# 3dREMLfit

## AFNI's New Approach to Dealing with Serial Correlation in FMRI Linear Regression (GLM)

RW Cox
Autumn 2008

# Conclusions First

- Serial correlation does not appreciably impact the activation magnitudes ($\beta$s) estimated using **`3dDeconvolve`** (= Ordinary Least Squares solution)
- Group activation maps made from combining these $\beta$s using **`3dANOVA`**, **`3dLME`**, etc., are essentially the same using **`3dDeconvolve`** or **`3dREMLfit`** (= Generalized Least Squares solution)
  - ➢ In other words, **there is no need to re-run old group analyses** to see if allowing for serial correlation will change the results
- Thresholded individual subject activation maps are potentially affected, depending on the task timing and on the scanner
  - ★ The biggest effect of serial (AKA *temporal*) correlation—when this correlation is significant—is on the estimates of the ***variance*** of the individual subjects' $\beta$s
  - ★ If the variance is under-estimated using **`3dDeconvolve`**, then the individual subject $t$- and $F$-statistics will be over-estimated
  - ★ Individual subject variances and statistics are not usually carried forward to the group analysis level
    - Since inter-subject variance is much larger than intra-subject variance
  - ★ Thus, group results are only marginally affected by serial correlation

# 3dDeconvolve and Ordinary Least Squares (OLSQ)

- OLSQ = consistent estimator of FMRI time series fit parameter vector $\beta$
  - ★ No matter what the temporal (AKA serial) correlation structure of the noise
    - o "Consistent" means that if you repeated the identical experiment infinitely many times, and averaged the estimated value (*e.g.*, $\beta$; variance), result would be the true value
- But OLSQ estimate of time series noise ***variance*** is not consistent when serial correlation is present
  - ★ OLSQ variance estimator will usually be biased too small with serial correlation
- Variance estimate is in denominators of formulas for $t$- and $F$-statistics
  - ★ Result: individual subject $t$- and $F$-values will be too large and/or their DOF parameters will be too large
  - ★ Upshot: Significance of individual subject activations will be over-estimated ($p$-values will be too small)
  - ★ Thresholded individual subject FMRI maps might show too much activation
  - ★ Obvious impacts on ROIs generated directly from individual subject activation maps (*e.g.*, for connectivity analysis)
  - ★ However, statistics taking into account serial correlation can be too conservative, and understate the extent of the "true" regions of activation
    - o For this reason, *and* to avoid selection bias, perhaps it is best to define FMRI-derived ROIs using a spherical "punch out" around each activation map peak

# A Tiny Amount of Mathematics

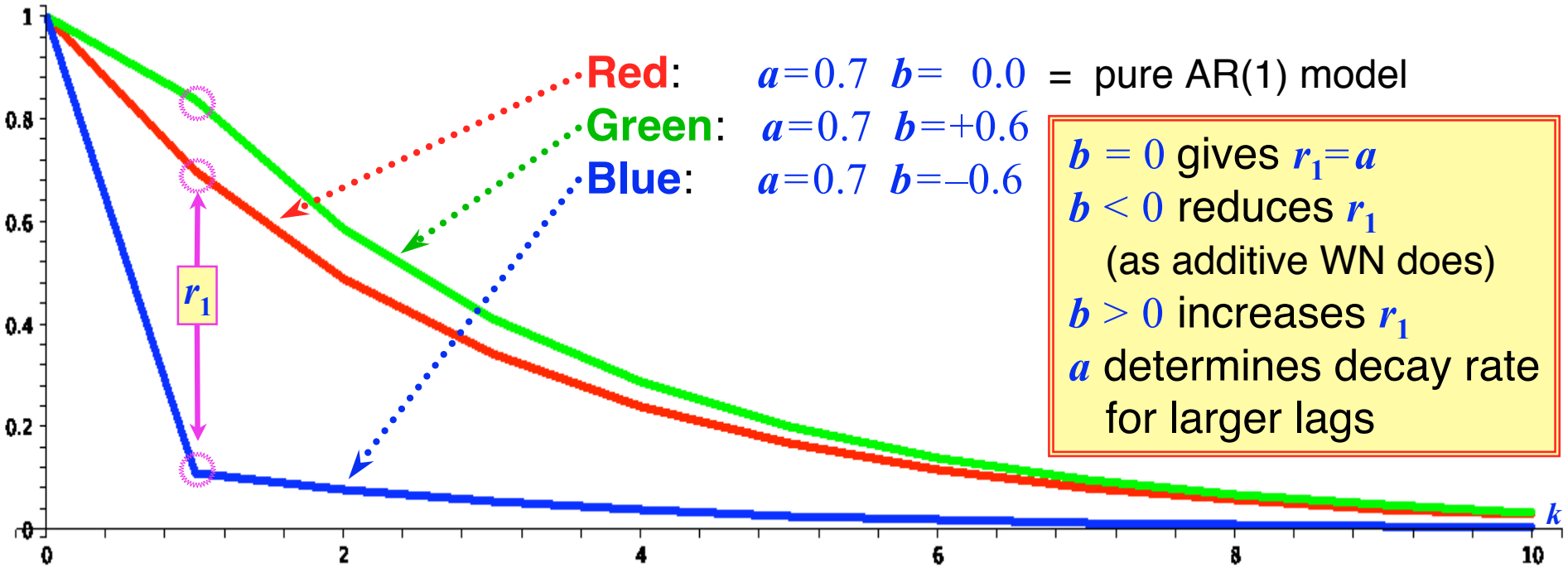
- White noise estimate of variance:

$$\hat{\sigma}^2 = \frac{1}{N-m}\sum_{i=0}^{N-1}[\text{data}_i - \text{fit}_i]^2$$

  - ★ $N$ = number of time points; $i$ = time index
  - ★ $m$ = number of fit parameters
  - ★ $N-m$ = degrees of freedom (DOF) = how many equal-variance independent random values are left after the time series is fit with $m$ regressors
    - o OLSQ assumption is that each of the $N$ noise values in the data time series are equal-variance and independent (AKA white noise)
- If noise values *aren't* independent, then $N-m$ is too large an estimate of DOF, so variance estimate is too small
- Two possible solutions are:
  1) Adjust variance estimate (and so the $t$- and $F$-values) to allow for too few DOF
  2) Come up with a different variance estimator that has all $N-m$ DOF possible
     - o Requires estimating the temporal correlation structure of the noise as well
     - o Once temporal correlation matrix is known, use Generalized Least Squares (GLSQ; AKA pre-whitening) to estimate $\beta$ parameter vector
     - o GLSQ is consistent and should produce $\beta$-values with smaller variance than OLSQ
- Solution #2 is what **`3dREMLfit`** implements

# Mathematical Model for Serial Correlation

- My choice: ARMA(1,1) = **A**uto**R**egressive order 1 **+** **M**oving **A**verage order 1
  - ★ Notation: $r_k$ = correlation at time lag #$k$ for $k$=1,2,…,$N$-1
- parameter $a$ = decay rate of the $r_k$ as $k$ increases: for FMRI, $0 \le a < 1$
- parameter $b$ = affects correlation at lag 1 ($r_1$): $-1 < b < 1$
  - ★ $r_1 = (a+b)\cdot(1+a\cdot b)/(1+2a\cdot b+b^2)$ $r_k = a^{k-1} r_1$ for $k = 1,2,...$
- For $a > 0$ and $-a < b < 0$, ARMA(1,1) noise can be thought of as a sum of AR(1) noise and white noise, with variance proportions determined by $b$
  - ★ Why I prefer 2 parameter ARMA(1,1) over easier 1 parameter AR(1) model ($b$=0)



# New Program: `3dREMLfit`

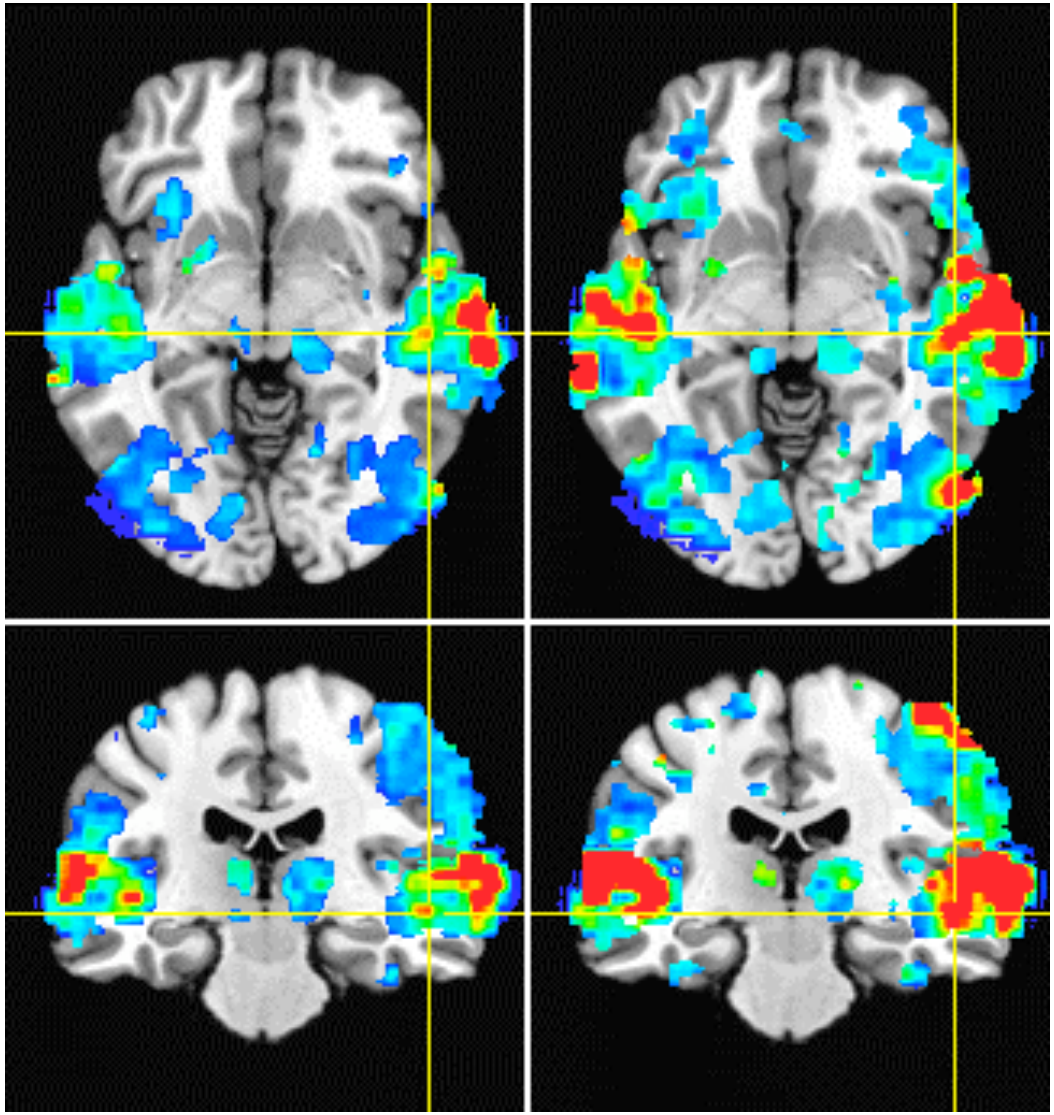
- Implements Solution #2: estimate correlation parameters and use GLSQ
  - ★ **REML** is a (partially nonlinear) method for simultaneously estimating variance + correlation parameters ***and*** estimating regression fit parameters ($\beta$s)
  - ★ *Each voxel* gets a separate estimate of its own correlation parameters $(a,b)$
    - o Estimates of $a$ and $b$ can be spatially smoothed before they are used to compute the $\beta$s
    - o Can also input $a$ and $b$ directly and skip their estimation (the slow part), if desired, and use *those* values to compute the $\beta$s
    - o Variance estimate uses pre-whitened residuals to keep DOF=$N-m$
  - ★ Even if correlation decay parameter $a$ was the same for all voxels, relative amount of white noise (measured by $b$) mixed in would vary spatially
    - o Sample analyses using 1-parameter AR(1) and MA(1) models shown later
- Inputs to `3dREMLfit`
  - ★ Run `3dDeconvolve` first to setup `.xmat.1D` matrix file, GLTs, etc.
    - o Don't have to let `3dDeconvolve` finish analysis: `-x1D_stop`
    - o `3dDeconvolve` also outputs a command line to run `3dREMLfit` with the same 3D+time dataset and the matrix file just created
  - ★ Then, input matrix file and 3D+time dataset to `3dREMLfit`
- Output datasets are structured to be similar to those in `3dDeconvolve`
  - ★ It should be easy to adapt scripts that use `3dDeconvolve` output files (*e.g.*, for group analysis) to use the new software

# Rapid Event Related Design (NIH 3 T: JJY)

**Individual** Maps from 17 Subjects

- Color map & Threshold**:** Full $F$ such that $p$=0.001 (Underlay = TT_N27+tlrc)

**REML**
$F$=3.35
$p$=0.001



**OLSQ**
$F$=3.35
$p$=0.001

GIF Animation:
time = subject
Not visible in PDF

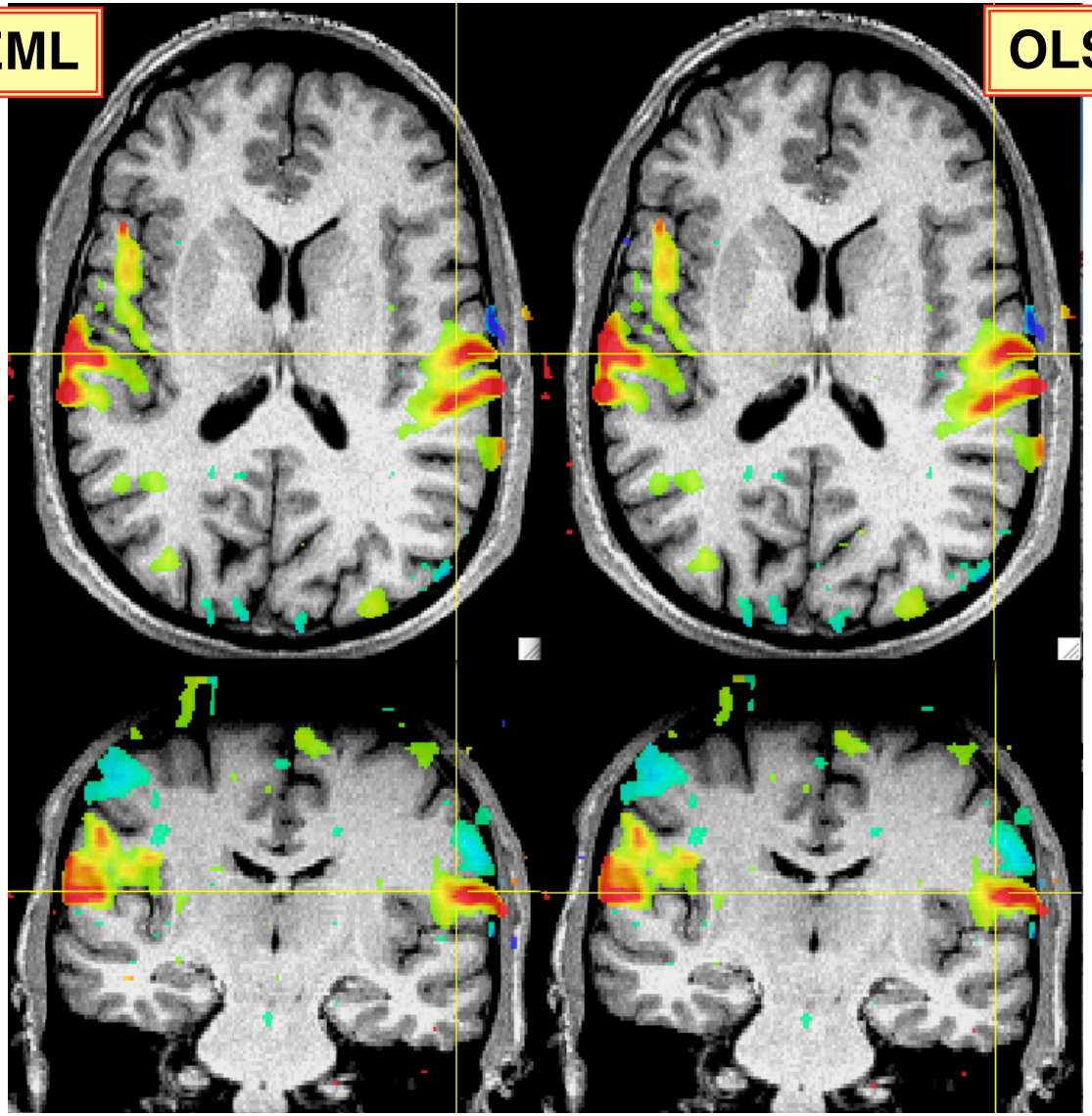
Differences between REML and OLSQ are noticeable with rapid event-related design (but activated regions are very similar)

# Block Design (15 s blocks: FBIRN-1 SM Task)

## 1 **Individual** Map (Subject#106)

Color=% signal change; Threshold: $p$=0.05 (uncorrected)

**REML** **OLSQ**



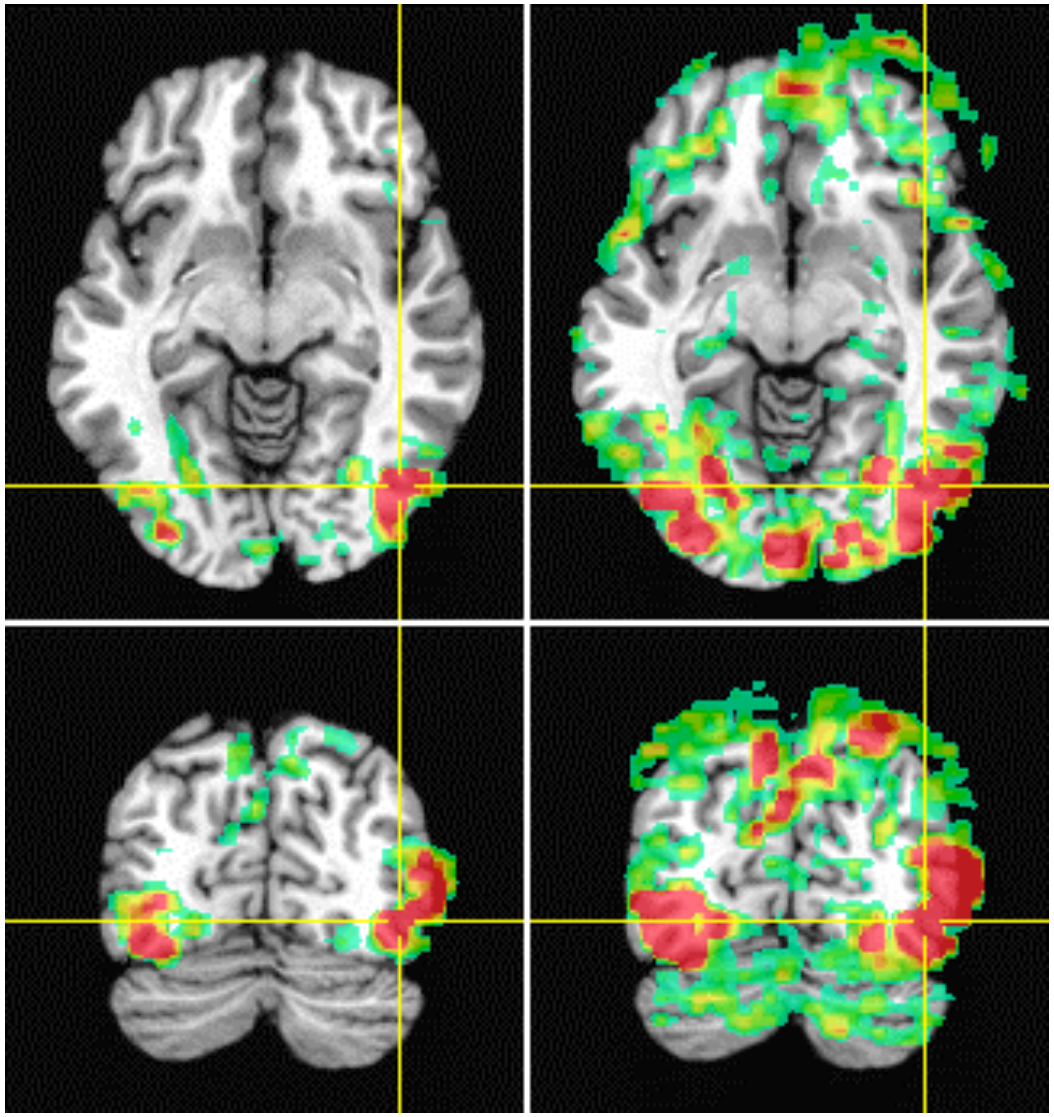
- Very little difference between OLSQ and REML, even at so low a threshold
- Data is markedly less correlated in time (UNM Siemens 1.5 T), as shown by maps of REML-estimated $r_1$
- Similar data from U Iowa GE 1.5 T has similarly low temporal correlation
- BWH & MGH 3 T data has higher temporal correlation than FBIRN 1.5 T, but lower than NIH 3 T —— **???**

# Block Design (30 s blocks: NIH 3T; JJY)

**Individual** Maps from 16 Subjects

- Color map & Threshold**:** Full $F$ such that $p$=0.001 (Underlay = TT_N27+tlrc)

**REML**
$F$=3.15
$p$=0.001



**OLSQ**
$F$=3.15
$p$=0.001

GIF Animation:
time = subject
Not visible in PDF

This is the **worst situation** for OLSQ: stimulus is at very low frequencies, where noise correlation affects variance the most

# Results Thus Far

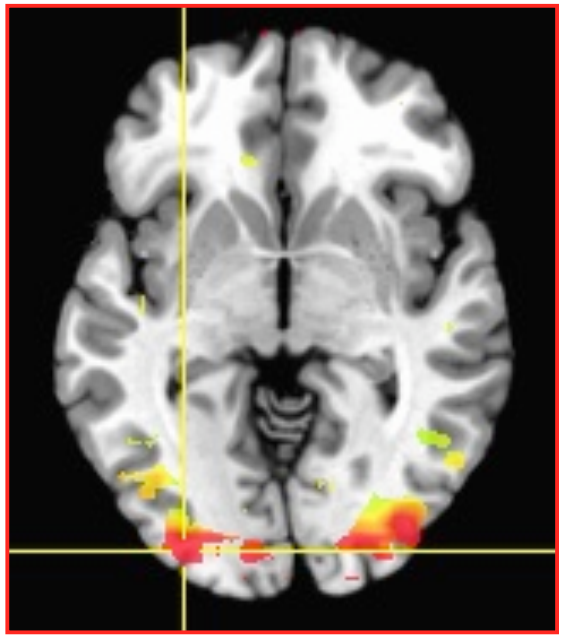
- Between OLSQ and GLSQ+REML:
  - ★ Individual subject thresholded activation maps may differ very little, some, or a lot
- Level of temporal correlation determines how much difference GLSQ makes to individual subject statistics
  - ★ Amount of temporal correlation seems to depend on magnetic field strength, other scanner details, pulse sequence, …
  - ★ Effect of temporal correlation also seems to depend on stimulus timing
  - ★ As theory indicates:
    - o Temporal correlation means noise variance depends on frequency
    - o So amount of noise that interferes with ("looks like") the signal will depend on frequencies at which the hemodynamic response is appreciable
- Next slides: Group activation maps, GLSQ+REML vs OLSQ
  - ★ 2 cases from NIH: Event-related and Block:30s designs
  - ★ Don't have enough FBIRN-1 subjects to do a group analysis

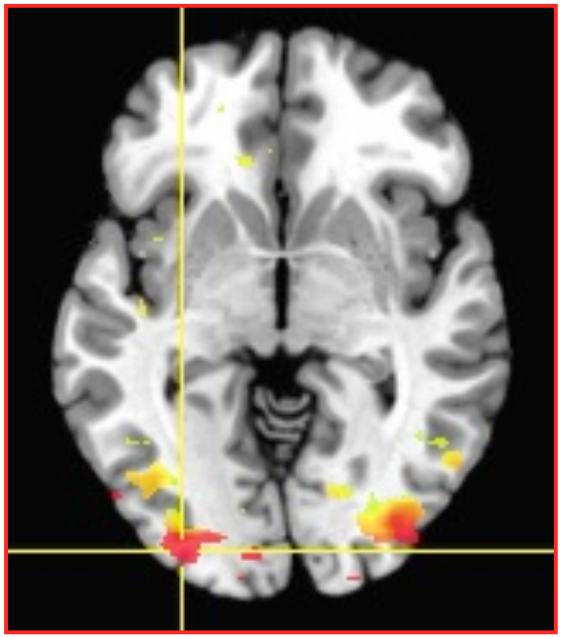
# Block Design: Group Results (3dANOVA3)

**REML**

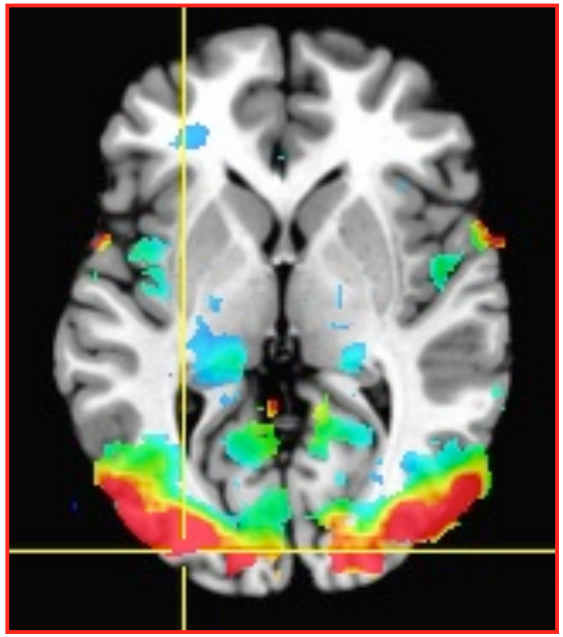
**OLSQ**

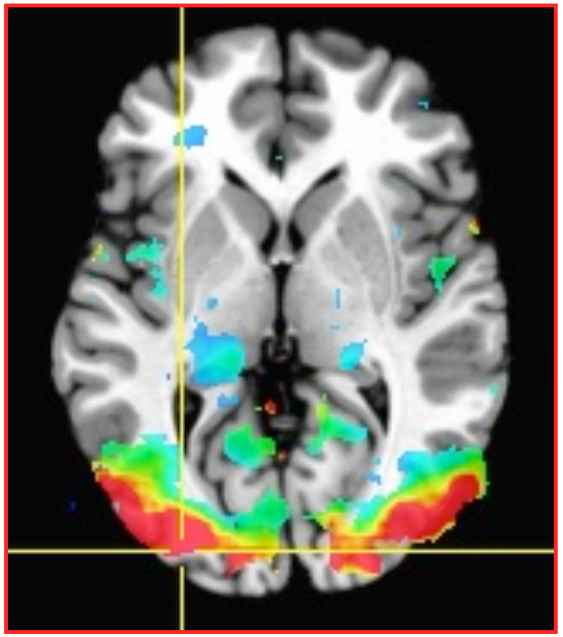
***F*-test for Affect condition**





***F*-test for Category condition**



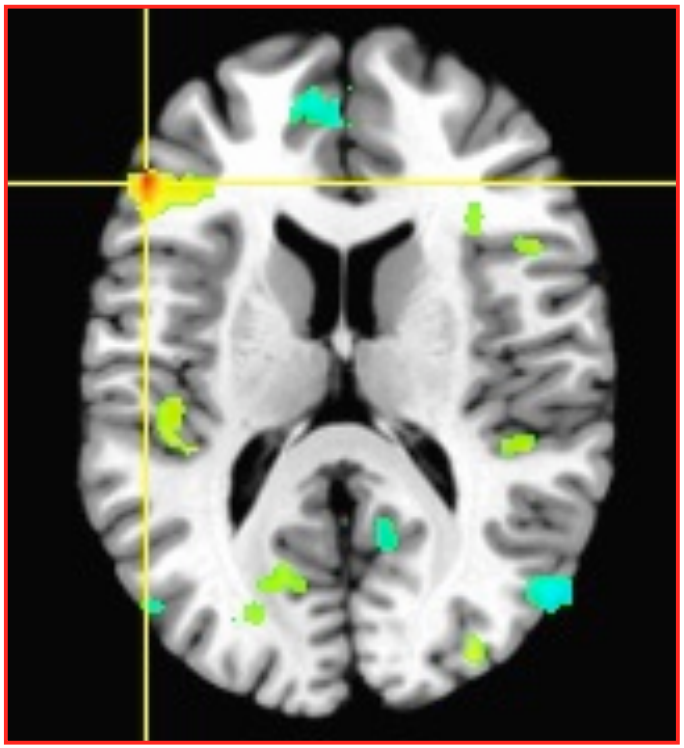


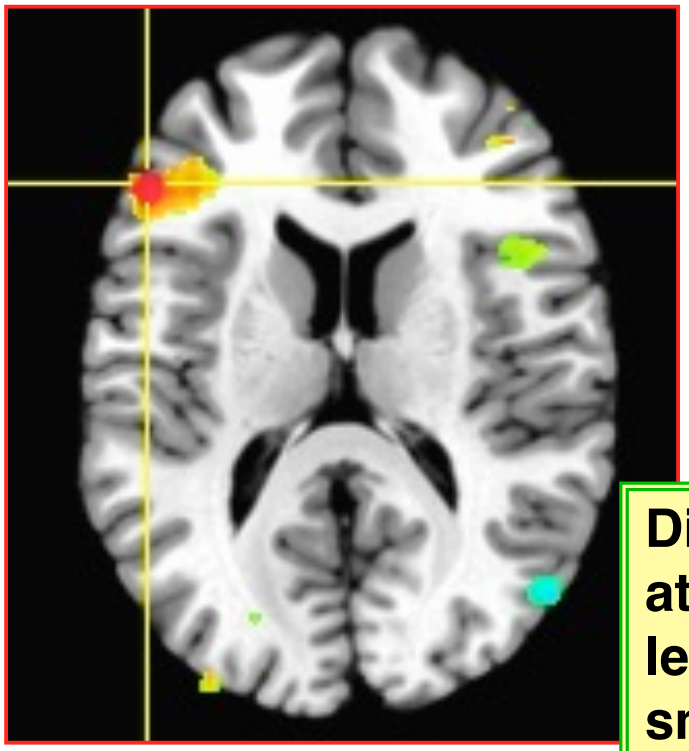
**Differences at group level are small:**

∃ Many false negatives in individual maps when using more conservative GLSQ+REML**?**

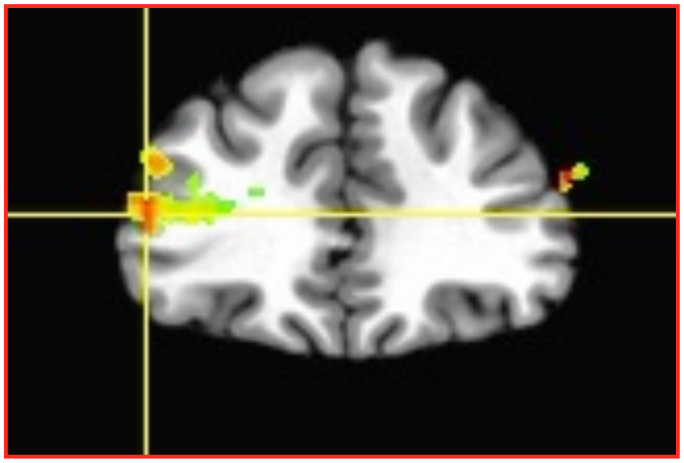
# Event-Related Design: Group Results (3dANOVA3)

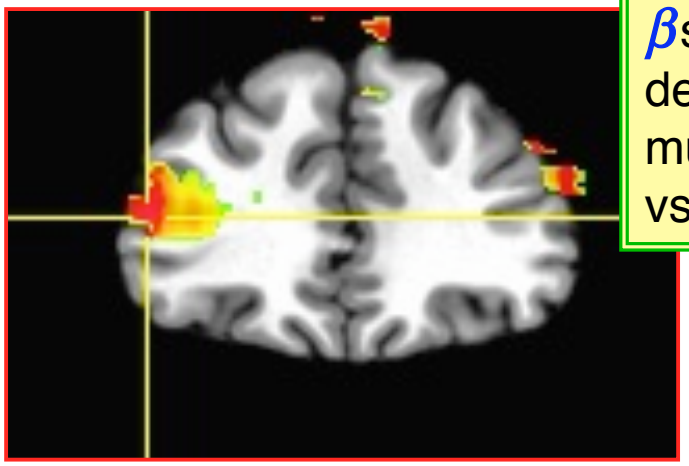
**REML**





**OLSQ**





**Differences at group level are small:**

$\beta$s don't depend very much on REML vs OLSQ

# Tentative Conclusions

- For individual subject thresholded activation maps:
  - ★ Use GLSQ/REML estimation, especially for slow block design experiments at 3+ Tesla
  - ★ Be aware that there may be many false negatives
    - o i.e., false acceptances of the null hypothesis
    - o am looking into an FDR-like procedure for estimating the false negative rate, similar to how FDR estimates the false positive rate
- For group maps using ANOVA (or similar statistics):
  - ★ Differences between OLSQ and GLSQ estimation are small
- Recommendations:
  - ★ **Don't need to re-visit group activation conclusions*!***
  - ★ Use **`3dREMLfit`** as a near drop-in replacment for **`3dDeconvolve`** for future work
    - o A little extra CPU time (usually from 1..3 times as long)

# Outline of SPM and FSL Approaches

- SPM5 and SPM2
  - ★ Estimate fixed **ARMA(1,1)** (more precisely, AR(1)+white noise) model for all "voxels of interest" (pass an OLSQ $F$-test)
    - o By averaging estimated auto-covariance matrix from OLSQ residuals over these voxels
    - o SPM assumes AR parameter $a \approx 0.2$, and approximates ARMA(1,1) correlations via linear Taylor series, to make correlation parameter estimation easier to program
  - ★ Use GLSQ (same for each voxel) to solve for $\beta$s
    - o SPM99: Use OLSQ and adjusts DOF downwards to allow for serial correlation
- FSL and FMRIstat (similar, but differ in important details at several points)
  - ★ Use OLSQ to get first-pass residuals; use these to estimate each voxel's auto-correlation matrix; smooth these matrices spatially (FSL & FMRIstat vary here)
  - ★ Estimate **AR(1)** parameter for each voxel separately from smoothed matrices
  - ★ Use GLSQ (different for each voxel) to solve for $\beta$s
- All these programs use a non-REML method to estimate serial correlation parameter(s) from the OLSQ residual auto-correlation matrix, and then adjust these estimates to reduce the bias thus introduced

# Using 3dREMLfit - 1

- Step 1: run **3dDeconvolve** as normal, setting up timing, GLTs
- **3dDeconvolve ... -bucket Adecon -x1D_stop**

**Screen output:**

filename re-used for **3dREMLfit** command

```
++ Wrote matrix values to file Adecon.xmat.1D
++ ========= Things you can do with the matrix file =========
++ (a) Linear regression with ARMA(1,1) modeling of serial
 correlation:

3dREMLfit -matrix Adecon.xmat.1D -input ss17.AllRuns.norm+orig
 -mask ss17_mask+orig -Rbeta Adecon_beta_REML -fout -Rbuck
 Adecon_REML -Rvar Adecon_REMLvar

++ N.B.: 3dREMLfit command above written to file Adecon.REML_cmd
++ (b) Visualization/analysis of the matrix via ExamineXmat.R
++ (c) Synthesis of sub-model datasets using 3dSynthesize
++ ================================================
++ 3dDeconvolve exits: -x1D_stop option was given
```

# Using 3dREMLfit - 2

- Step 2: run **3dREMLfit** ; perhaps adding options to the command line:
  - ★ **-addbase** : add extra baseline columns to the regression matrix
  - ★ **-slibase** : add extra baseline columns to the regression matrix, ***on a per slice basis*** = intended to aid in removal of physiological noise
  - ★ **-gltsym** : add extra GLTs (beyond those from **3dDeconvolve**)
  - ★ **-usetemp** : **-slibase** can require a lot of memory
    - o Generates REML matrices for many $(a,b)$ cases for each slice
    - o This option writes & reads temporary matrices to disk to reduce RAM usage
      - ➥ **-verb** : outputs information about memory usage as program runs
  - ★ **-Obuck** : output OLSQ bucket dataset (etc.)
    - o **-Rbuck** : output GLSQ bucket (stimulus $\beta$s and statistics)
    - o **-Rbeta** : output GLSQ (all the $\beta$s and only the $\beta$s; no statistics)
    - o **-Rfitts** : output GLSQ fitted model
    - o **-Rvar** : output GLSQ $(a,b)$ parameters and variance estimate (per voxel)
  - ★ **-NEGcor** : allow negative correlations in the estimation
    - o Probably not really needed for FMRI, but option is there just in case
    - o There are more options to control estimation of the $(a,b)$ parameters
- **Of course**: read the output of **3dREMLfit -help**

# Potential Add-ons to **3dREMLfit**

- Add option to use this program to **afni_proc.py** super-script
- Add **-iresp** and **-sresp** options
- Output variances for $\beta$s
  - ⋆ *e.g.*, to be carried to the group analysis level? Need to implement a new approach for this option to be useful.
- Matrix error checking when **-addbase** or **-slibase** is used
  - ⋆ In case the bumbling user puts in a collinear column
  - ⋆ Program cannot handle an all-zero column (unlike **3dDeconvolve**)
- Re-run with extra GLTs to be added to existing bucket
  - ⋆ Or at least have a GLT-only output option: **-Rglt** ??
- Finish work with **R Birn**'s physiological noise regressors and integrate these into time series analysis via **-slibase**
- **-jobs** option to spread load across multiple CPUs
  - ⋆ Especially loop where parameters $(a,b)$ are estimated: the slowest part
- … ???

Next: more details on ARMA vs AR vs MA

# Serial Correlation Model & Notation: ARMA(1,1)

- Denote noise value at time index $i$ by $\xi_i$ for $i$=0..$N$–1
- Variance is average (AKA expected) value of noise squared:
  - ★ $\sigma^2 = E[\xi_i^2]$ where $E[\bullet]$ means "expected value of •"
- Covariance is similar to variance, measured between different time points:
  - ★ $\Sigma_{|i-j|} = E[\xi_i \xi_j]$ which depends on time *difference* between time points $i$ and $j$
- Correlation is covariance with variance factored out
  - ★ $E[\xi_i \xi_j] = \sigma^2 r_{|i-j|}$ (with $r_0$=1)
    - o N.B.: $r_k$ measures predictability of noise value at time $j$+$k$ given value at time $j$
- For entire time series, express variance/correlation as a matrix
  - ★ $E[\xi \xi^T] = \sigma^2 \mathbf{R}$ with correlation matrix $\mathbf{R}$ having elements $R_{i,j} = r_{|i-j|}$
- Need to have a simplified model for $\mathbf{R}$ (*i.e.*, the $r_k$ for $k$=1,2,…,$N$-1)
  - ★ Otherwise, have too many parameters to estimate
  - ★ My choice: ARMA(1,1) = **A**uto**R**egressive order 1 + **M**oving **A**verage order 1
  - ★ parameter $a$ = decay rate of the $r_k$ as $k$ increases: for FMRI, $0 \le a < 1$
  - ★ parameter $b$ = determines correlation at lag 1 ($r_1$): $-1 < b < 1$
    - o $r_1 = (a+b)\cdot(1+a\cdot b)/(1+2a\cdot b+b^2)$ $r_k = a^{k-1} r_1$ for $k = 1,2,...$
  - ★ For $a > 0$ and $-a < b < 0$, ARMA(1,1) noise can be thought of as a sum of AR(1) noise and white noise, with variance proportions determined by $b$
    - o This feature is one reason I prefer ARMA(1,1) as a noise correlation model over AR(1)

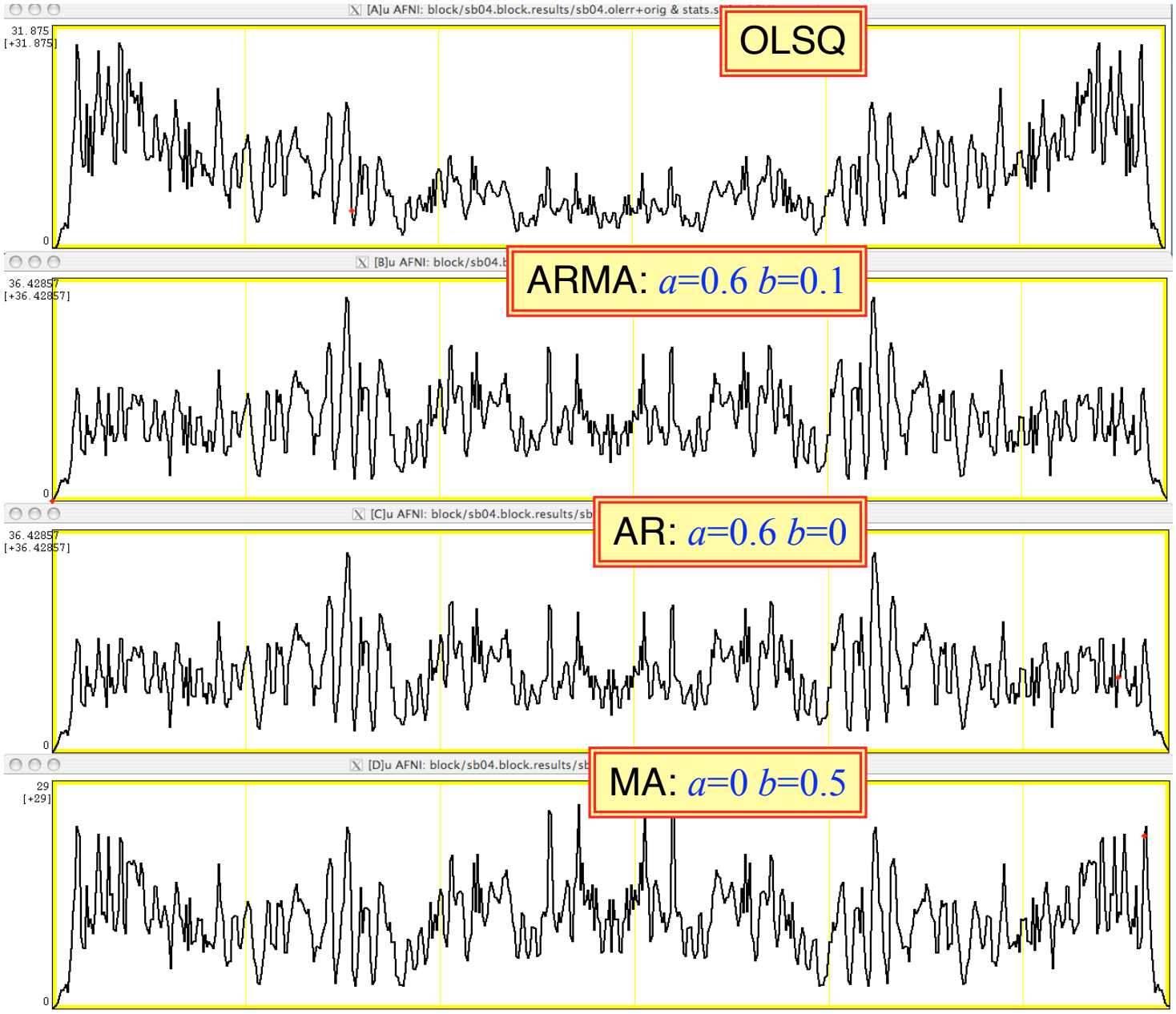
# AR(1): $a$ vs. MA(1): $b$ vs. ARMA(1,1): $a$ & $b$

- Check the effectiveness of GLSQ pre-whitening solution by examining pre-whitened residuals
  - ⋆ Pre-whitening: applying a linear transformation to the time series data to de-correlate the noise
    - o Symbolically, $\mathbf{R}^{-1/2}$ where $\mathbf{R}$ is the correlation matrix
- After pre-whitening, residuals (difference between data and fitted time series) should be (mostly) uncorrelated
- Power spectrum of white noise is flat
  - ⋆ Power spectrum = expected value of absolute value of Fourier transform, averaged over an infinity of repeated identical experiments
- Visually inspect graph of abs[FFT(pre-whitened residuals)]
  - ⋆ Should be flattish, with random excursions
    - o This is noise, after all, and we don't have an infinity of data over which to average
- Next 4 slides:
  - ⋆ Graphs of "spectrum" for OLSQ and GLSQ using ARMA(1,1), AR(1), and MA(1) correlation models (generated using interactive AFNI, of course)
  - ⋆ For 3 strongly "active" voxels in one subject (block design: 30 s blocks; NIH 3T)
  - ⋆ Then the single subject activation maps for 6 types of analysis

# Spectrum (slightly smoothed absFFT) of Residuals

**In this voxel**:

- **OLSQ**: definitely not "white"
- **GLSQ**: "white-ish" for all 3 correlation models

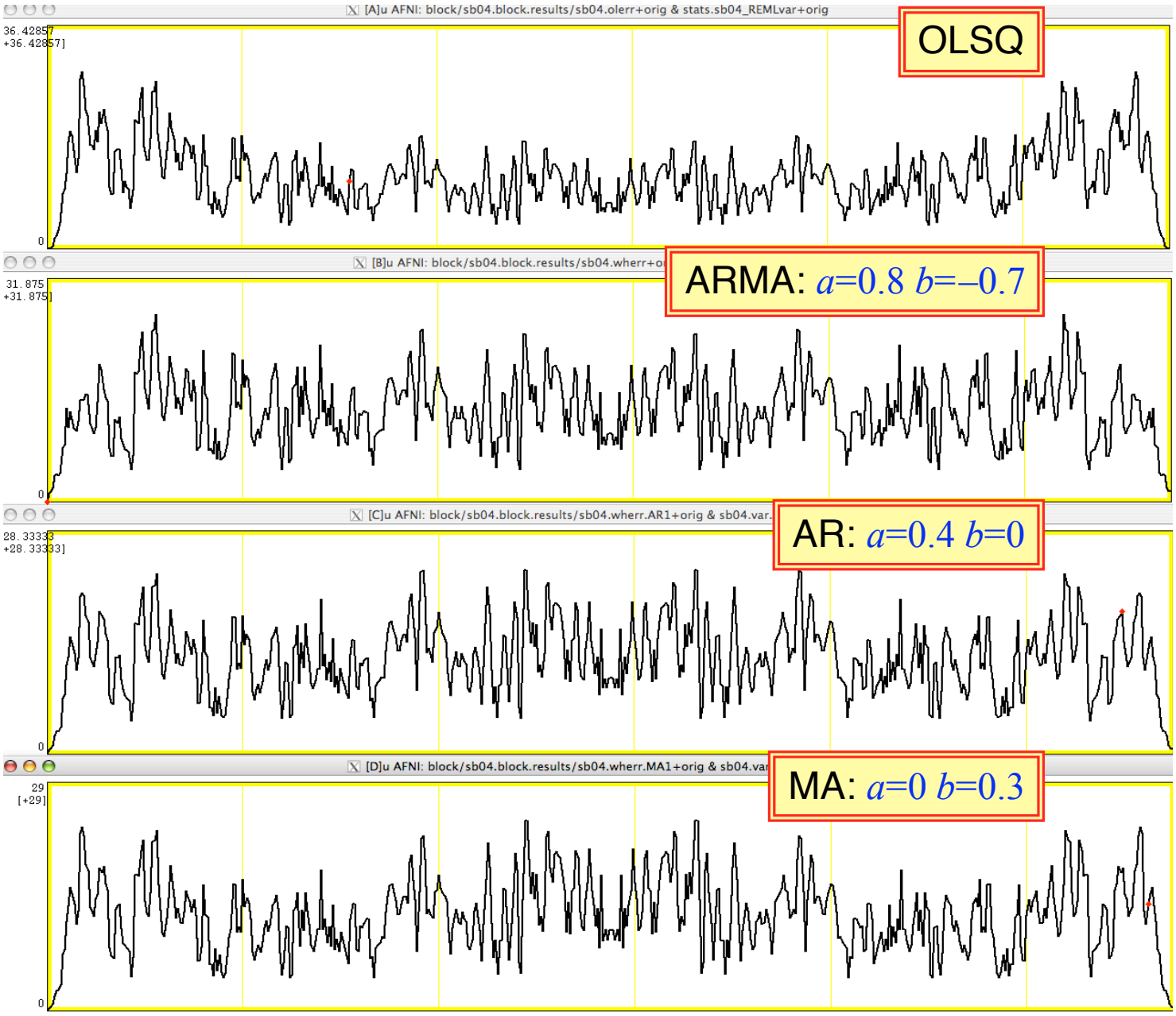



Block:30s

# Spectrum of Residuals

**In this voxel**:

- **OLSQ**: not "white" but not very "colored" either
- **GLSQ**: All methods about the same in fixing up what little needs to be fixed

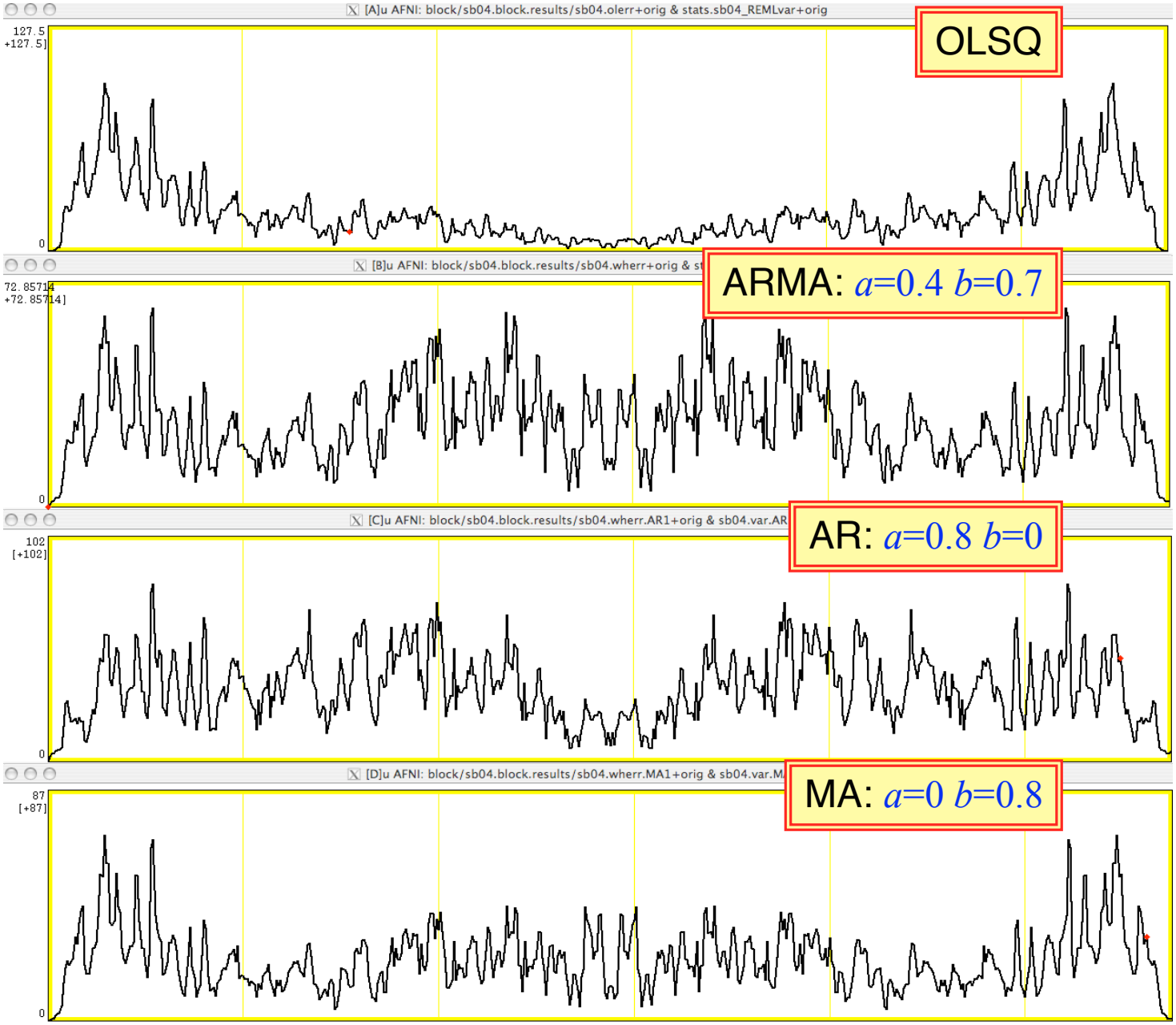



Block:30s

# Spectrum of Residuals

**In this voxel**:

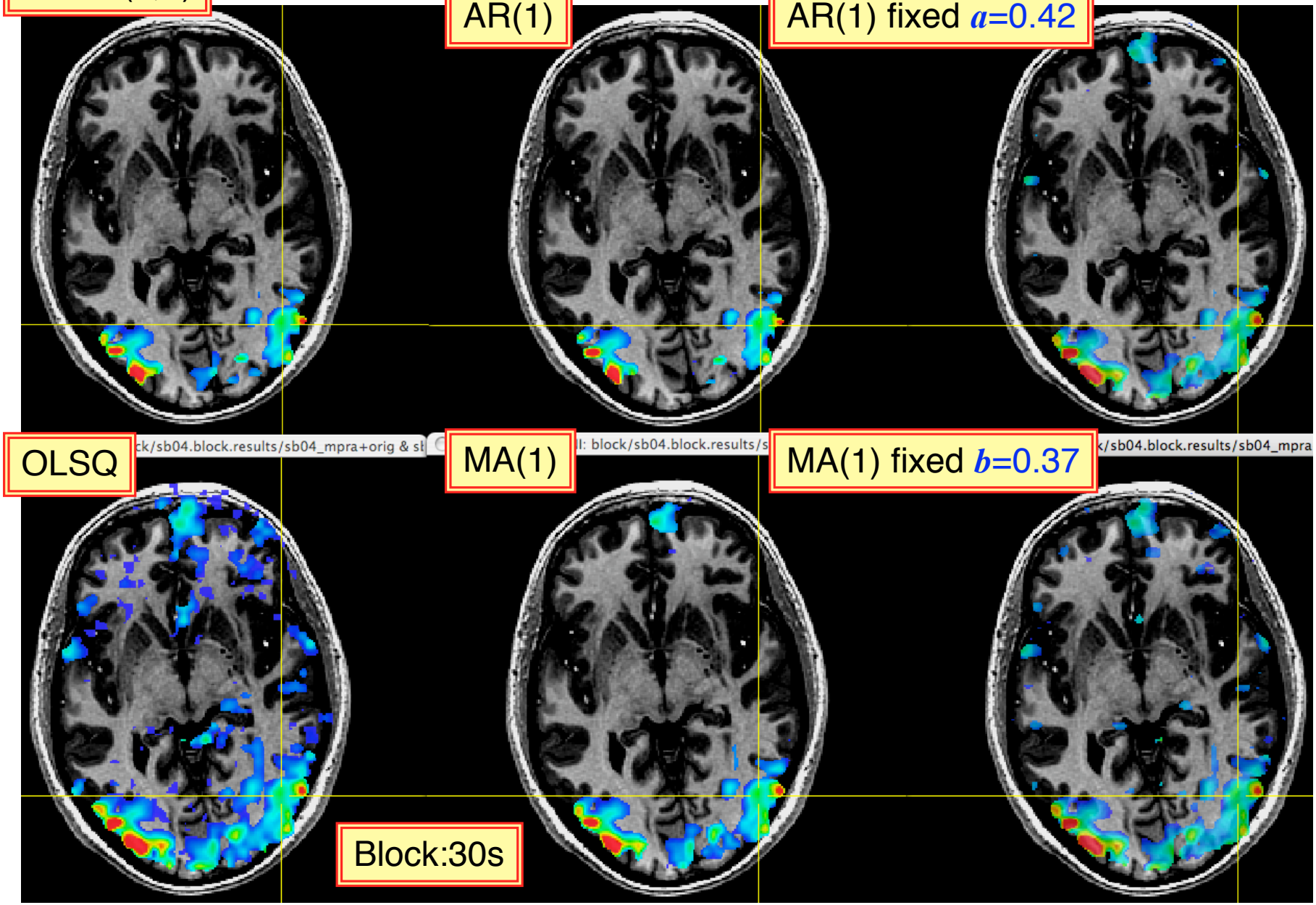
- **OLSQ**: definitely not "white"
- **GLSQ**: ARMA appears a little "whiter" than either AR or MA alone




Block:30s

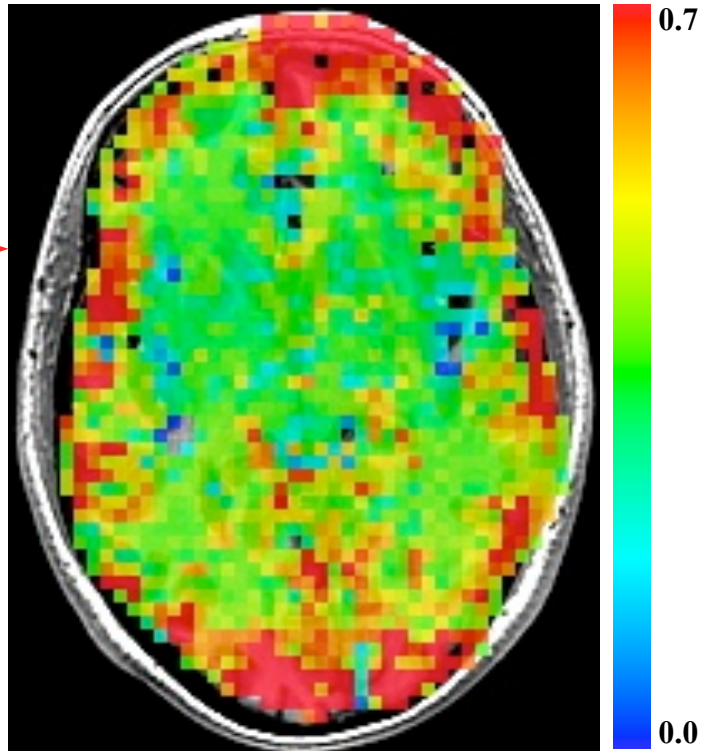
# 6 Types of Analysis

Threshold=$F$
Color=$\beta_{task\#1}$



# Conclusions from Previous Slides

- It is possible to find voxels where pre-whitening of different types (AR-only or MA-only or ARMA) is "optimal"
  - ★ And voxels where pre-whitening makes little difference
- For many (most?) voxels, the pre-whitening details don't make a lot of difference in the statistics
  - ★ As long as *something* is done that is about right
  - ★ *e.g.*, Using fixed AR(1) or MA(1) single parameter method was still OK-ish for single subject maps
    - o A few more extraneous small blobs
    - o But fewer than pure OLSQ solution statistics
- Map of $r_1$=correlation at neighboring TRs, as output by REML and ARMA(1,1) fit
  - ★ Same slice as previous slides (NIH 3 T data)
  - ★ In general, cortical gray matter shows more correlation, but this result is not universal



# Mathematics and Implementation

- Available in PDF (scanned from hand-written pages) for the truly devoted
  - ★ File **3dREMLfit_mathnotes.pdf**
- Outline of REML estimation methodology
  - ★ What is REML and why do we care?
- Matrix algebra for efficient solution of the many linear systems that must be solved for each voxel
  - ★ Sparse matrix factorizations, multiplications, and solvers
- How ARMA(1,1) parameters are estimated in **3dREMLfit**
  - ★ Optimizing REML log-likelihood function over a discrete grid of $(a,b)$ values, using 2D binary search
  - ★ Must solve a GLSQ problem for each $(a,b)$ tested, for each voxel
- How statistics are implemented as GLTs
  - ★ Testing null hypothesis $\mathbf{G}\boldsymbol{\beta}=0$ for arbitrary matrix $\mathbf{G}$
- Derivation of ARMA(1,1) formulas
  - ★ For completeness, and because we all *love* equations