fMRI 2: single participant GLM

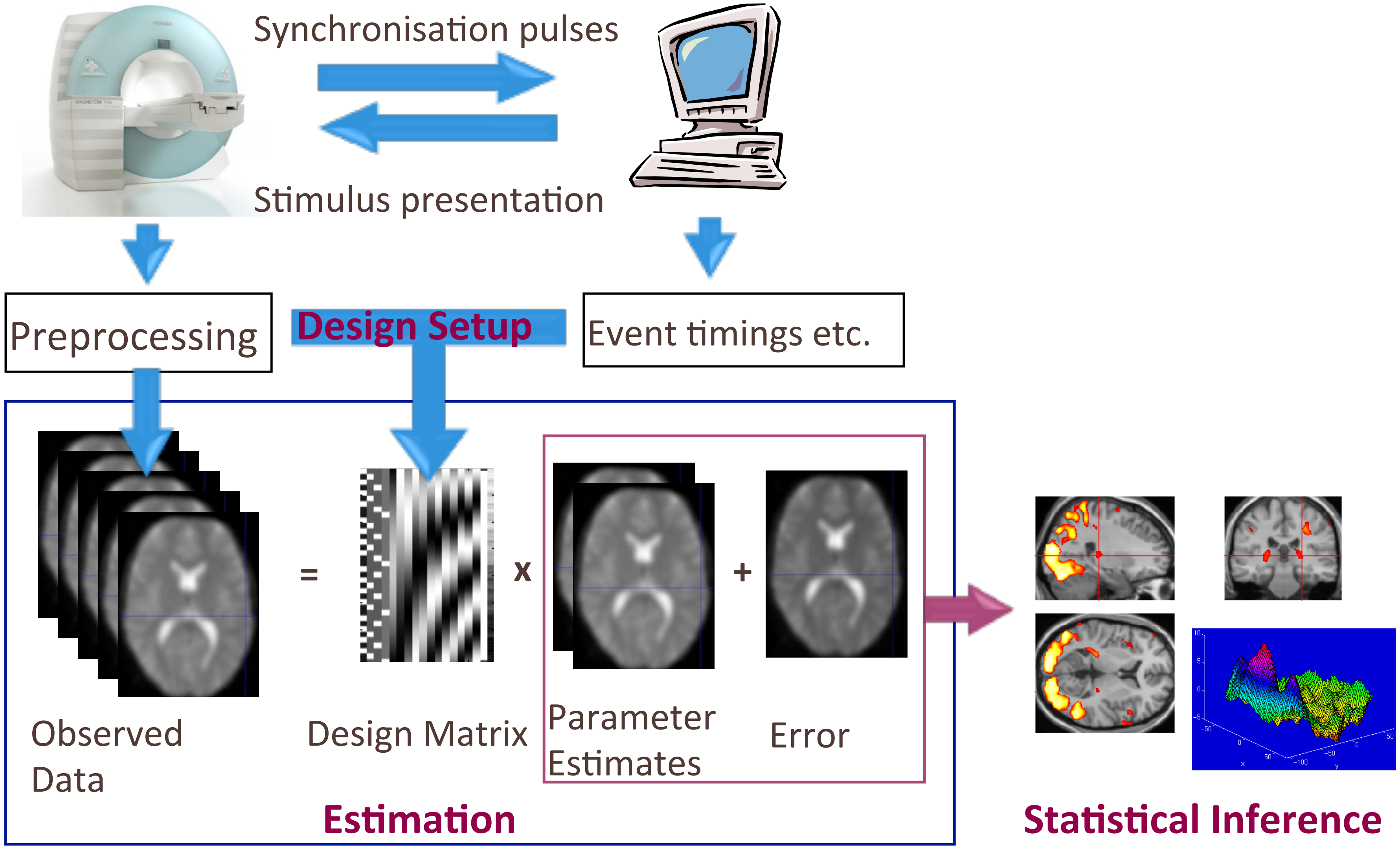
Johan Carlin
MRC Cognition and Brain Sciences Unit, Methods Group
johan.carlin@mrc-cbu.cam.ac.uk

Acknowledgment: Tibor Auer, Kendrick Kay

MRC | Cognition and Brain Sciences Unit

Outline

- How to fit fMRI responses with a general linear model
- Implications for experimental design
- Workshop: What kind of design is the most efficient?



Synchronisation pulses

Stimulus presentation

Preprocessing

Design Setup

Event timings etc.

Observed Data

=

Design Matrix

x

Parameter Estimates

+

Error

Estimation

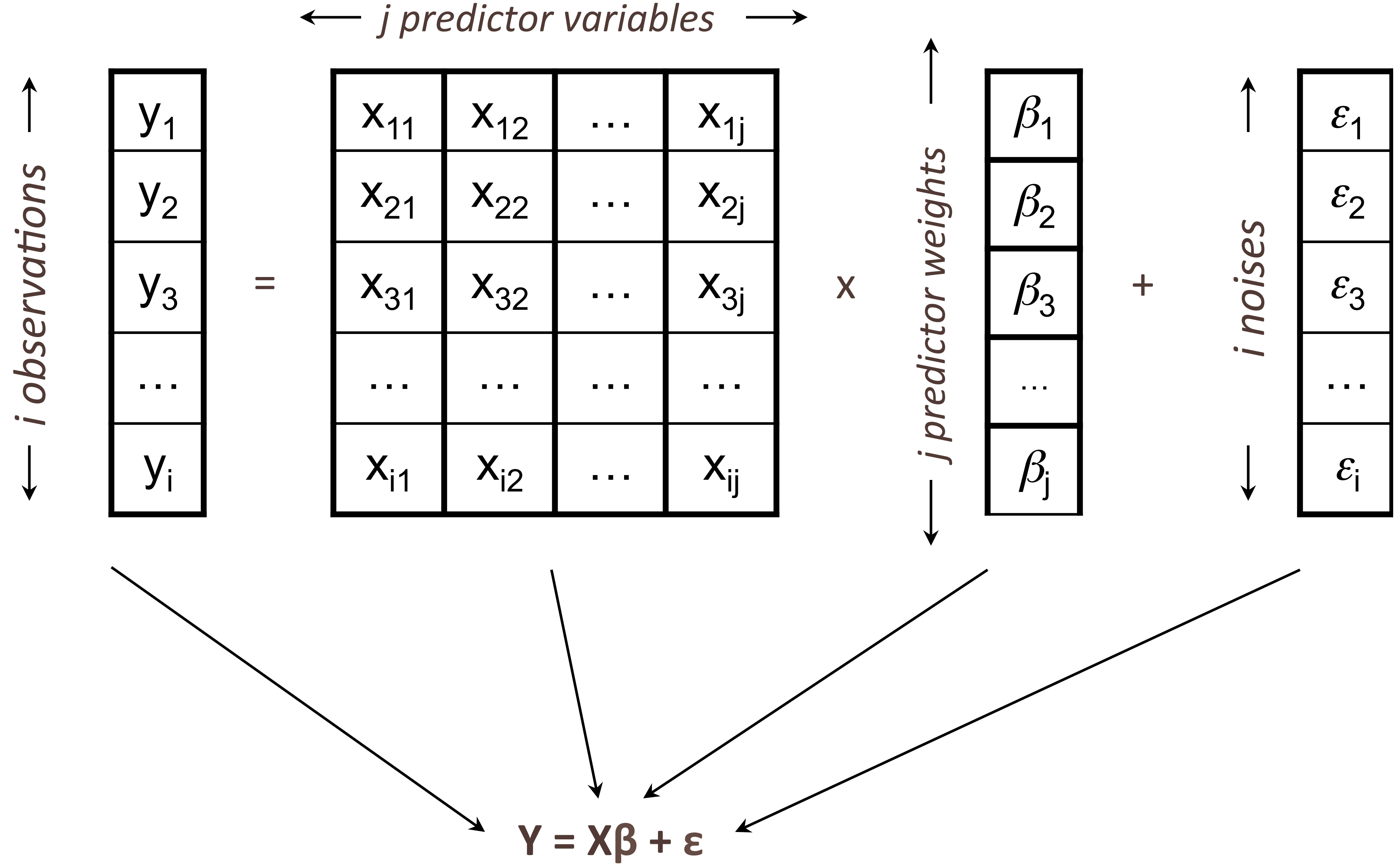
Statistical Inference

last session

today

next session

The general linear model



j predictor variables

i observations

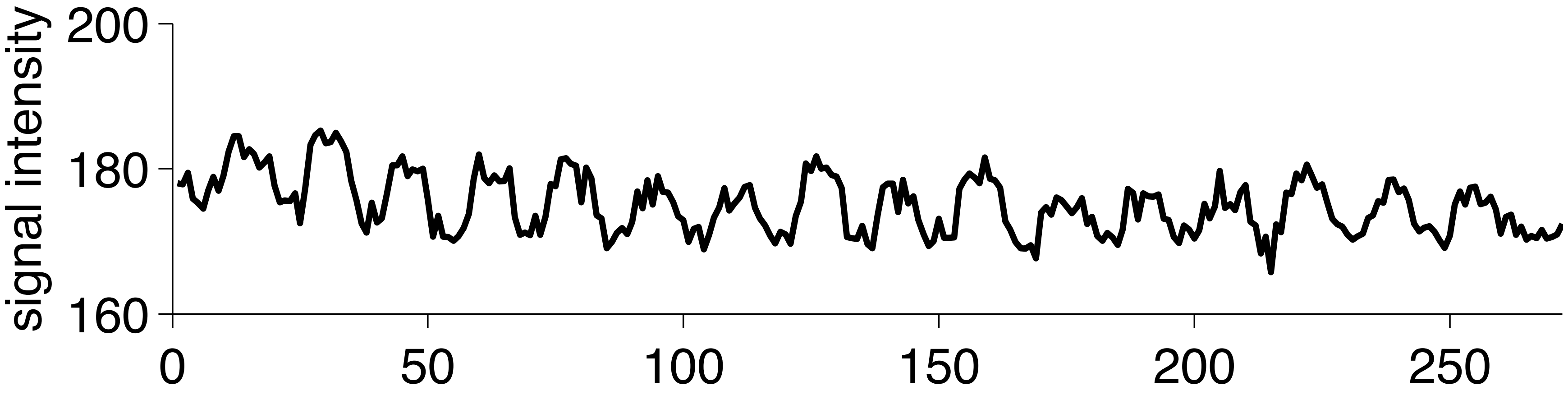
j predictor weights

i noises

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_i \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} \\ x_{21} & x_{22} & \dots & x_{2j} \\ x_{31} & x_{32} & \dots & x_{3j} \\ \dots & \dots & \dots & \dots \\ x_{i1} & x_{i2} & \dots & x_{ij} \end{bmatrix} \text{x} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \dots \\ \beta_j \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \dots \\ \varepsilon_i \end{bmatrix}$$

Y = Xβ + ε

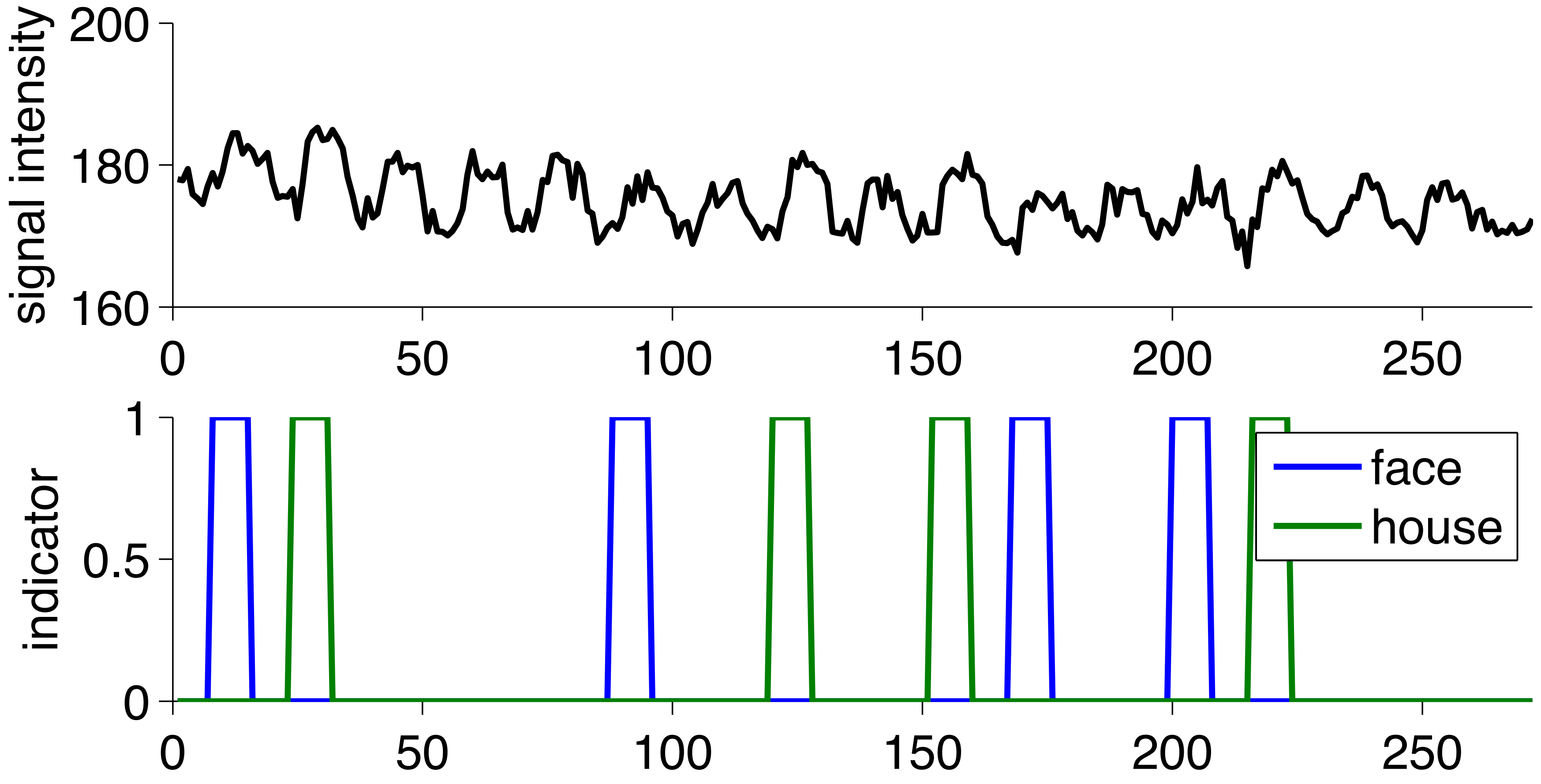
A typical fMRI task



- Face blocks appeared at volumes 8, 88, 168, 200
- House blocks at volumes 24, 120, 152, 216
- All blocks lasted 8 volumes

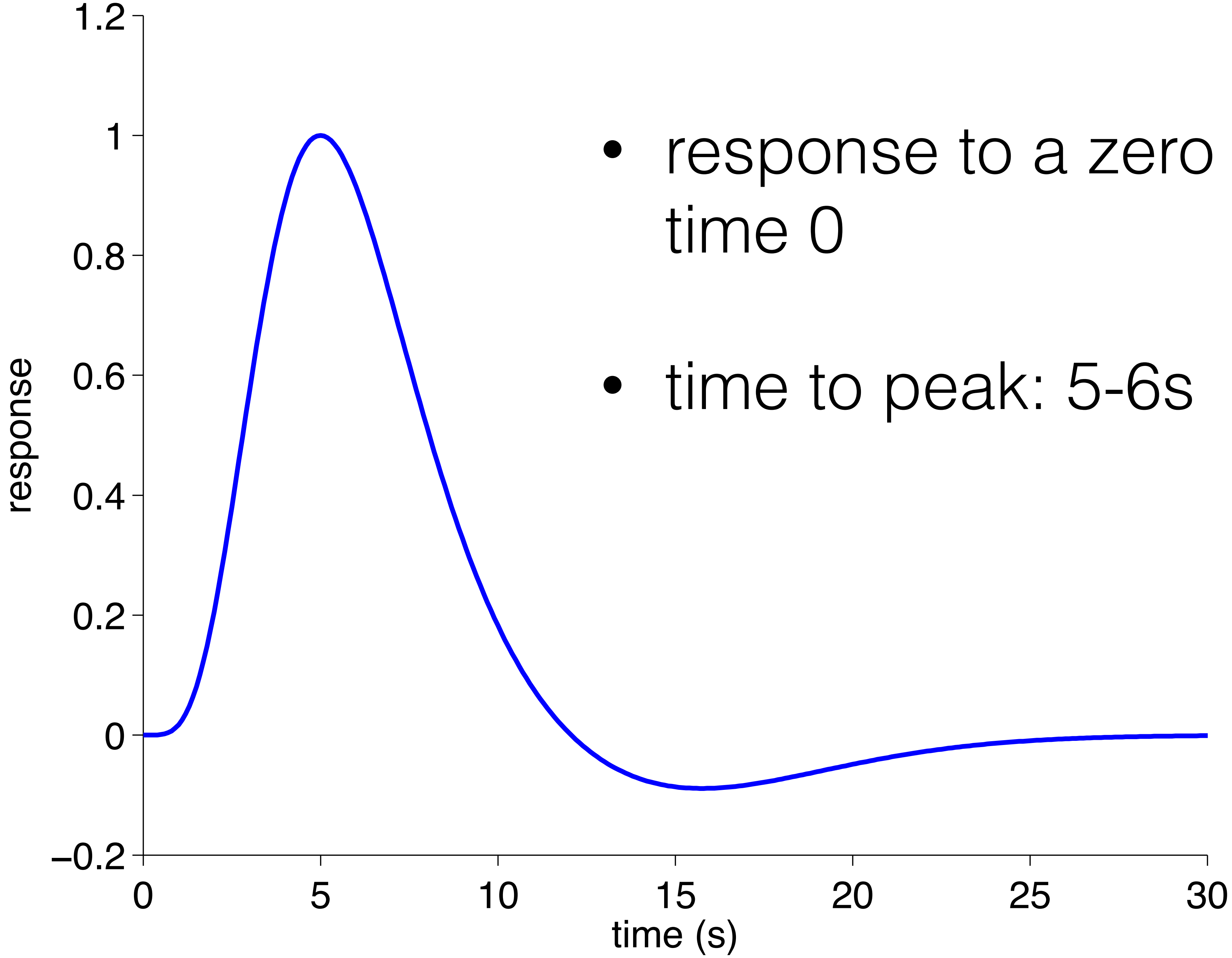
time (scans)

A typical fMRI task



But fMRI responses are not instantaneous…

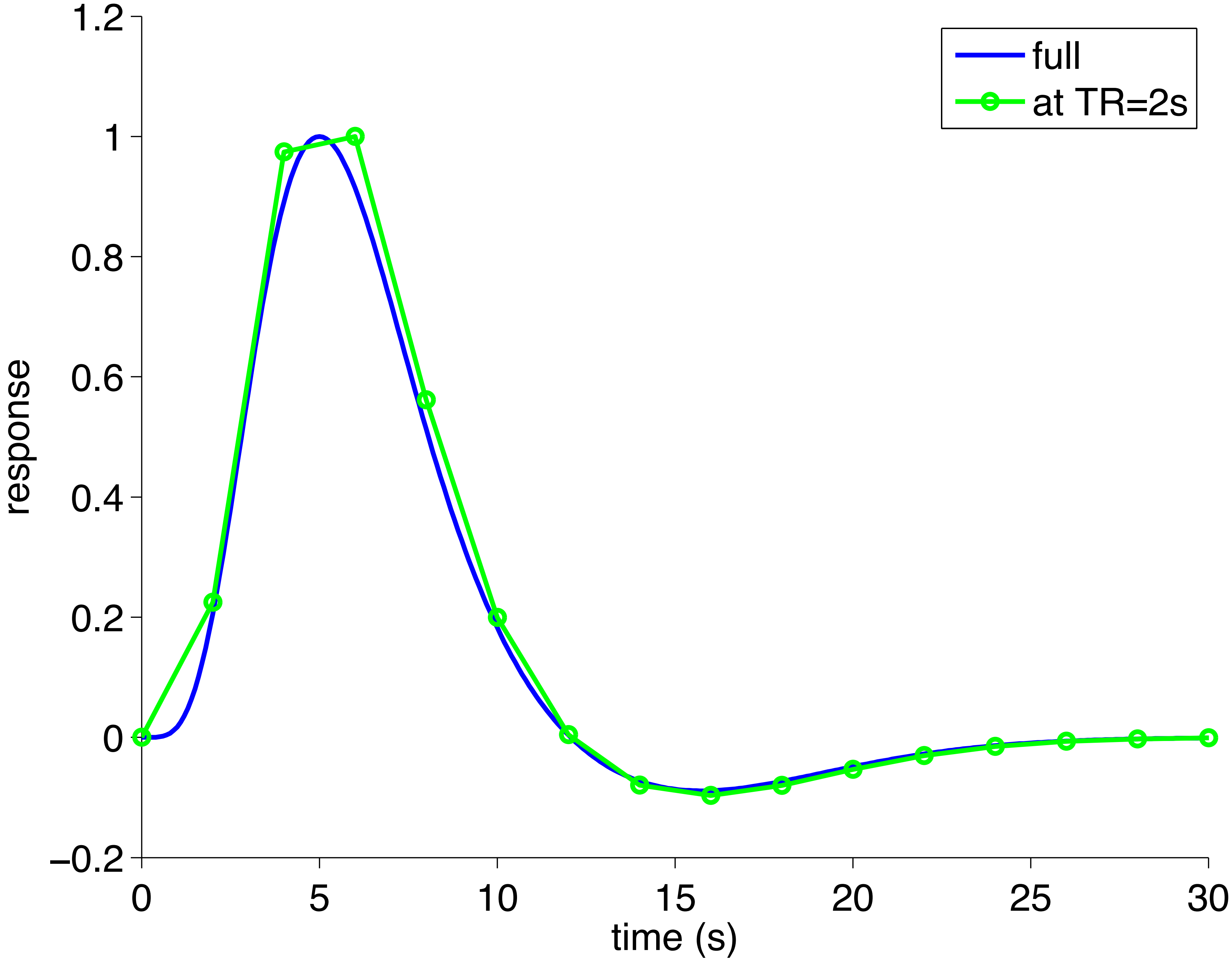
The canonical SPM HRF



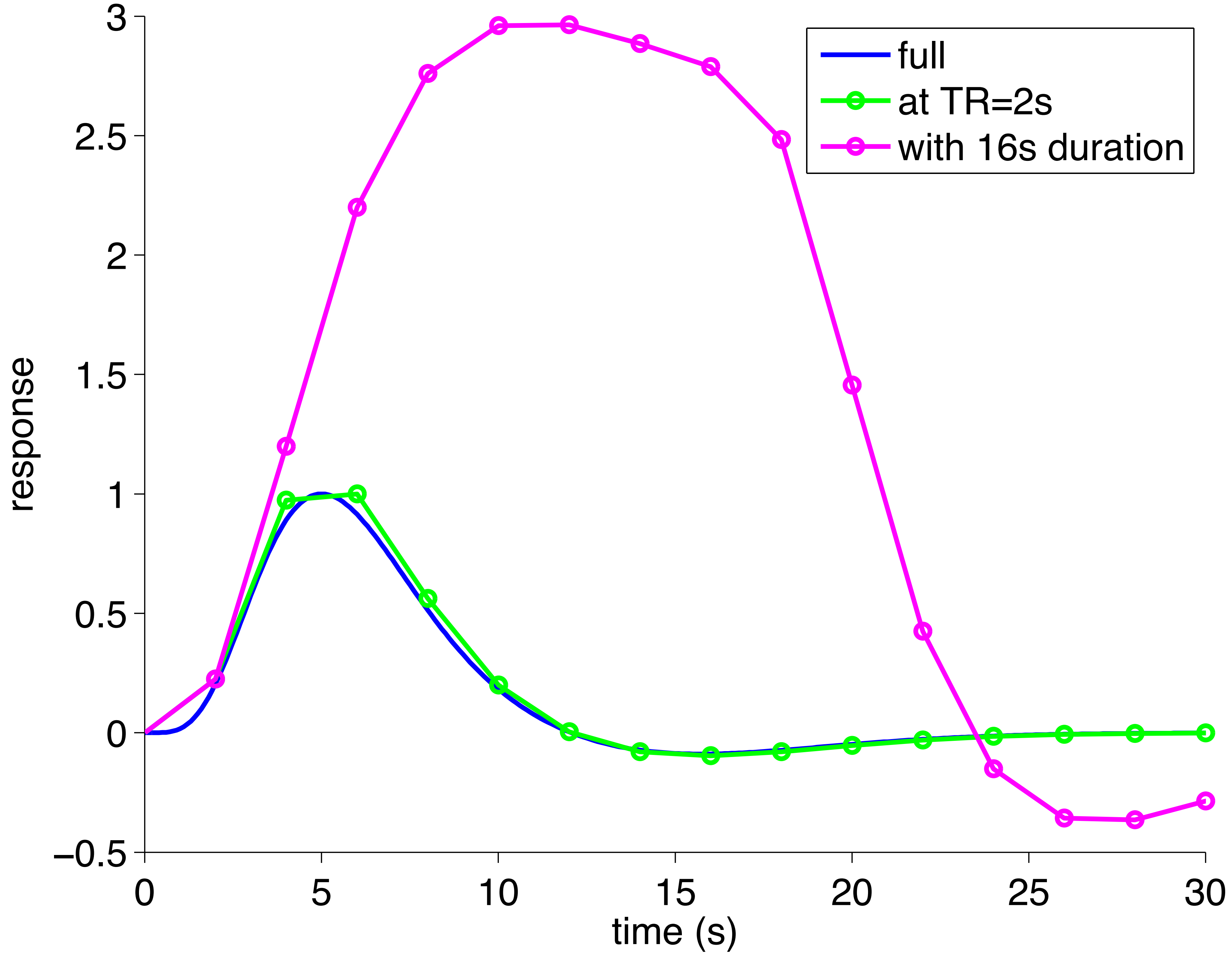
- response to a zero duration stimulus at time 0

- time to peak: 5-6s

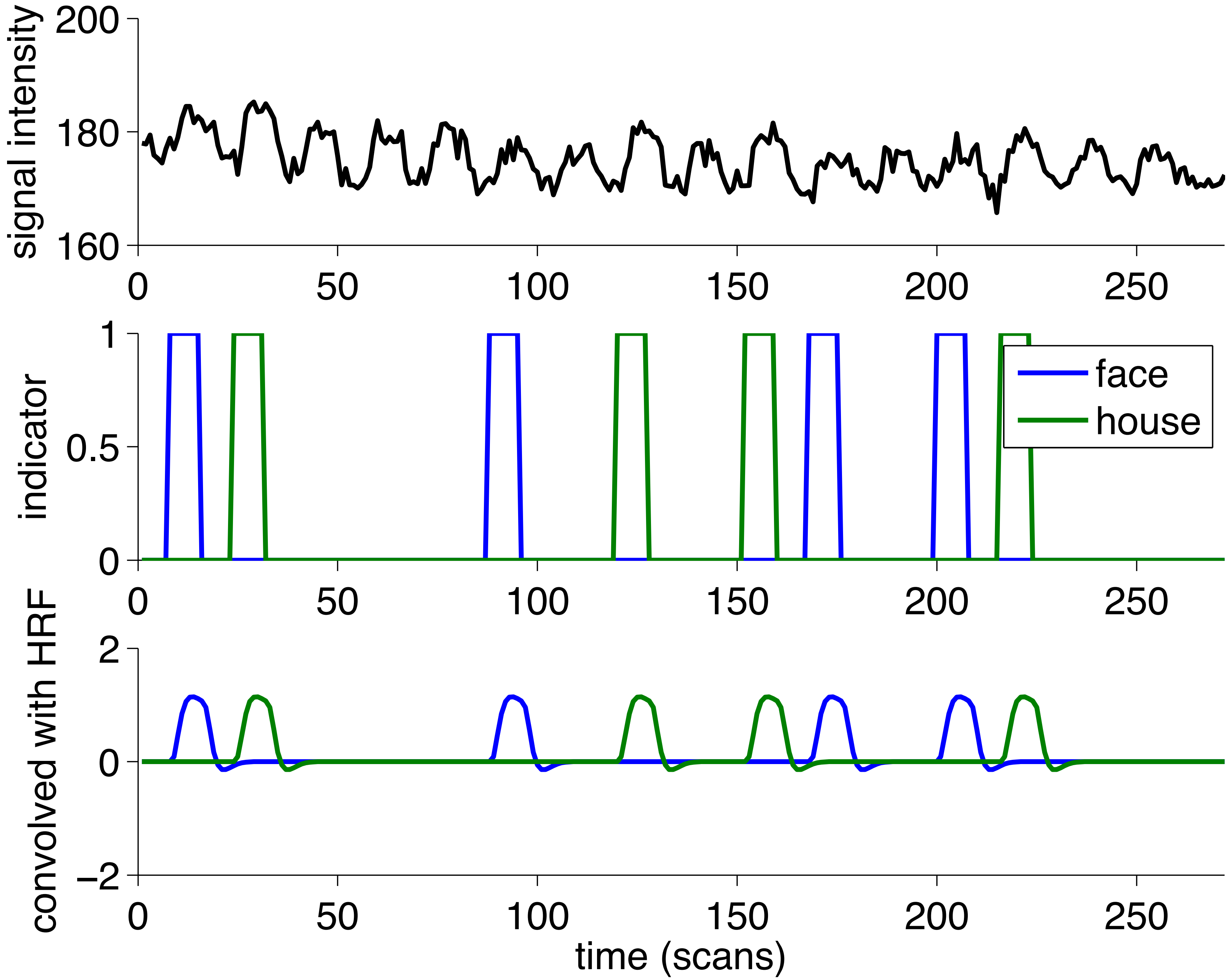
The canonical SPM HRF



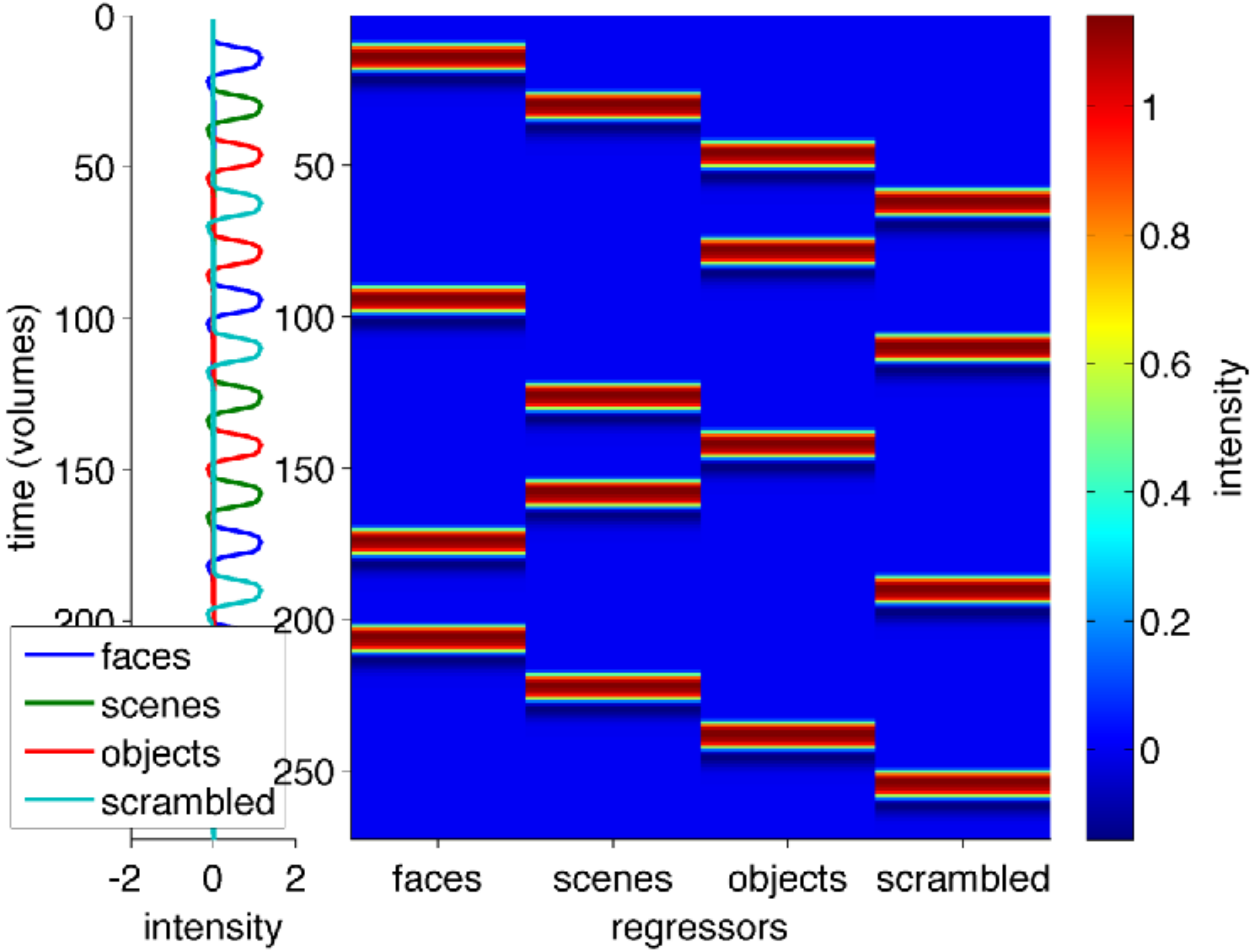
The canonical SPM HRF



A typical fMRI task



The convolved design matrix

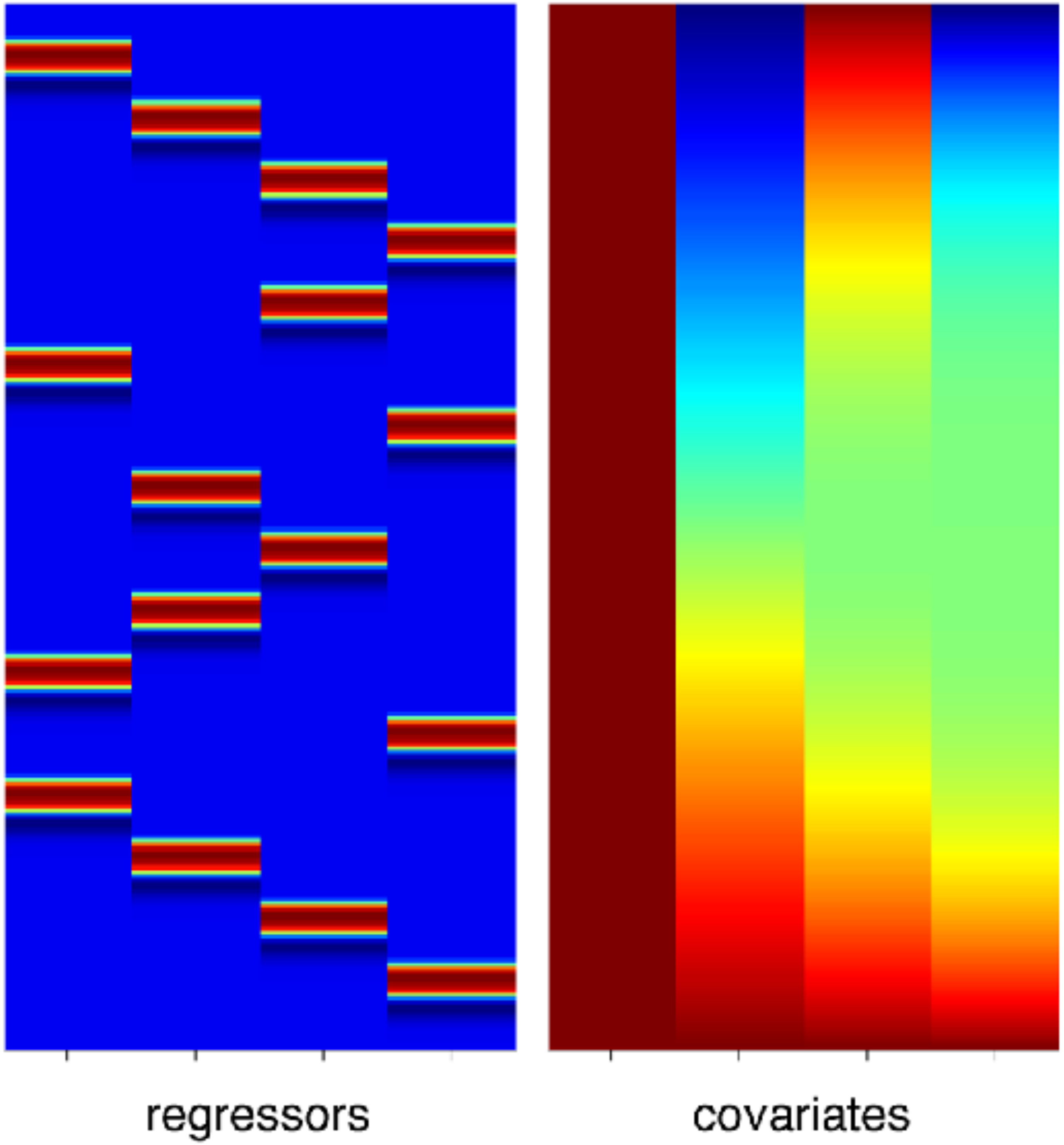


Trend removal

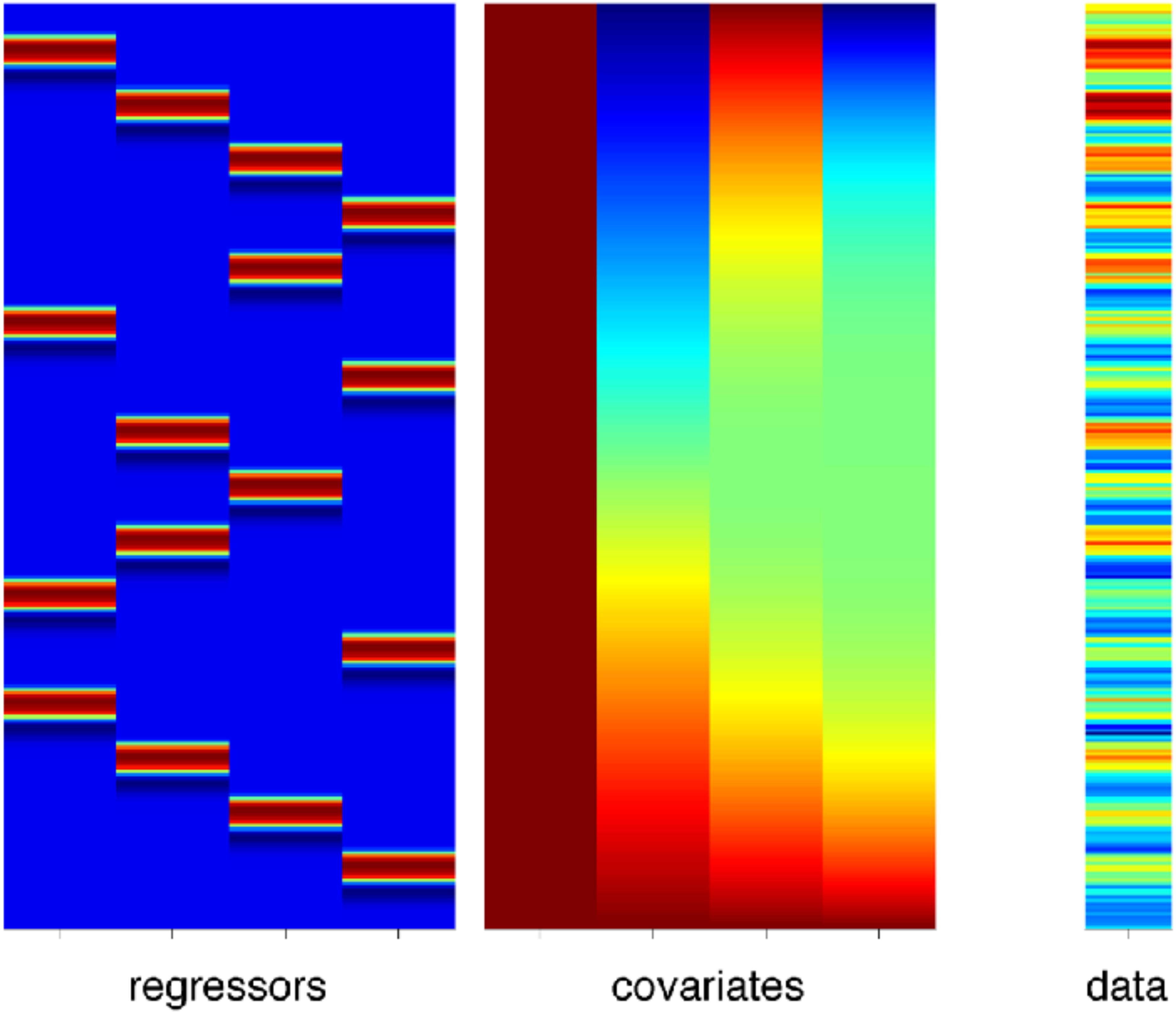
In SPM you won't see the trend covariates in the design matrix, except for the constant.

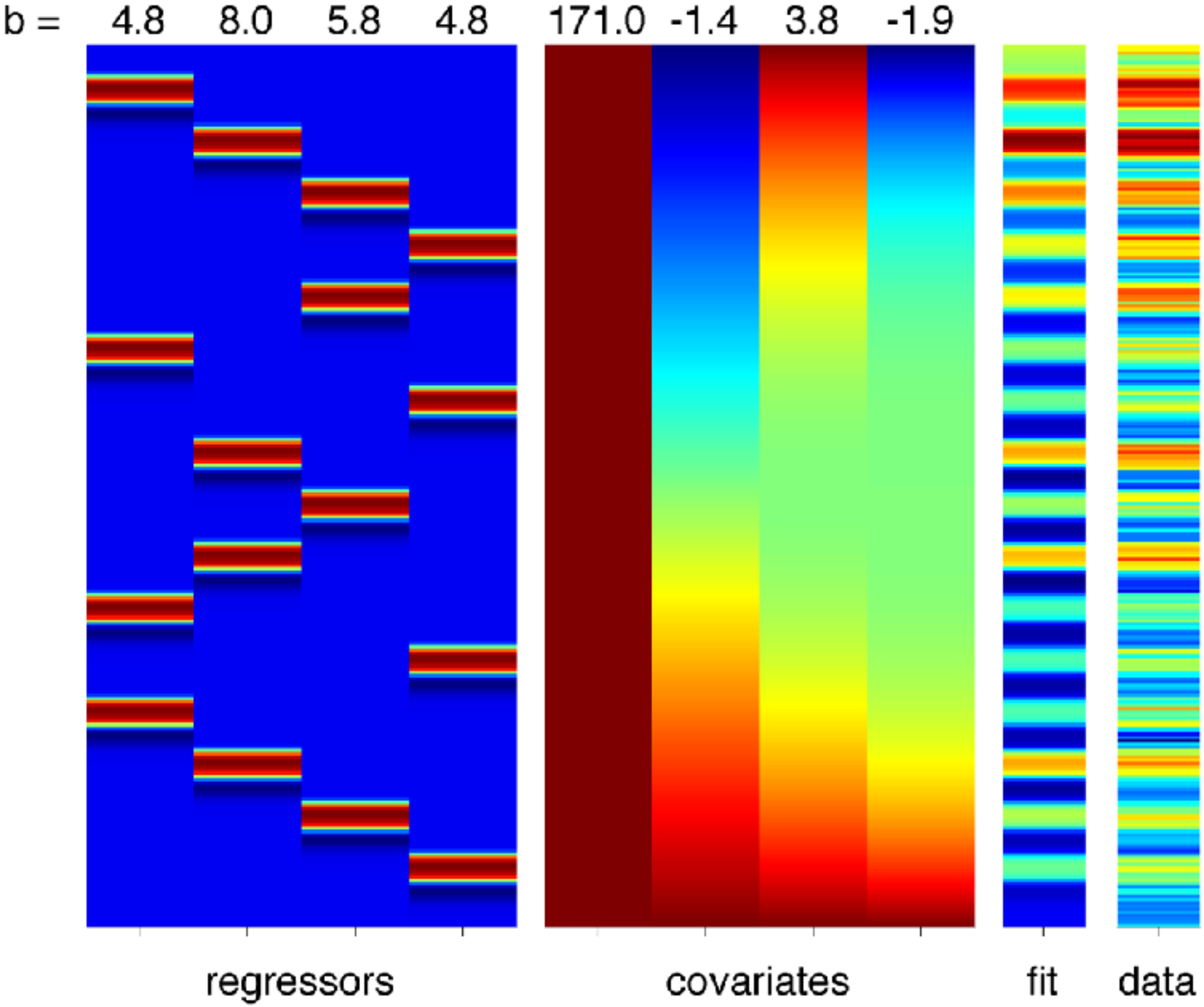
Instead you specify a cutoff for a high-pass filter (typically 128s), and the data and design matrix are both detrended by this.

(This is mathematically equivalent to including the trend regressors in the design matrix - you will get the same parameter estimates for your task regressors either way)



What weights should we use to obtain a fitted timecourse that is as close as possible to the data (ie, smallest squared deviation)?

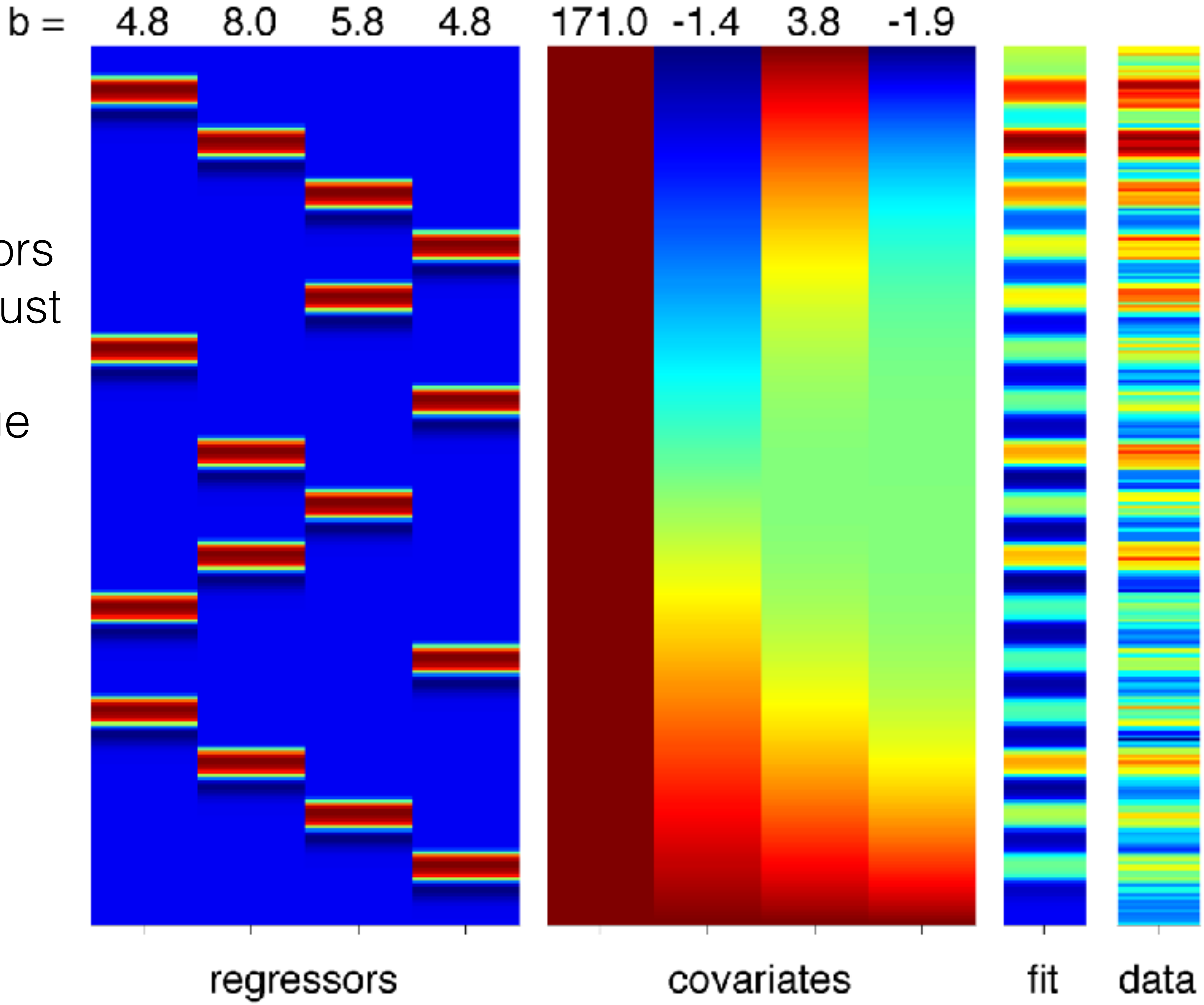



b =
4.8
8.0
5.8
4.8
171.0
-1.4
3.8
-1.9
regressors
covariates
fit
data

[-1 1 0 0 0 0 0 0] * b = 3.2, ie 8.0-4.8

[.25 .25 .25 .25 0 0 0 0] * b = 5.8, ie mean(b(1:4))

Contrast vectors are basically just a convenient way to average or subtract parameter estimates



Serial autocorrelation

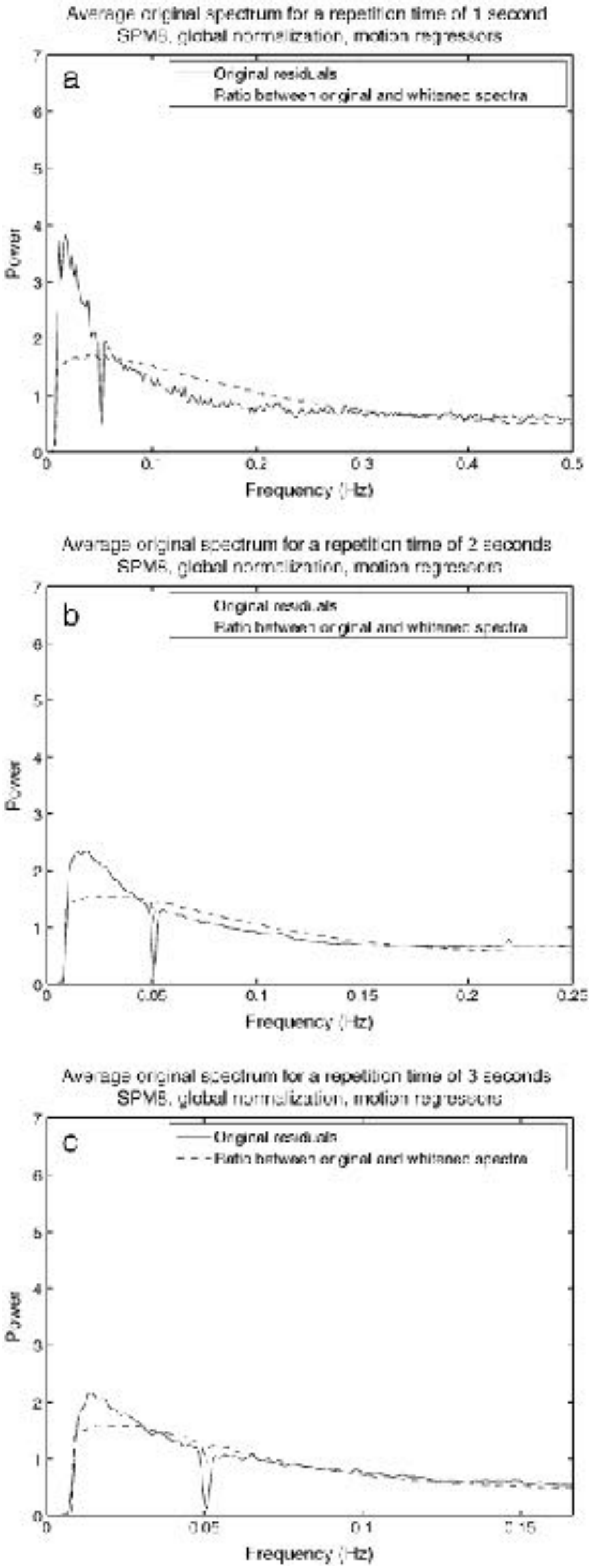
- fMRI residuals are not independent and identically distributed (*iid*)
- Why not? Breathing, heartbeat cycle, unmodelled neuronal activity
- This causes *serial autocorrelation*
    - if you correlate the residual timecourse with a time-shifted version of itself, rho > 0
    - The power spectrum of the residuals is higher at low frequencies (figure)
- Which invalidates the error term that is used for parametric stats inference (*T* tests, standard errors, *p* values) - we have fewer effective df than we thought!

Eklund et al., 2012, NeuroImage

TR=1s

TR=2s

TR=3s



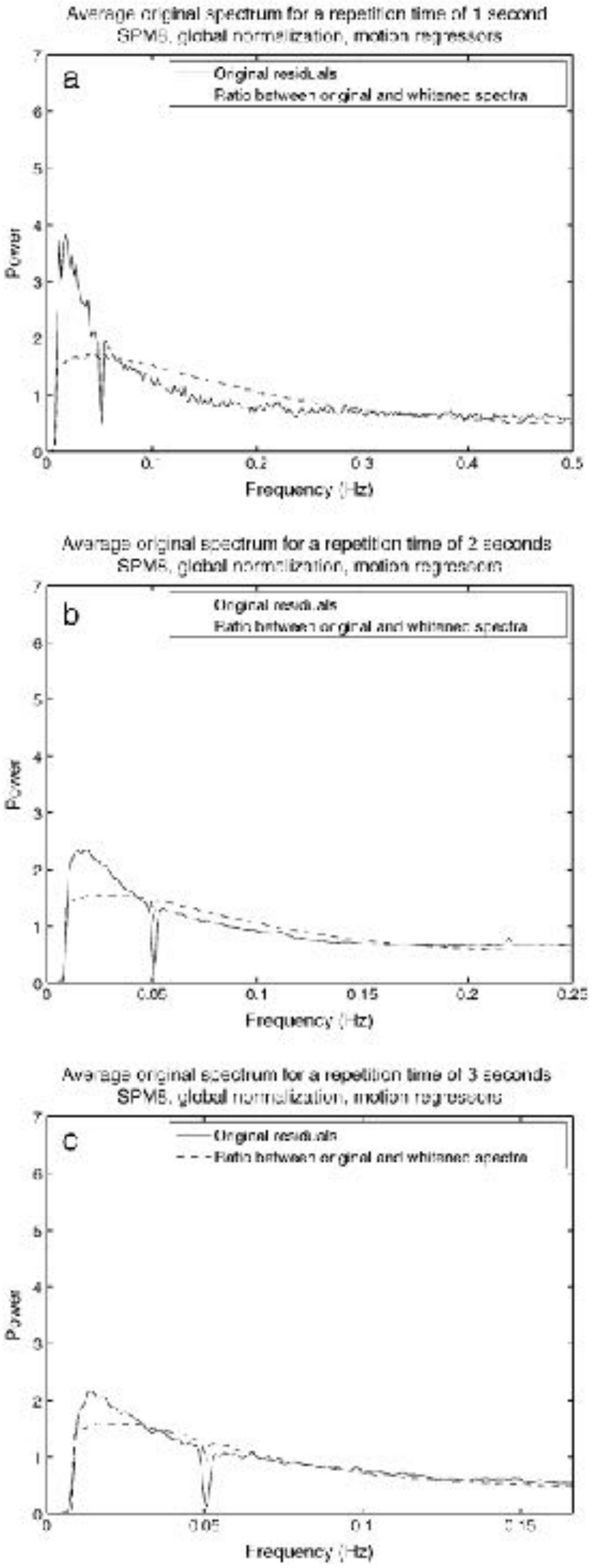
Serial autocorrelation

- fMRI residuals are not independent and identically distributed (*iid*)
- Why not? Breathing, heartbeat cycle, unmodelled neuronal activity
- This causes *serial autocorrelation*
    - if you correlate the residual timecourse with a time-shifted version of itself, rho > 0
    - The power spectrum of the residuals is higher at low frequencies (figure)
- Which invalidates the error term that is used for parametric stats inference (*T* tests, standard errors, *p* values) - we have fewer effective df than we thought!

TR=1s

TR=2s

TR=3s



Eklund et al., 2012, NeuroImage

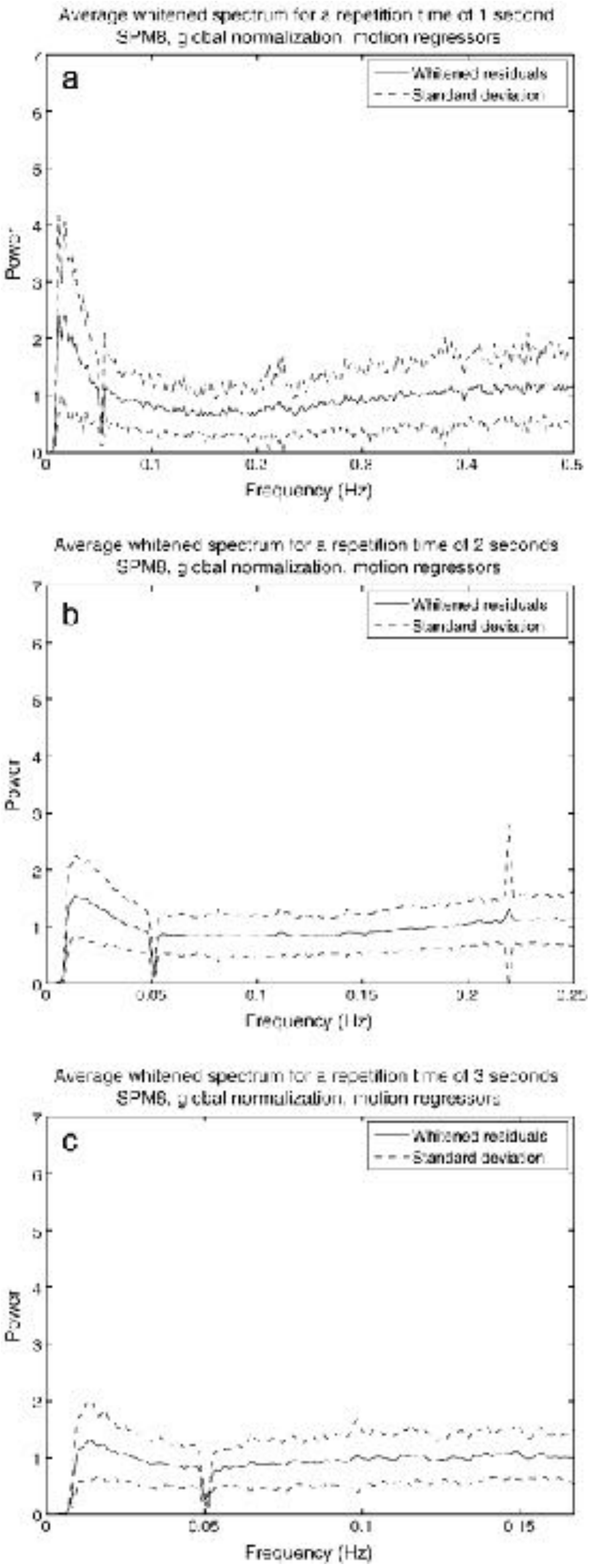
SPM AR(1) correction

- By default, SPM estimates (1st-order) autocorrelation and whitening data and design by this (nb, the same global adjustment for all voxels)
- This works, but not perfectly. Especially problematic with fast TR (figure)
- Why not? autocorrelation probably differs over the brain (e.g., veiny voxels), and probably extends for longer than 1 volume (so 1st-order isn't really enough)
- If you need single-participant *p* values you may want to use the new SPM FAST correction, or consider alternative (permutation test) approaches.

TR=1s

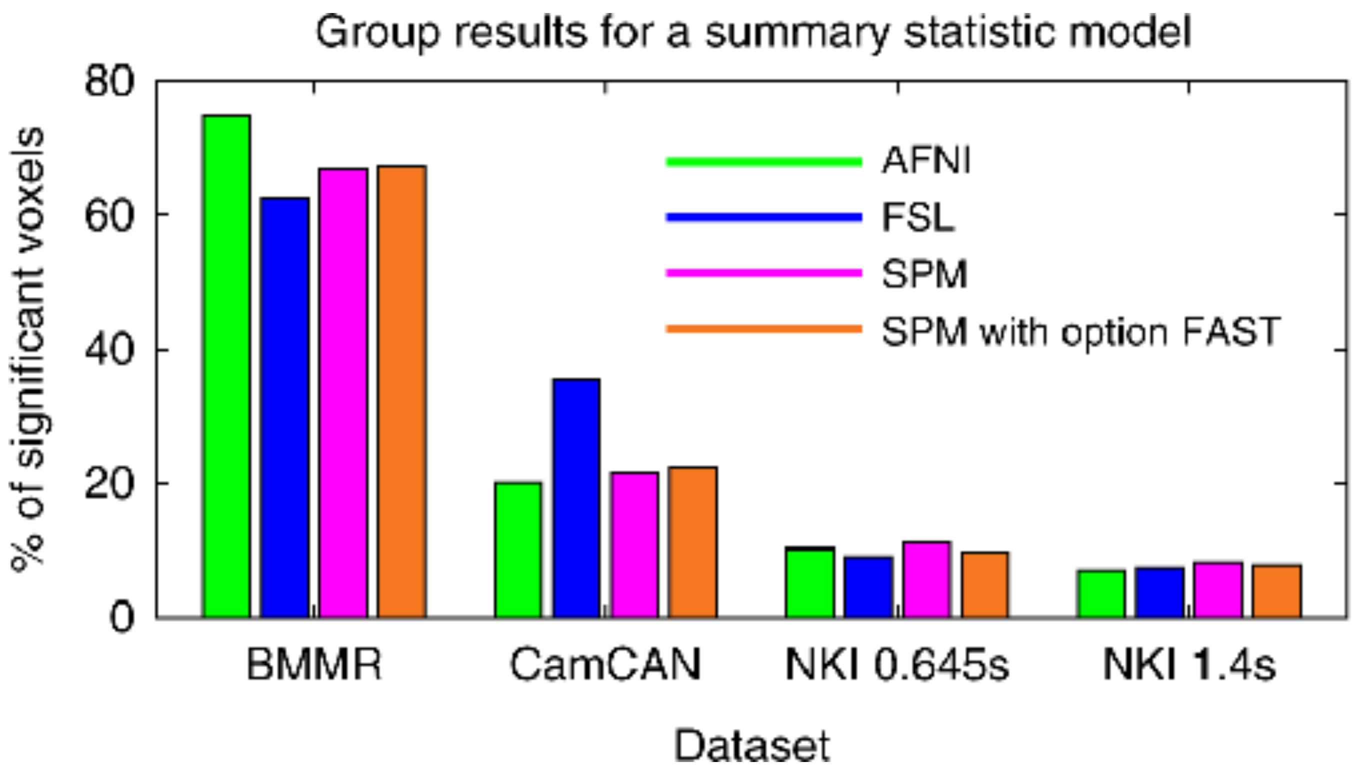
TR=2s

TR=3s



Eklund et al., 2012, NeuroImage

Does autocorrelation modelling matter for group results?

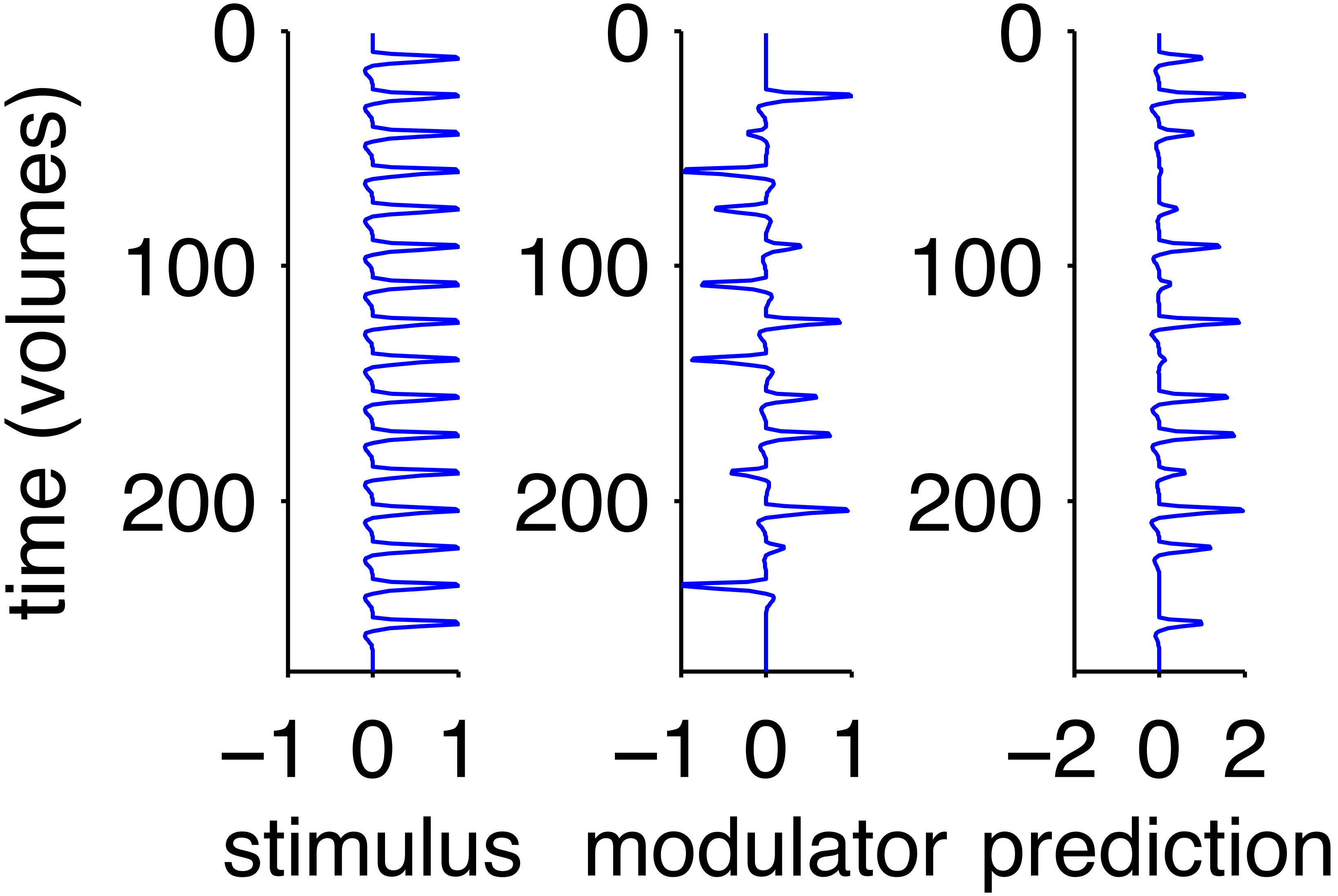


No.

For the (typical) group analysis case, AR modelling issues will potentially add (a very small amount of) variance to your single-participant estimates, but won’t bias your inferences (more on this next time)

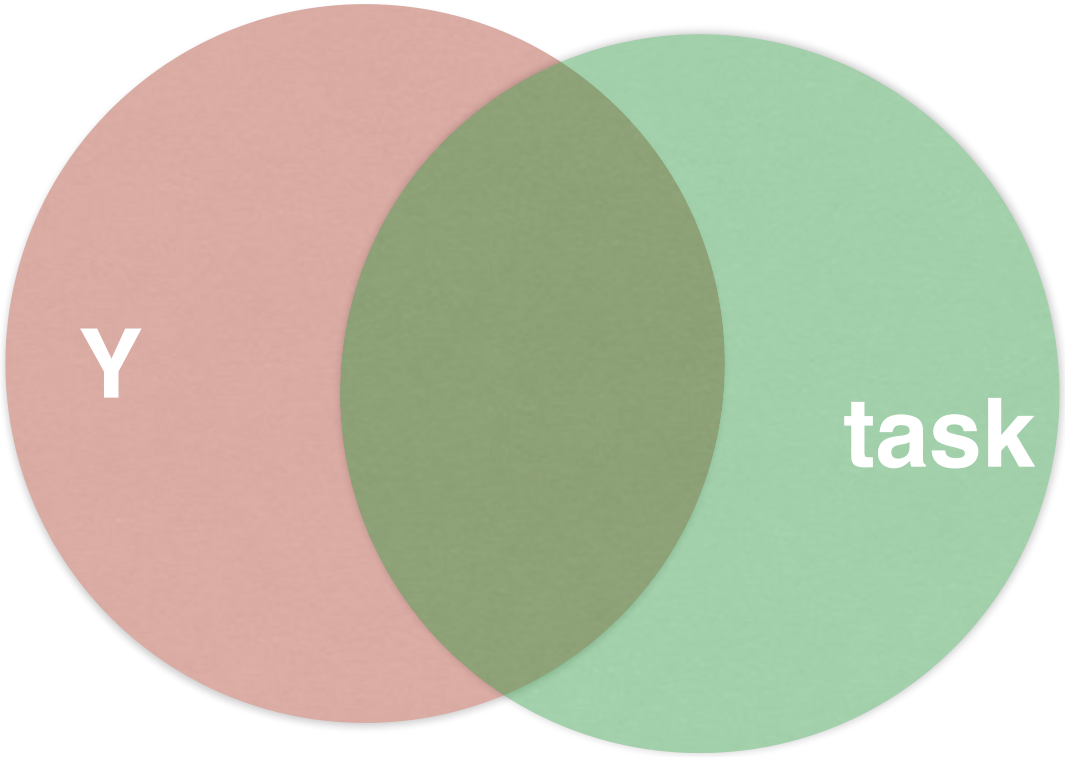
Olszowy et al., 2019, Nat Comm

Parametric modulators



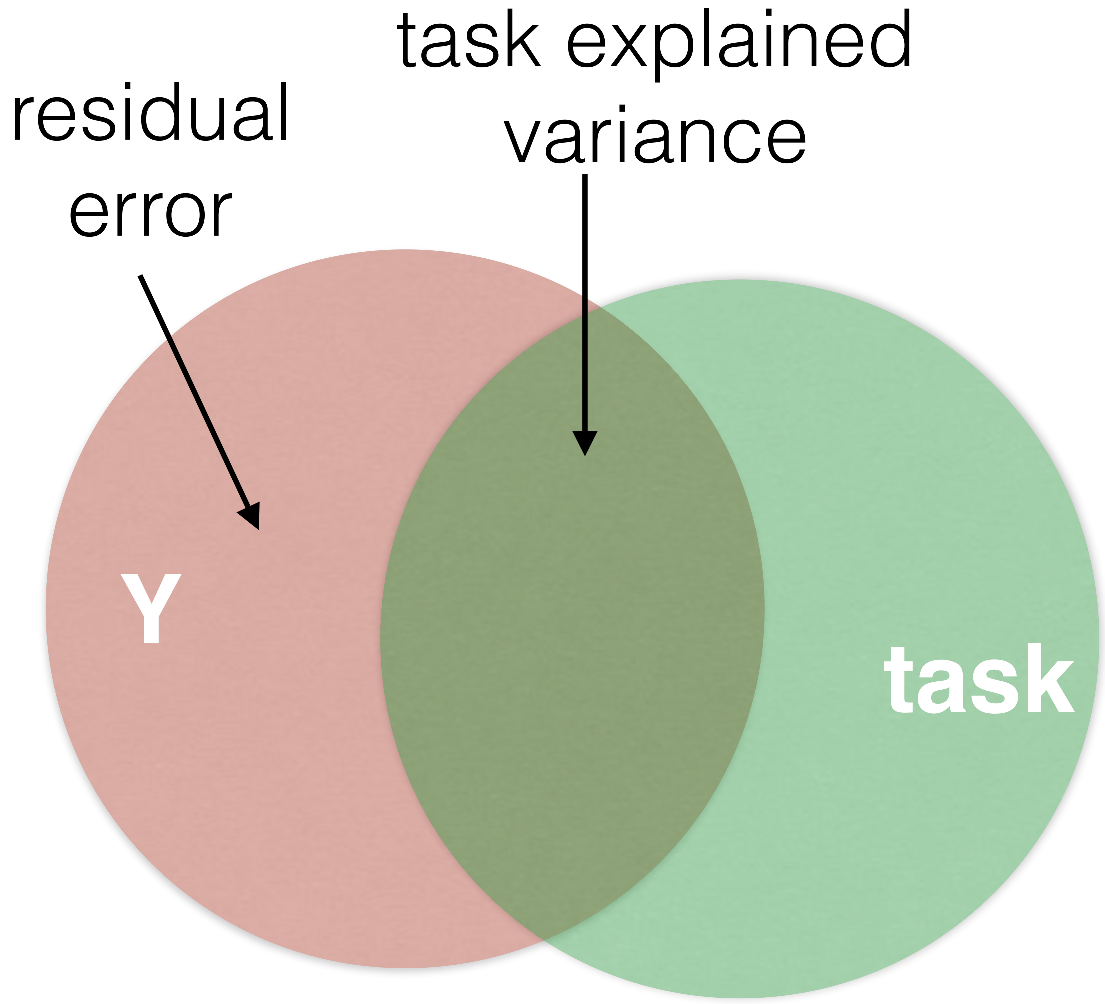
- Encode modulations of stimulus responses by continuous variables
- SPM solution is one regressor for the stimulus effect and another mean-centered regressor for each modulator on that response
- Why not just the modulator? Because we don't want to assume zero response when modulator=0
- Typical applications: Reinforcement learning (Dolan, O'Doherty), vision (Gallant), 'carry-over' fMRI (Aguirre), fancy grid cell stuff (Behrens)

Orthogonalisation



Mumford et al., 2015, *PLoS ONE*

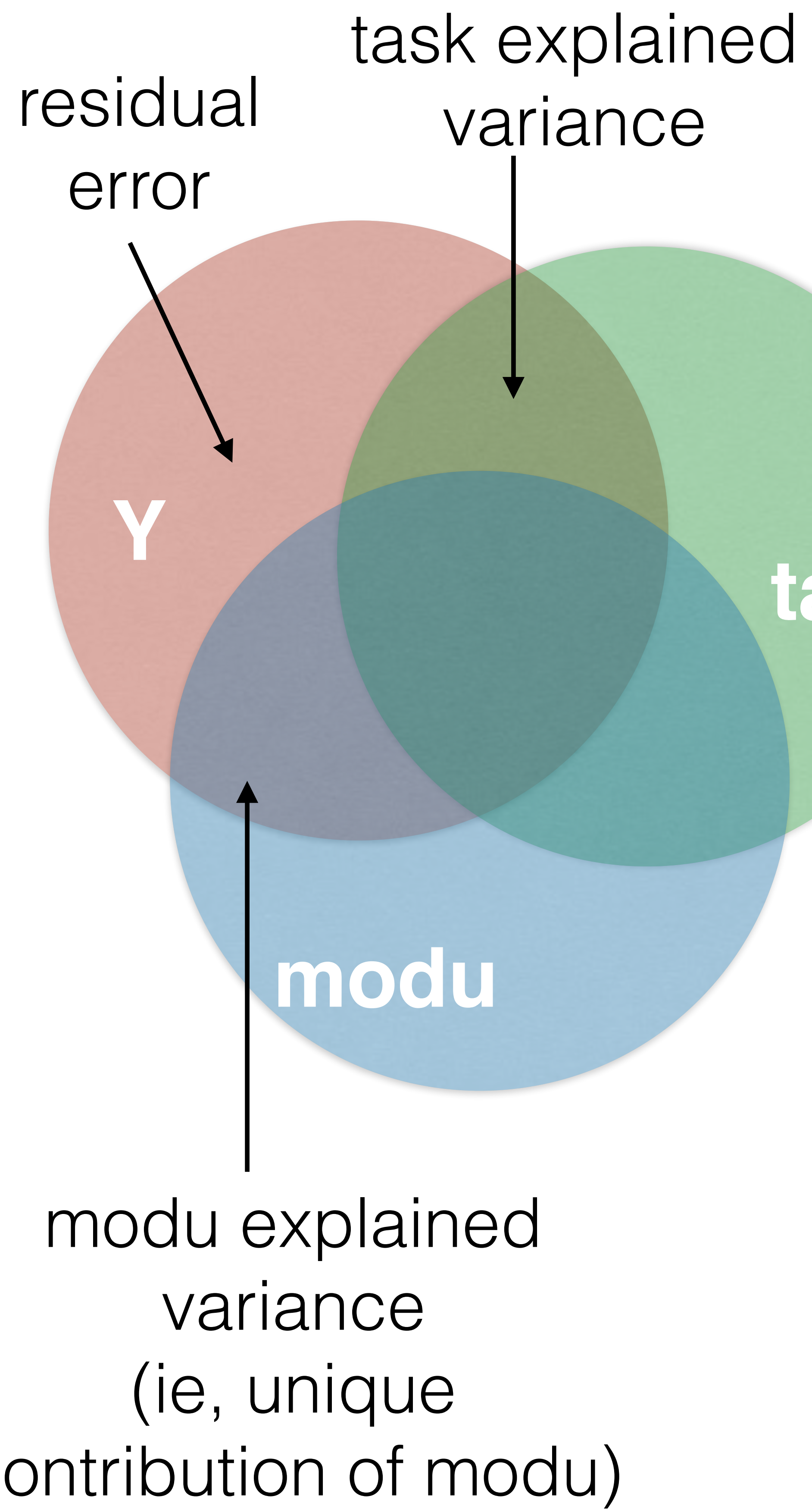
Orthogonalisation



Mumford et al., 2015, *PLoS ONE*

Orthogonalisation

- If we compare the models Y = task and Y = task + modu, the beta(task) will be larger when it is the only predictor in the model


residual error

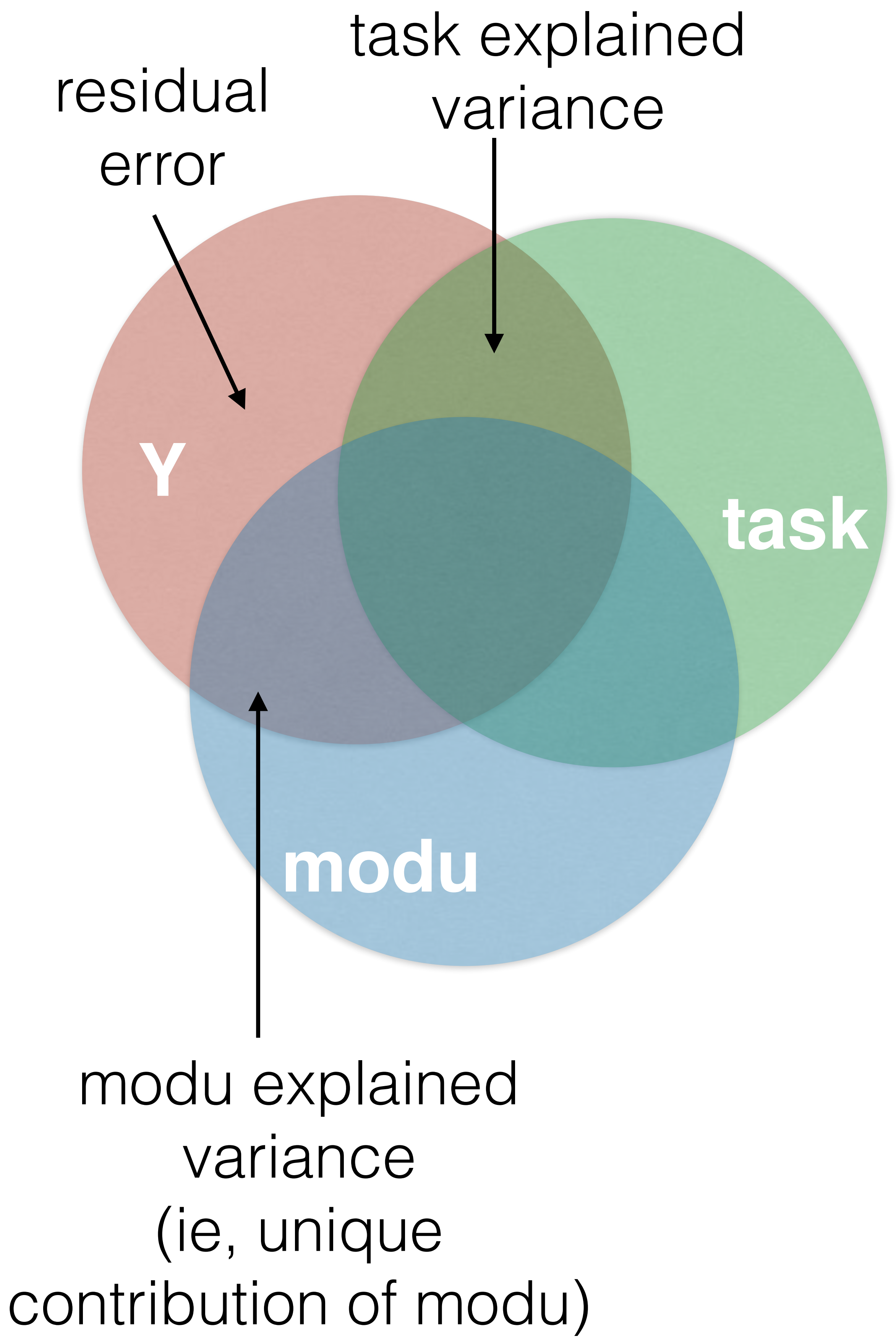
task explained variance

modu explained variance (ie, unique contribution of modu)

Mumford et al., 2015, *PLoS ONE*

Orthogonalisation

- If we compare the models Y = task and Y = task + modu, the beta(task) will be larger when it is the only predictor in the model
    - We can think of beta(task) as being ‘adjusted for modu’ when modu is in the model



residual error

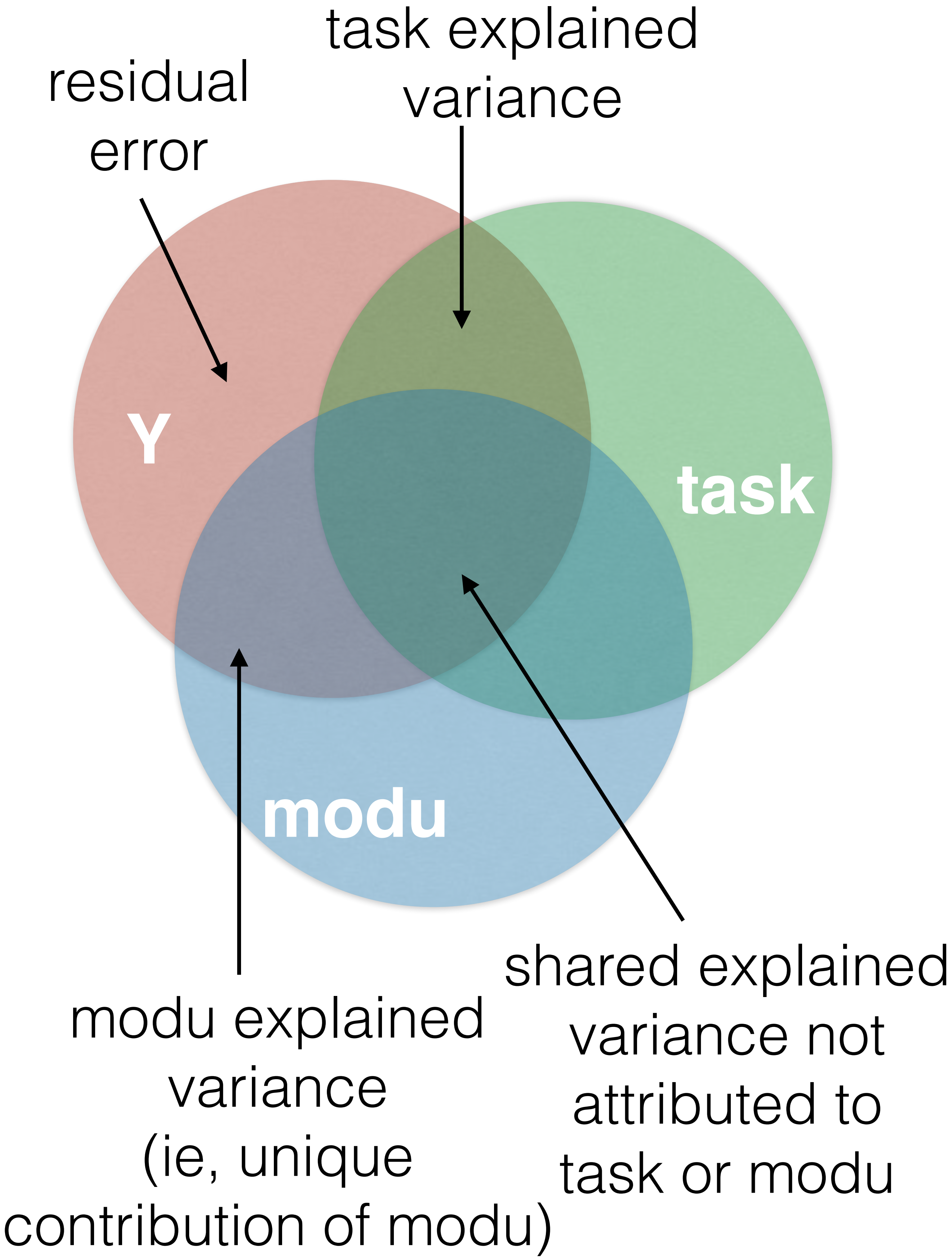
task explained variance

modu explained variance (ie, unique contribution of modu)

Mumford et al., 2015, *PLoS ONE*

Orthogonalisation

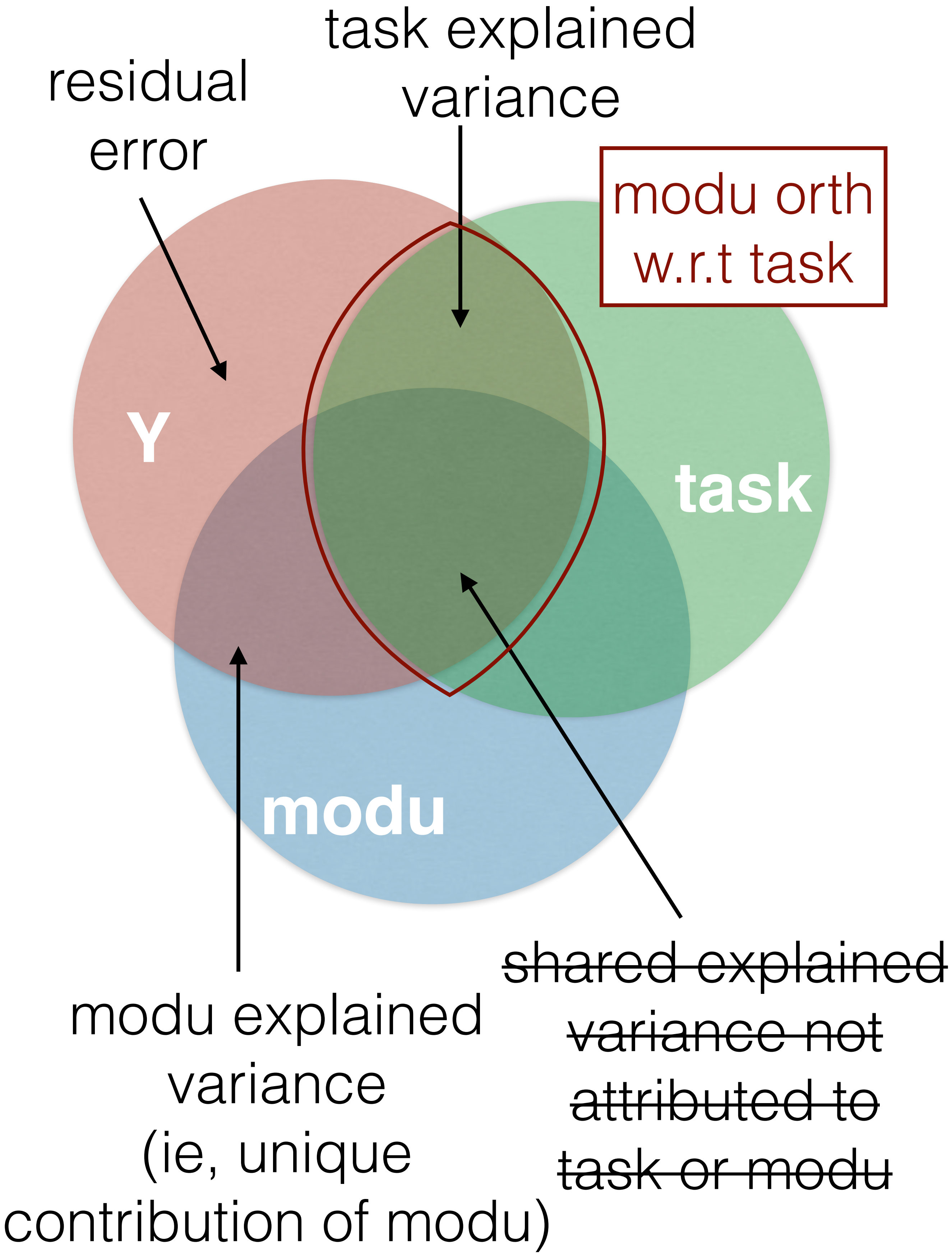
- If we compare the models Y = task and Y = task + modu, the beta(task) will be larger when it is the only predictor in the model
    - We can think of beta(task) as being 'adjusted for modu' when modu is in the model


task explained variance
residual error
Y
task
modu
modu explained variance (ie, unique contribution of modu)
shared explained variance not attributed to task or modu

Mumford et al., 2015, *PLoS ONE*

Orthogonalisation

- If we compare the models Y = task and Y = task + modu, the beta(task) will be larger when it is the only predictor in the model
    - We can think of beta(task) as being ‘adjusted for modu’ when modu is in the model
- What about if we orthogonalise modu with respect to task, ie, we regress out the contribution of task from modu?

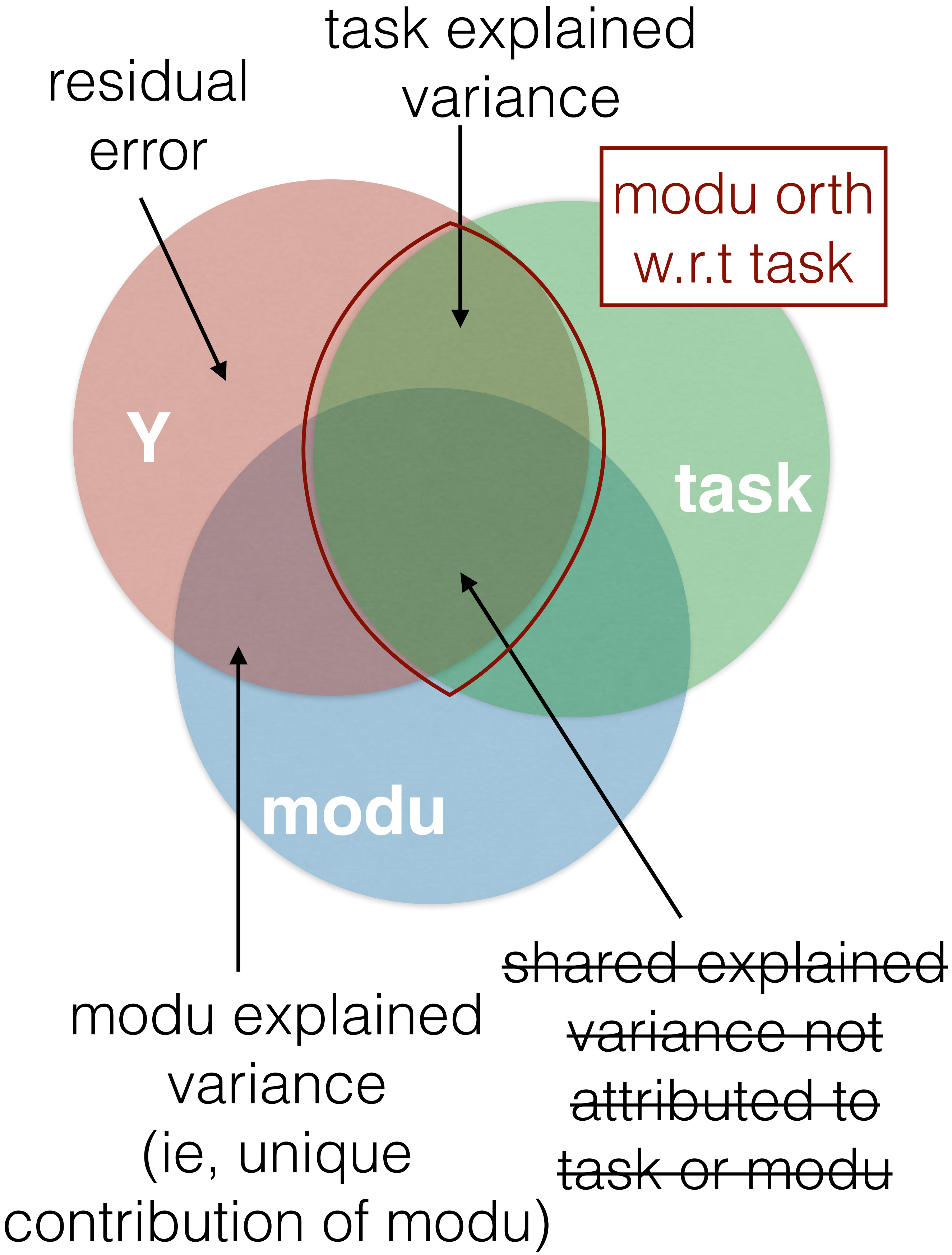


residual error
task explained variance
modu orth w.r.t task
Y
task
modu
~~shared explained variance not attributed to task or modu~~
modu explained variance (ie, unique contribution of modu)

Mumford et al., 2015, *PLoS ONE*

Orthogonalisation

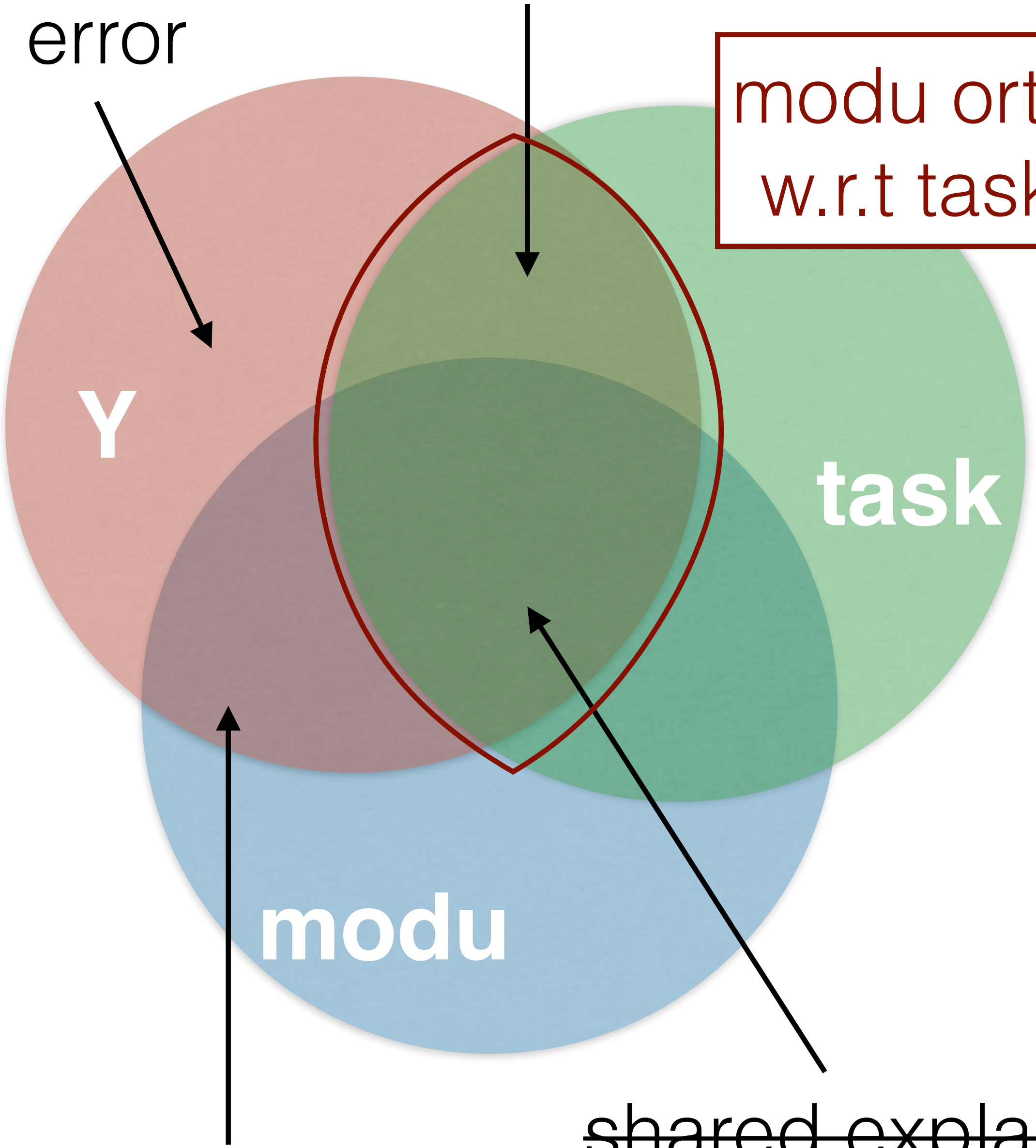
- If we compare the models Y = task and Y = task + modu, the beta(task) will be larger when it is the only predictor in the model
  - We can think of beta(task) as being ‘adjusted for modu’ when modu is in the model
- What about if we orthogonalise modu with respect to task, ie, we regress out the contribution of task from modu?
  - The modu predictor will change - *but modu’s parameter estimates will stay exactly the same* (counterintuitive but true!)
  - The task predictor stays the same - *but beta(task) will change*, because all the shared variance now goes to task


residual error
task explained variance
modu orth w.r.t task
Y
task
modu
modu explained variance (ie, unique contribution of modu)
~~shared explained variance not attributed to task or modu~~

Mumford et al., 2015, *PLoS ONE*

Orthogonalisation

- If we compare the models Y = task and Y = task + modu, the beta(task) will be larger when it is the only predictor in the model
  - We can think of beta(task) as being ‘adjusted for modu’ when modu is in the model
- What about if we orthogonalise modu with respect to task, ie, we regress out the contribution of task from modu?
  - The modu predictor will change - *but modu’s parameter estimates will stay exactly the same* (counterintuitive but true!)
  - The task predictor stays the same - *but beta(task) will change*, because all the shared variance now goes to task
- In effect, beta(task) from the model Y = task + modu(orth(task)) will be similar to the model Y = task, but with less residual error
  - Same beta but less error = ‘better’ stats



residual error

task explained variance

modu orth w.r.t task

modu explained variance (ie, unique contribution of modu)

~~shared explained variance not attributed to task or modu~~

Mumford et al., 2015, *PLoS ONE*

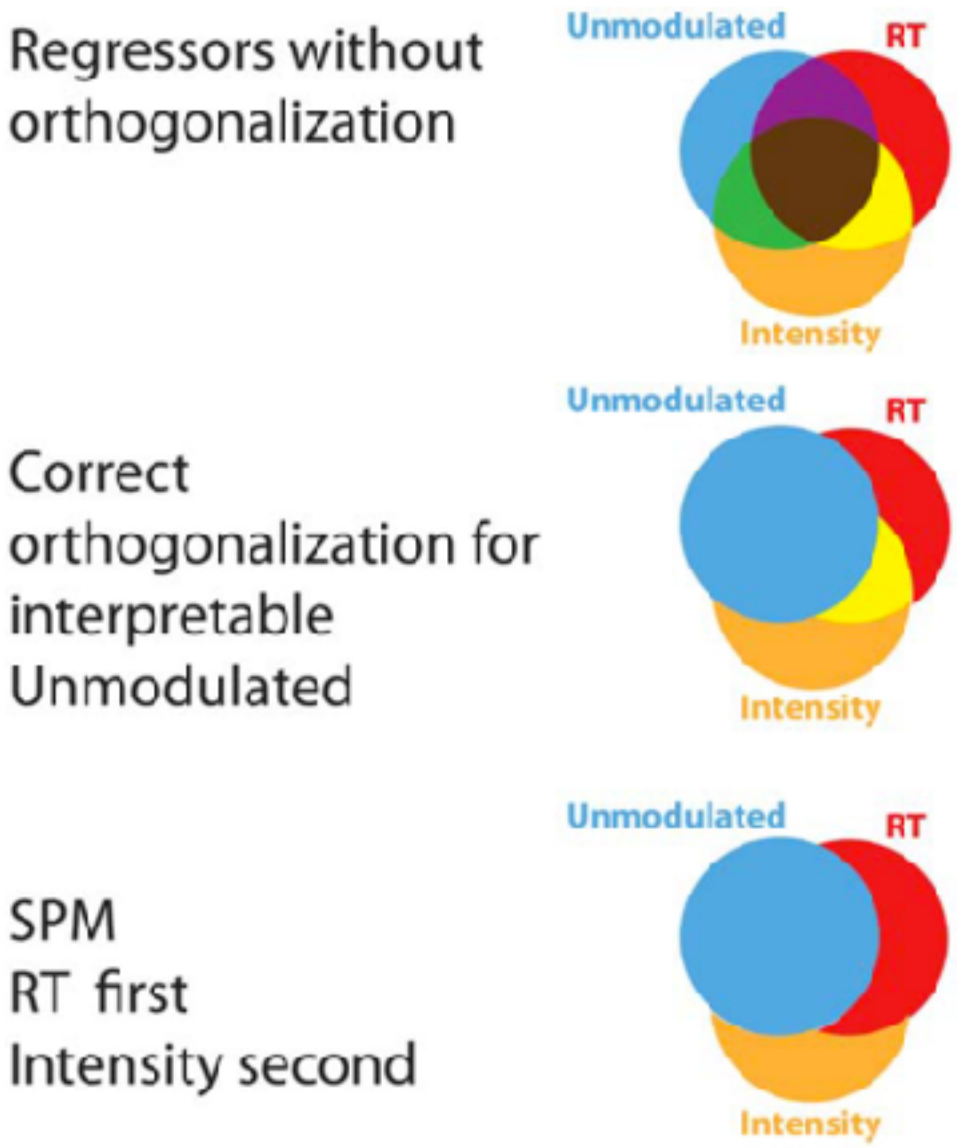
SPM orthogonalisation quirks

- When you have multiple modulators, SPM by default orthogonalises them *sequentially* - so the order in which you enter the modulators in the design matrix is significant!

- You may not want this 'feature' since it really complicates interpretation

- If you want to change this you will have to hand code the orthogonalised modulators and add them (disabling SPM's own orth).

- Or fit the model twice, taking the second modulator's beta from each



RT wrt Unmod
Intensity wrt Unmod

RT wrt Unmod
Intensity wrt Unmod and RT

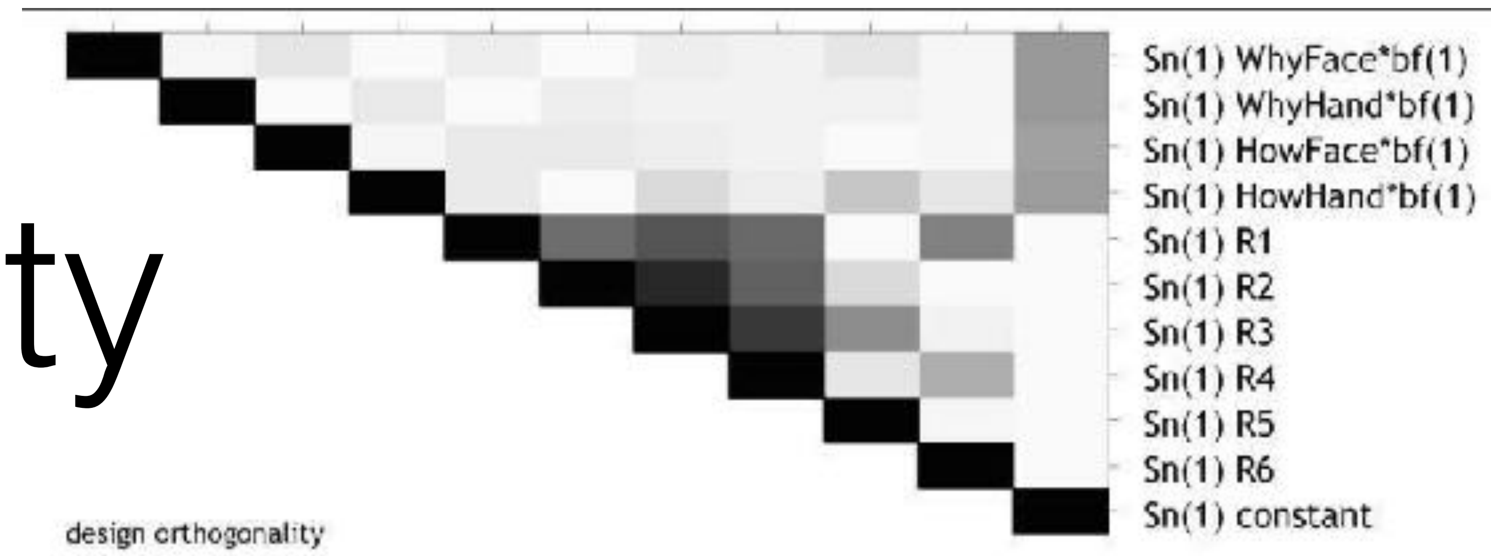
Mumford et al., 2015, *PLoS ONE*

Designing fMRI experiments

Given what we now know about the assumed HRF shape and the noise model, what kind of design is most efficient for detecting a hypothesised effect?

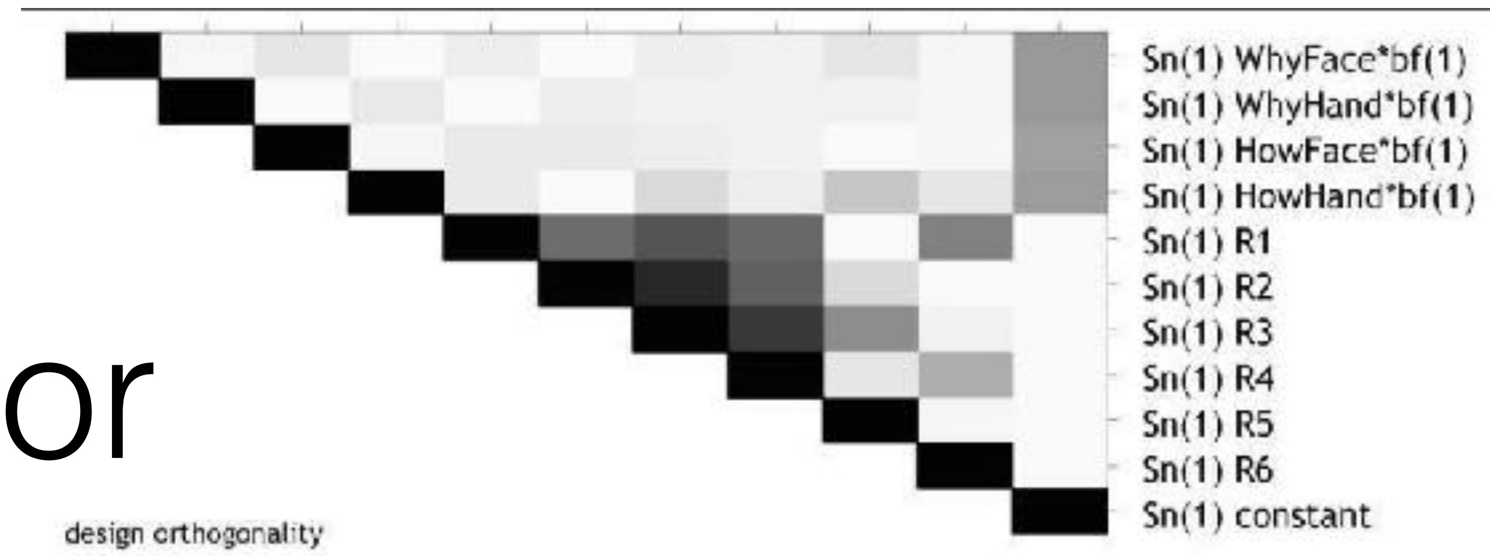
Experimental design has a *huge* effect on detection power in fMRI — this can make or break your study

Collinearity



- Dependencies between convolved regressors increases the variance of parameter estimates
  - NB, does *not* bias the fit - but can make it almost impossible to detect effects (e.g. the single-participant betas that go into group analysis will be highly variable)
- Big problem in fMRI, since convolution with HRF introduces dependencies between neighbouring events (e.g., encoding and recall phase in memory experiment)
- SPM outputs collinearity estimates (see above - basically predictor correlation matrix). Useful for finding pairs of dependent conditions. But a bit late to find this out at model fit stage!

Variance inflation factor

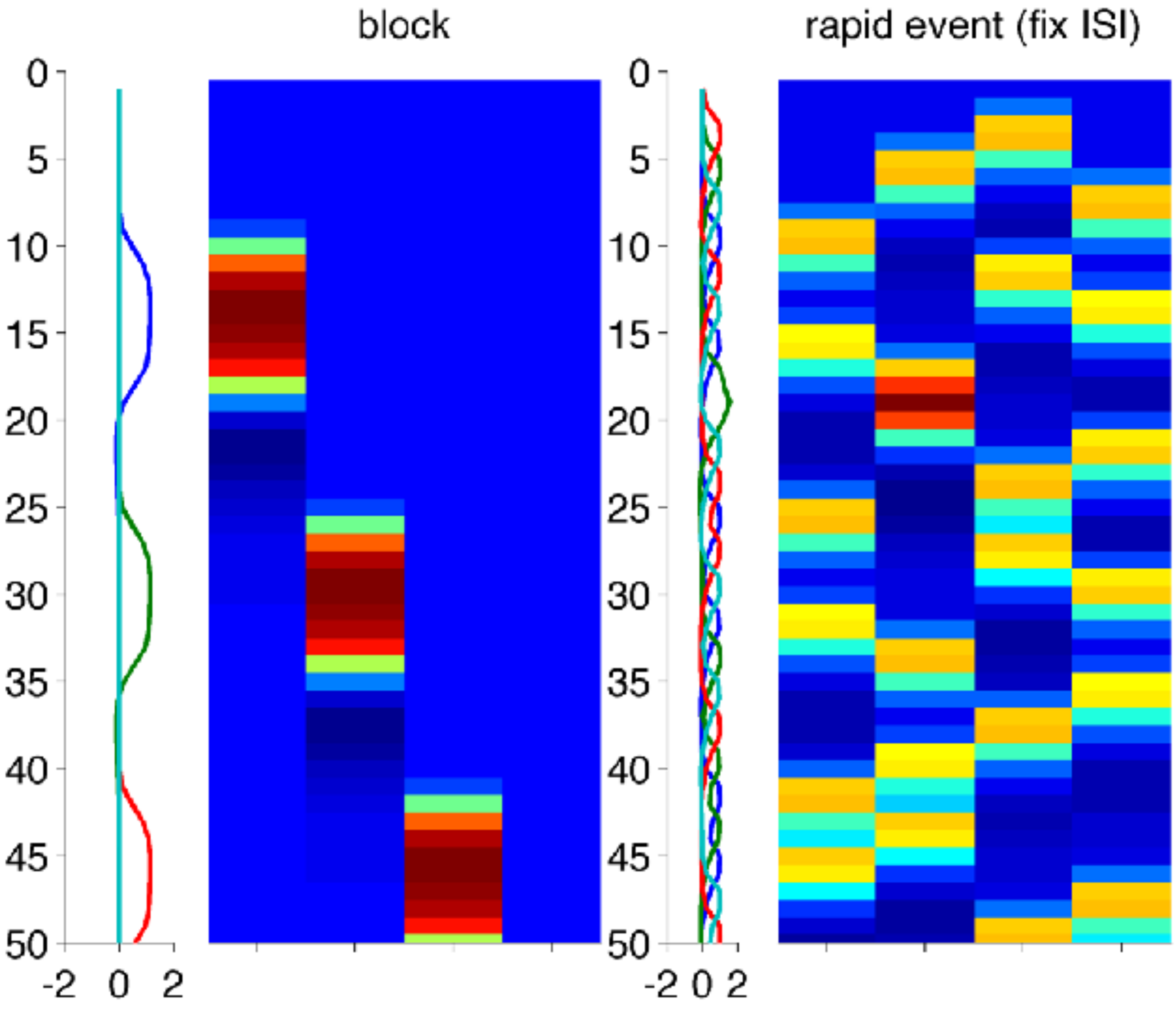


- Collinearity can also arise over sets of regressors - consider using variance inflation factor (VIF) to test for this *at experimental design stage*
- VIF = 1 / (1-R2) where R2 comes from using all regressors but one to predict the final regressor
- Typical values:
    - VIF=1 for completely orthogonal designs (zero correlation between prediction and left-out regressor)
    - VIF=Inf for rank deficient designs (perfect correlation between prediction and left-out regressor)
    - By convention, VIF>5 indicates a problem (but lower is better)

Design efficiency

- In Matlab code **designeff = 1 / trace(c * inv(X’*X) * c’)**, Where
  - *c* is a matrix of contrasts (here with contrasts stacked in rows)
  - *X* is the convolved and filtered design matrix
  - If you’ve seen the formula for calculating standard errors in a GLM, this may seem familiar… It’s **sterr = sqrt(diag(c * inv(X’*X) * c’) * mrss)**, where mrss is the mean residual sum of squares (one value per voxel)
- A design with higher design efficiency should produce contrast estimates with less variance
  - Importantly, the contrast estimate won’t be affected, so optimising your efficiency for a particular contrast doesn’t bias your estimates (although that contrast will have smaller standard errors and hence smaller p values)
- Caveat: Efficiency is not in meaningful units (it scales with design matrix, and it’s monotonic but nonlinear with respect to expected standard errors)
  - So unlike VIF there are no fixed rules for what is a ‘good’ efficiency
  - It’s also not the case that a design with 2x efficiency has exactly 2x power - but higher efficiency is always associated with increases in power

Typical fMRI designs



Very efficient

Very inefficient

Rules of thumb for fMRI design

1. Randomise trial order for each run to minimise collinearity
2. Cluster trials (pseudorandom event-related design or just block) to keep signal in low frequency band (the HRF convolution basically *low-pass filters* the regressor)
3. Don’t put conditions you want to compare too far apart (>60s) (the de-trend *high-pass filters* the regressor)
4. Keep the number of conditions as small as possible to make the above easier (and to enable shorter runs)
5. For differential effects (ie, what you usually care about), fixed ISI works best
6. For much more on this, see CBU imaging wiki entry on design efficiency or the associated Henson paper

Useful references

- Rik's design efficiency wiki: http://imaging.mrc-cbu.cam.ac.uk/imaging/DesignEfficiency

- Jeanette Mumford's brain stats blog: mumfordbrainstats.tumblr.com (see also facebook group)

- The SPM mailing list: https://www.jiscmail.ac.uk/lists/SPM.html (vast searchable archive)

- Kendrick Kay's course on Statistics and Data Analysis in MATLAB: http://kendrickkay.net/psych5007/ (if you want to roll your own GLM)

Workshop time

workshop laptops (using X2Go):

In matlab, type

```
practical_efficiency
```

Use cursor to place 20 trials for each condition into the design matrix figure. Press return/enter when finished to obtain a convolved and filtered design matrix, with efficiency and VIF estimates.

Repeat as needed

personal laptops (requires Matlab, SPM12):

download workshop materials from

http://imaging.mrc-cbu.cam.ac.uk/methods/IntroductionNeuroimagingLectures - unzip to directory

In matlab: ensure SPM12 is on your path (`addpath spmdir`), move to said directory (`cd unzippeddir`)

follow above steps for workshop laptops