = response at time $t = 2 \cdot L$ after stim)
- User must decide on relationship of tent function grid spacing L and time grid spacing TR (usually would choose $L \geq$ TR)
- In `3dDeconvolve`: specify duration of HRF and number of β parameters

Master Script for Data Analysis

```
afni_proc.py                                                    \
  -dsets ED/ED_r??+orig.HEAD                                    \
  -subj_id ED.8.glt                                             \
  -copy_anat ED/EDspgr                                          \
  -tcat_remove_first_trs 2                                      \
  -volreg_align_to first                                        \
  -regress_stim_times misc_files/stim_times.*.1D                \
  -regress_stim_labels ToolMovie HumanMovie                     \
                       ToolPoint HumanPoint                     \
  -regress_basis 'TENT(0,14,8)'                                 \
  -regress_opts_3dD                                             \
  -gltsym ../misc_files/glt1.txt -glt_label 1 FullF             \
  -gltsym ../misc_files/glt2.txt -glt_label 2 HvsT              \
  -gltsym ../misc_files/glt3.txt -glt_label 3 MvsP              \
  -gltsym ../misc_files/glt4.txt -glt_label 4 HMvsHP            \
  -gltsym ../misc_files/glt5.txt -glt_label 5 TMvsTP            \
  -gltsym ../misc_files/glt6.txt -glt_label 6 HPvsTP            \
  -gltsym ../misc_files/glt7.txt -glt_label 7 HMvsTM
```

- Master script program
- 10 input datasets
- Set output filenames
- Copy anat to output dir
- Discard first 2 TRs
- Where to align all EPIs
- Stimulus timing files (4)
- Stimulus labels
- HRF model
- Specifies that next lines are options to be passed to `3dDeconvolve` directly (in this case, the GLTs we want computed)

This script generates file `proc.ED.8.glt` (180 lines), which contains all the AFNI commands to produce analysis results into directory `ED.8.glt.results/` (148 files)

Shell Script for Deconvolution - Outline

- Copy datasets into output directory for processing
- Examine each imaging run for outliers: **3dToutcount**
- Time shift each run's slices to a common origin: **3dTshift**
- Registration of each imaging run: **3dvolreg**
- Smooth each volume in space (136 sub-bricks per run): **3dmerge**
- Create a brain mask: **3dAutomask** and **3dcalc**
- Rescale each voxel time series in each imaging run so that its average through time is 100: **3dTstat** and **3dcalc**
  - ⋆ If baseline is 100, then a β_q of 5 (say) indicates a 5% signal change in that voxel at tent function knot #q after stimulus
  - ⋆ Biophysics: believe % signal change is relevant physiological parameter
- Catenate all imaging runs together into one big dataset (1360 time points): **3dTcat**
  - ⋆ This dataset is useful for plotting **-fitts** output from **3dDeconvolve** and visually examining time series fitting
- Compute HRFs and statistics: **3dDeconvolve**

Script - 3dDeconvolve

```
3dDeconvolve -input pb04.$subj.r??.scale+orig.HEAD -polort 2          \
  -mask full_mask.$subj+orig -basis_normall 1 -num_stimts 10          \
  -stim_times 1 stimuli/stim_times.01.1D 'TENT(0,14,8)'               \
  -stim_label 1 ToolMovie                                             \
  -stim_times 2 stimuli/stim_times.02.1D 'TENT(0,14,8)'               \
  -stim_label 2 HumanMovie                                            \
  -stim_times 3 stimuli/stim_times.03.1D 'TENT(0,14,8)'               \
  -stim_label 3 ToolPoint                                             \
  -stim_times 4 stimuli/stim_times.04.1D 'TENT(0,14,8)'               \
  -stim_label 4 HumanPoint                                            \
  -stim_file 5 dfile.rall.1D'[0]' -stim_base 5 -stim_label 5 roll     \
  -stim_file 6 dfile.rall.1D'[1]' -stim_base 6 -stim_label 6 pitch    \
  -stim_file 7 dfile.rall.1D'[2]' -stim_base 7 -stim_label 7 yaw      \
  -stim_file 8 dfile.rall.1D'[3]' -stim_base 8 -stim_label 8 dS       \
  -stim_file 9 dfile.rall.1D'[4]' -stim_base 9 -stim_label 9 dL       \
  -stim_file 10 dfile.rall.1D'[5]' -stim_base 10 -stim_label 10 dP    \
  -iresp 1 iresp_ToolMovie.$subj -iresp 2 iresp_HumanMovie.$subj      \
  -iresp 3 iresp_ToolPoint.$subj -iresp 4 iresp_HumanPoint.$subj      \
  -gltsym ../misc_files/glt1.txt -glt_label 1 FullF                   \
  -gltsym ../misc_files/glt2.txt -glt_label 2 HvsT                    \
  -gltsym ../misc_files/glt3.txt -glt_label 3 MvsP                    \
  -gltsym ../misc_files/glt4.txt -glt_label 4 HMvsHP                  \
  -gltsym ../misc_files/glt5.txt -glt_label 5 TMvsTP                  \
  -gltsym ../misc_files/glt6.txt -glt_label 6 HPvsTP                  \
  -gltsym ../misc_files/glt7.txt -glt_label 7 HMvsTM                  \
  -fout -tout -full_first -x1D Xmat.x1D -fitts fitts.$subj -bucket stats.$subj
```

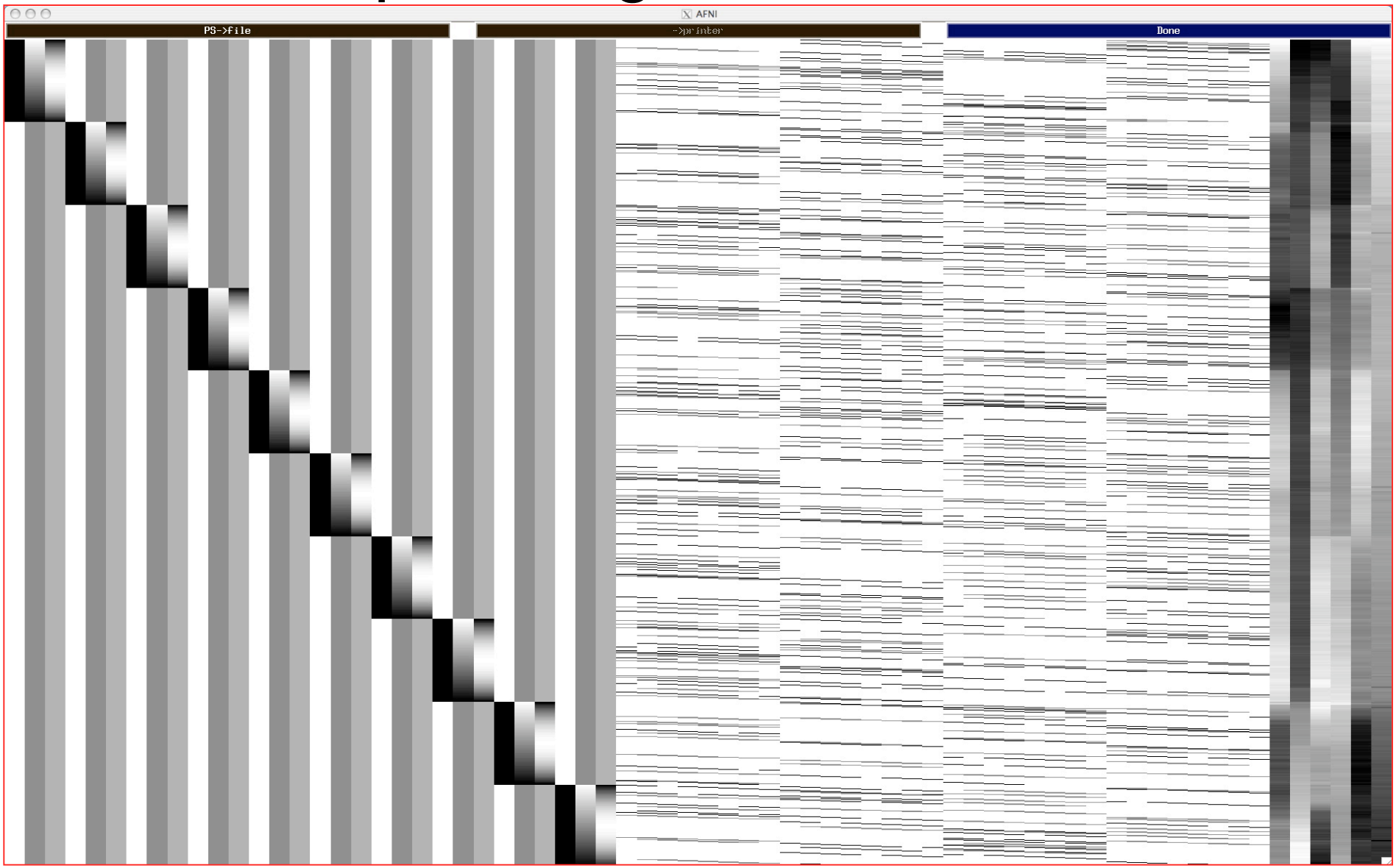
4 stim types

motion params

HRF outputs

GLTs

Script - Image of the X Matrix



Via `1grayplot -sep Xmat.x1D` *or* `-xjpeg` option

Smaller Changes to `3dDeconvolve`

- Equation solver: Program computes **condition number** for **X** matrix (measures of how sensitive regression results are to changes in **X**)
  - ★ If the condition number is "bad" (too big), then the program will not actually proceed to compute the results
  - ★ You can use the `-GOFORIT` option on the command line to force the program to run despite **X** matrix warnings
    - o But you should strive to <u>understand</u> why you are getting these warnings**!!**
- Other matrix checks:
  - ★ Duplicate stimulus filenames, collinear pairs of regression matrix columns, all zero matrix columns


Check the screen output for `WARNING`s and `ERROR`s


  - ★ Such messages also saved into file **`3dDeconvolve.err`**

Smaller Changes - 2

- All-zero regressors *are* allowed (with `-GOFORIT`)
  - ★ Will get zero weight in the solution
  - ★ Example: task where subject makes a choice for each stimulus (e.g., male or female face?)
    - o You want to analyze correct and incorrect trials as separate cases
    - o What if some subject makes no mistakes? Hmmm…
      - ➥ Can keep the all-zero regressor (e.g., all `-stim_times` = `*`)
      - ➥ Input files and output datasets for error-making and perfect-performing subjects will be organized the same way

---

- `3dDeconvolve_f` program can be used to compute linear regression results in single precision (7 decimal places) rather than double precision (16 places)
  - ★ For better speed, but with lower numerical accuracy
  - ★ Best to do at least one run ***both*** ways to check if results differ significantly (Equation solver *should* be safe, but …)

Smaller Changes - 3

- Default output format is 16-bit short integers, with a scaling factor for each 3D volume to convert it to floating point values
  - ★ `-float` option can be used to get 32-bit floating point format output — more precision, *and* more disk space

---

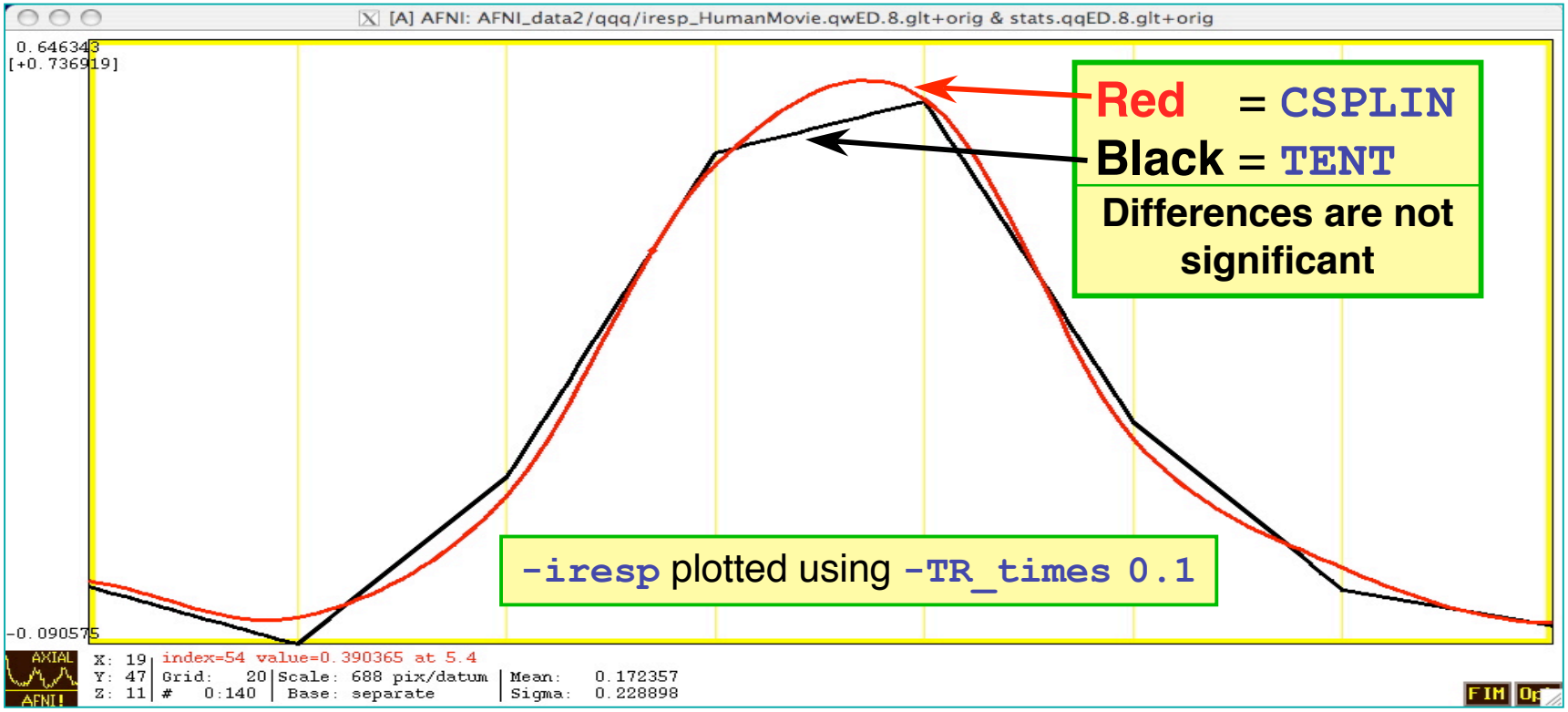
- `3dDeconvolve` recommends a `-polort` value, and prints that out as well as the value you chose (or defaulted to)
  - ★ `-polort A` can be used to let the program set the detrending (AKA “high pass filtering”) level automatically

---

- `-CENSORTR 2:37-39` can be used to censor out time indexes 37, 38, 39 from run #2
  - ★ Simpler to use than the older `-censor` option
  - ★ `-CENSORTR '*:0-1'` removes time pts 0 & 1 from all runs
  - ★ run boundaries specifed by `-concat`

Smaller Changes - 4

- `-stim_times` has some other basis function options for the HRF model besides **`BLOCK`** and **`TENT`**
  - ★ most recent: **`CSPLIN`** = cardinal cubic spline (for smoother HRFs)
  - ★ instead of **`TENT`** = linear spline
    - o Same parameters: `(start,stop,number of regressors)`
    - o Can be used as a “drop in” replacement for **`TENT`**

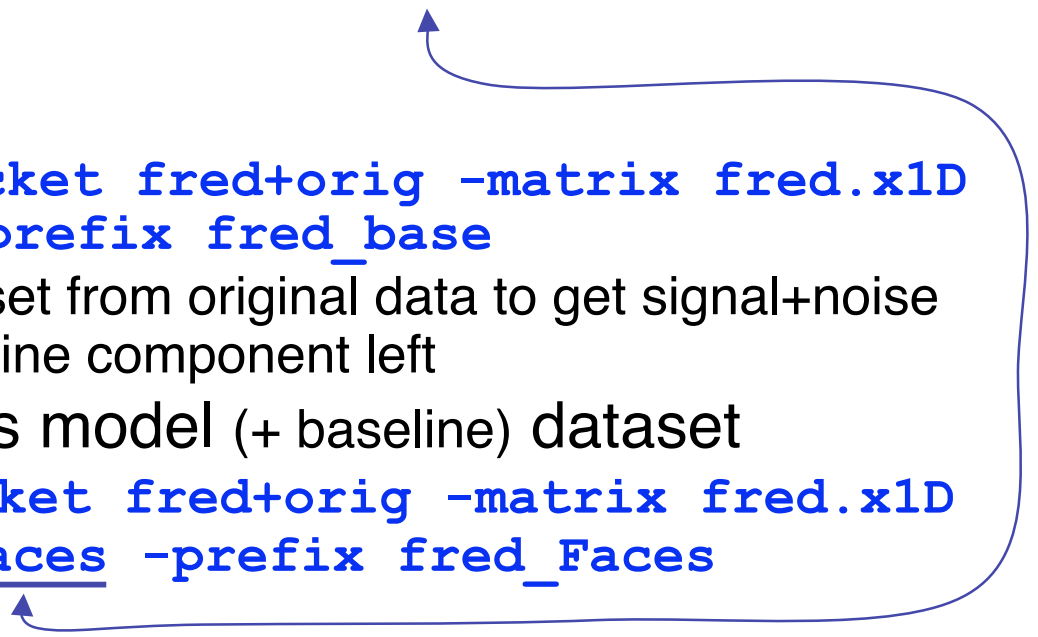


Smaller Changes - 5

- `-fitts` option is used to create a synthetic dataset
  - ★ each voxel time series is full (signal+baseline) model as fitted to the data time series in the corresponding voxel location

---

- **`3dSynthesize`** program can be used to create synthetic datasets from *subsets* of the full model
  - ★ Uses `-x1D` and `-cbucket` outputs from **`3dDeconvolve`**
    - o `-cbucket` stores β coefficients for each $\mathbf{X}$ matrix column into dataset
    - o `-x1D` stores the matrix columns (and `-stim_label`s)
  - ★ Potential uses:
    - o Baseline only dataset
      - ➥ `3dSynthesize -cbucket fred+orig -matrix fred.x1D -select baseline -prefix fred_base`
      - ➥ Could subtract this dataset from original data to get signal+noise dataset that has no baseline component left
    - o Just one stimulus class model (+ baseline) dataset
      - ➥ `3dSynthesize -cbucket fred+orig -matrix fred.x1D -select baseline Faces -prefix fred_Faces`

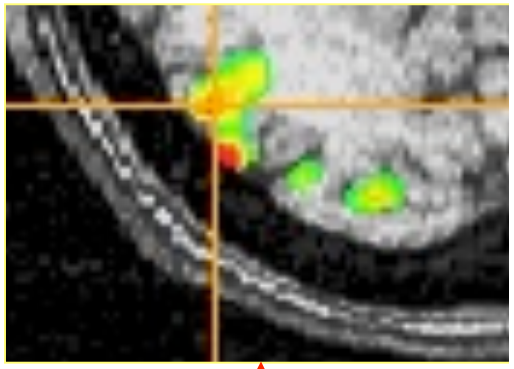


Even Smaller Changes - 6

- Defaults are changed:
  - ★ `-nobout` & `-full_first` & `-bucket` & `-x1D` are always implied
  - ★ Names of statistics sub-bricks are slightly altered (to be more consistent)
- Checks if `-stim_times` inputs are out of range (AKA: the PSFB syndrome)
  - ★ Prints `WARNING` message, but continues analysis
- When using `-nodata` with `-stim_times`, important to give the number of time points and the TR, as in `-nodata 250 2.3`
  - ★ With `-input1D`, use `-TR_1D 2.3` to specify TR

3dBlurToFWHM

- New program to smooth FMRI time series datasets to a specified smoothness (as estimated by FWHM of noise spatial correlation function)
  - ★ Don't just add smoothness (à la **3dmerge**) but control it (locally and globally)
  - ★ Goal: use datasets from diverse scanners



- Why blur FMRI time series?
  - ★ Averaging neighbors will reduce noise
  - ★ Activations are (usually) blob-ish (several voxels across)
  - ★ Diminishes the multiple comparisons problem
- **3dBlurToFWHM** blurs only inside a mask
  - ★ To avoid mixing air (noise-only) and brain voxels
  - ★ Partial Differential Equation (PDE) based blurring method
    - o 2D (intra-slice) or 3D blurring

In the Planning Stages

- "Area under curve" addition to `-gltsym` to allow testing of pieces of HRF models from `-stim_times`
- Slice- and/or voxel-dependent regressors
  - ★ For physiological noise cancellation, etc.
  - ★ To save memory? (Could process each slice separately)
    - o One slice-at-a-time regression can be done in a Unix script, using 3dZcutup and 3dZcat programs
- Extend AM regression to allow for more than 1 piece of auxiliary information at each stimulus time
- Interactive tool to examine `-x1D` matrix for problems
  - ★ and `3dDeconvolve` testing of GLT submatrices