TUTORIAL
Model-based fMRI

Birte Toussaint

Translational Neuromodeling Unit

Universität Zürich[UZH]

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Methods & Models for fMRI Analysis
3 December 2019

Prerequisites

- Previously preprocessed data files
  - or: run ***teach_prepro_subject('path/to/Sub01', 1)*** on raw functional scans
- Behavioural data
  - BehaviorSummary files in /Sub01/behav/
- Tapas toolbox
  - added to MATLAB path by running ***tapas_init***

Behavioural parameters

- BehaviorSummary files contain:
  - tLeftStim & tRightStim: presentation time of left/ right arrow
  - tLeftPress & tRightPress → time of left/ right button presses

- Generate input and response vectors
  - run ***teach_analyse_behaviour_hgf('path/to/Sub01')***

    → BehavHGF files
  - **inputs u:** experimental stimuli (left arrow = 0; right arrow = 1)
  - **responses y:** vector of Sub01's button presses (left = 0; right = 1)

Steps for model-based fMRI

1. Choose a model
2. Find best-fitting parameters of model to behavioral data
3. Generate model-based time series
4. Convolve time series with HRF
5. Regress against fMRI data

1. Choose a model

Level 3: Phasic volatility

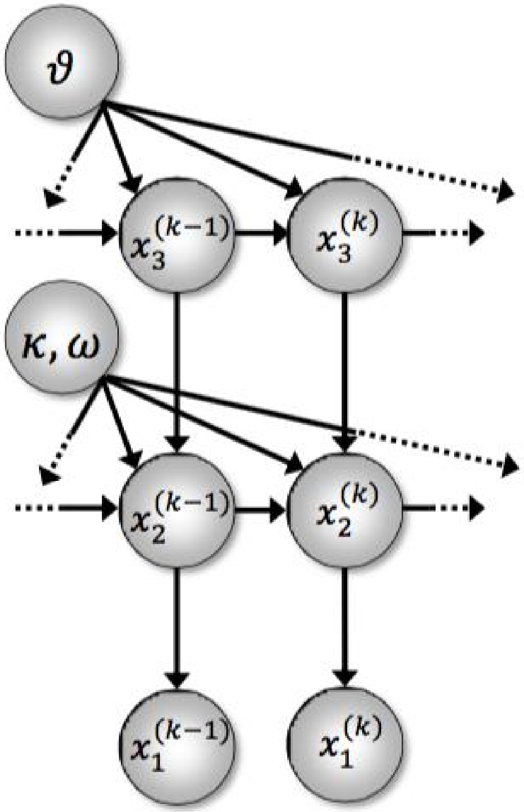
$$p\left(x_3^{(k)}\right) \sim \mathcal{N}(x_3^{(k-1)}, \vartheta)$$

Level 2: Tendency towards category 1

$$p\left(x_2^{(k)}\right) \sim \mathcal{N}(x_2^{(k-1)}, e^{(\kappa x_3^{(k-1)} + \omega)})$$

Level 1: Stimulus category

$$p(x_1 = 1) = \frac{1}{1 + e^{-x_2}}$$



Mathys et al., *Front Hum Neurosci*, 2011

Specify model priors

- Decide which parameter(s) to estimate
  - example: κ_2 (coupling strength between levels 1 and 2)

- Adjust HGF configuration file
  - Open /tapas/HGF/tapas_hgf_binary_config.m

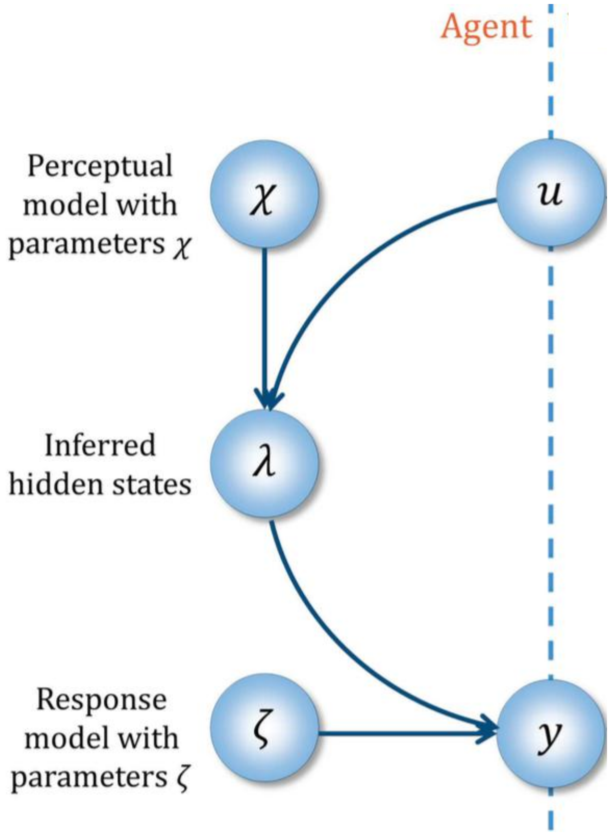
```
% Kappas
% Format: row vector of length n_levels-1.
% Fixing log(kappa1) to log(1) leads to the original HGF model.
% Higher log(kappas) should be fixed (preferably to log(1)) if the
% observation model does not use mu_i+1 (kappa then determines the
% scaling of x_i+1).
c.logkamu = [log(1), log(1)];
c.logkasa = [     0,      4^2]; % estimate kappa2

% Omegas
% Format: row vector of length n_levels.
% Undefined (therefore NaN) at the first level.
c.ommu = [NaN,  -3,  -6];
c.omsa = [NaN, 0, 0];
```

set prior variance of $\kappa_2 > 0$ to free the parameter

set prior variance of ω_1 and ω_2 to 0: we don't want to estimate them

Specify model priors

- You can just use the following file, which is already adjusted:
  - ***teach_tapas_hgf_binary_config.m***
- Load behavioural parameters:
  - BehavHGF02.mat
- Use tapas_fitModel to estimate parameters:
  - est = tapas_fitModel(responses, inputs, <prc_model>, <obs_model>, <opt_algo>);



Specify model priors

- Find Bayes optimal parameters:

leave empty: optimal parameter values are independent of responses

bayes_opt = tapas_fitModel([], u, 'teach_tapas_hgf_binary_config', 'tapas_bayes_optimal_binary_config', 'tapas_quasinewton_optim_config');

- Set the prior mean of κ_2 to the Bayes optimal value
  - Adjust tapas_hgf_binary_config.m

```
% Kappas
% Format: row vector of length n_levels-1.
% Fixing log(kappa1) to log(1) leads to the original HGF model.
% Higher log(kappas) should be fixed (preferably to log(1)) if the
% observation model does not use mu_i+1 (kappa then determines the
% scaling of x_i+1).
c.logkamu = [log(1), 0.090349]; % Bayes optimal prior
c.logkasa = [     0,      4^2];
```

set prior mean of κ_2 to bayes_opt.p_prc.ka(2)

  - or use ***teach_tapas_hgf_binary_config_adjusted.m***

2. Find best-fitting parameters of model to behavioral data

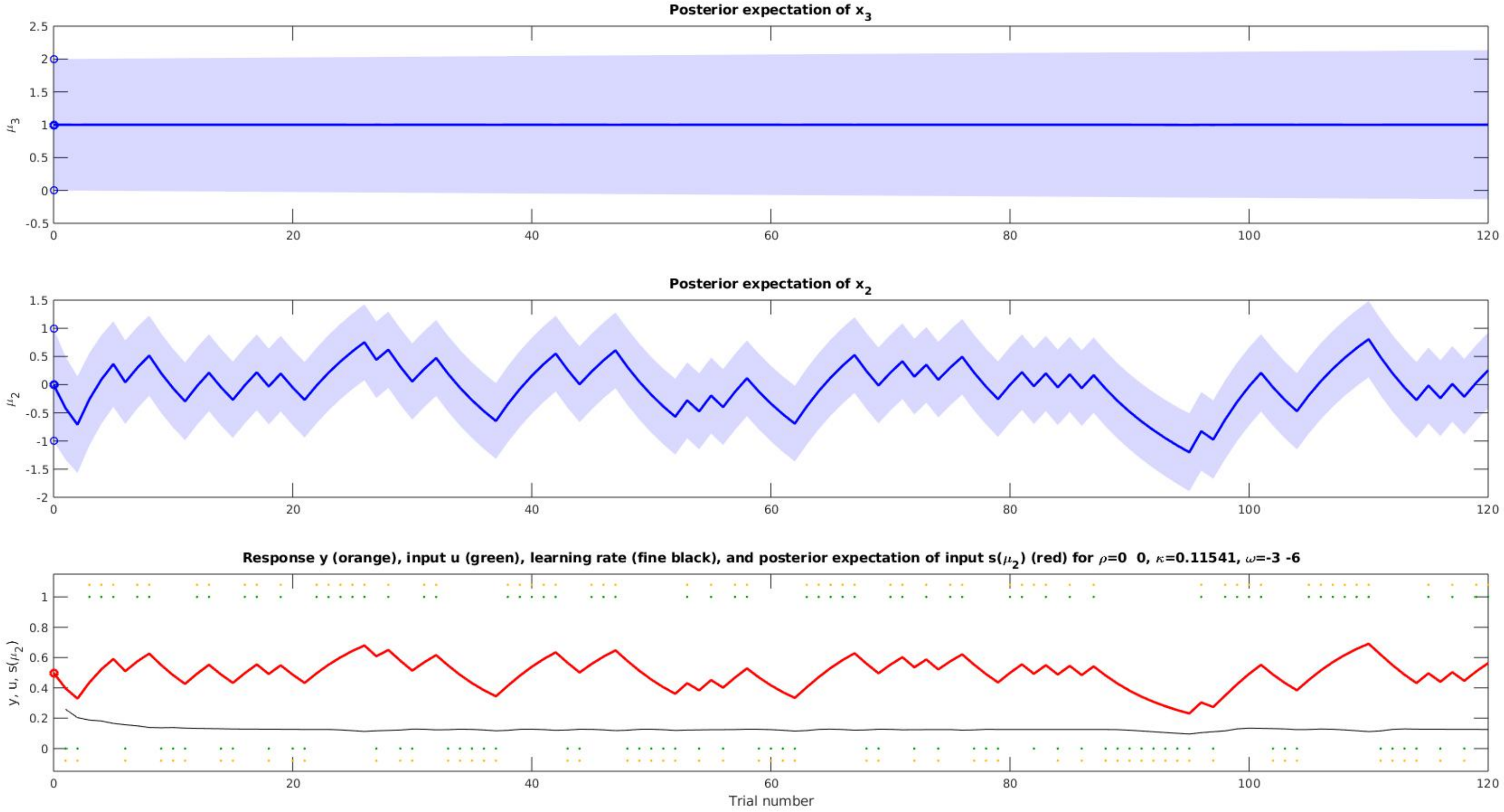
- Fit an HGF to the data:

  est = tapas_fitModel(y, u, 'teach_tapas_hgf_binary_config_adjusted', 'tapas_unitsq_sgm_config', 'tapas_quasinewton_optim_config');

```
Results:

Parameter estimates for the perceptual model:
    mu_0: [NaN 0 1]
    sa_0: [NaN 1 1]
     rho: [NaN 0 0]
      ka: [1 0.1154]
      om: [NaN -3 -6]


Parameter estimates for the observation model:
    ze: 0.6961

Model quality:
    LME (more is better): -96.8247
    AIC (less is better): 176.4899
    BIC (less is better): 182.0649

    AIC and BIC are approximations to -2*LME = 193.6495.
```

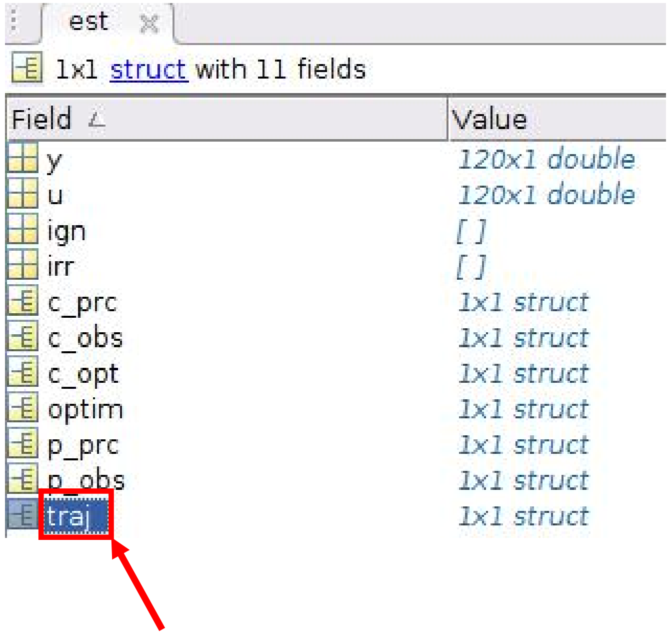
2. Find best-fitting parameters of model to behavioral data

- Visualise the inferred belief trajectories:
  tapas_hgf_binary_plotTraj(est)

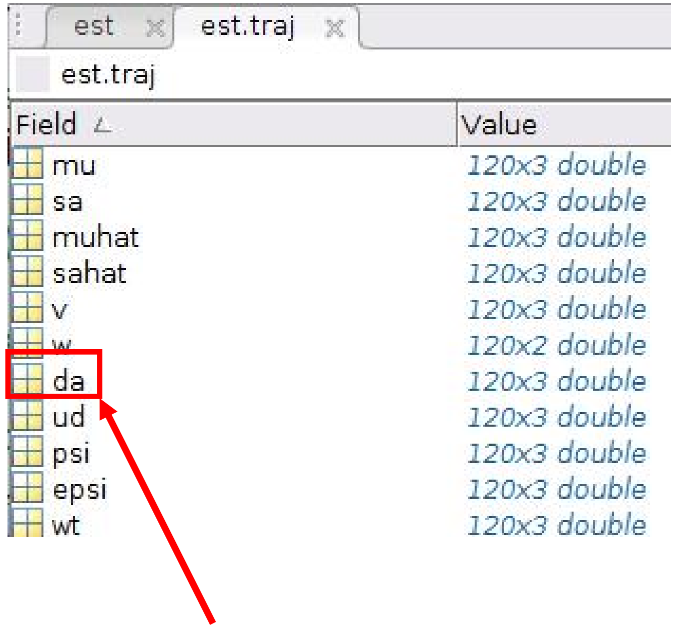


3. Generate model-based time series

- **Model outputs:**



trajectories of the environmental states tracked by the perceptual model

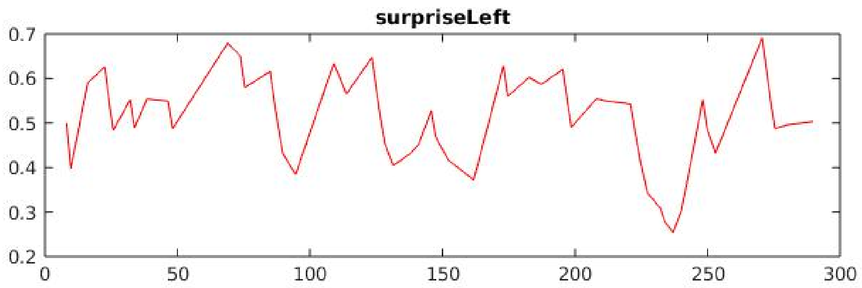


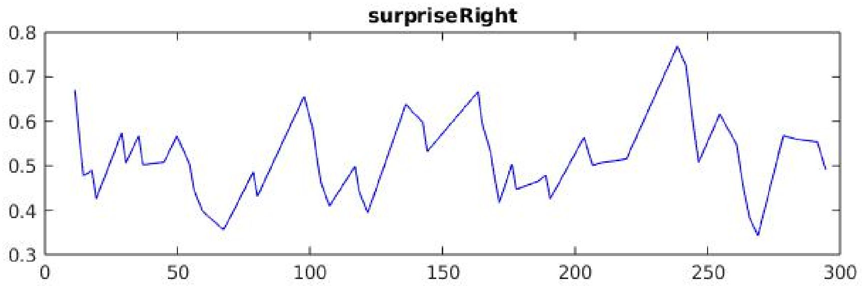
δ: prediction errors

3. Generate model-based time series

- E.g. time series representing surprise about the arrow direction
- In the Bayesian framework ***surprise*** is given by the unsigned PE

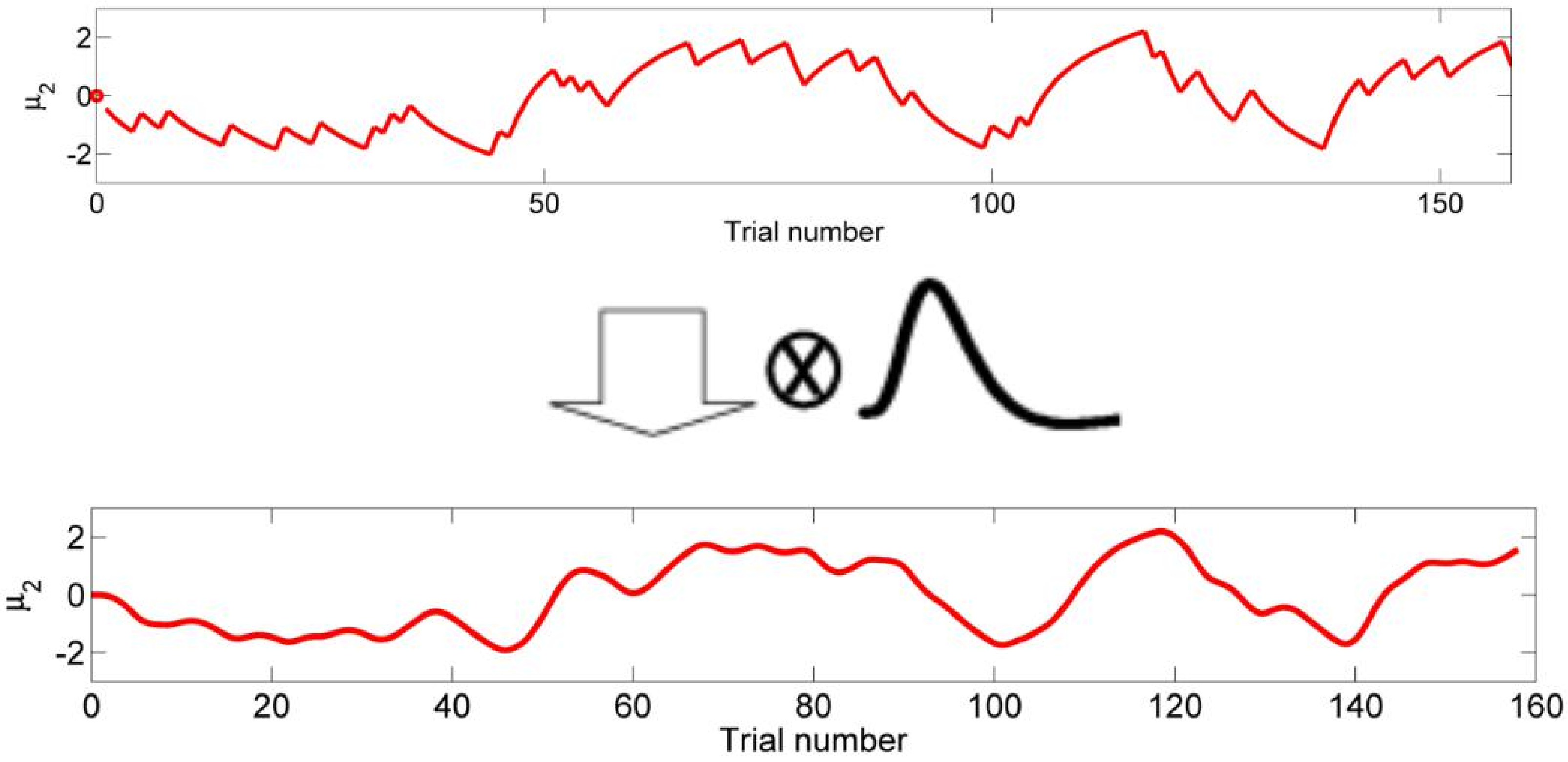
```
% Extract inferred delta(1): PE about arrow direction
pe = est.traj.da(:,1);
surprise = abs(pe); % take absolute value to get surprise

% Split into surprise about left vs. right arrow
surpriseLeft = surprise(u==0);
surpriseRight = surprise(u==1);
```





4. Convolve time series with HRF

- E.g.:

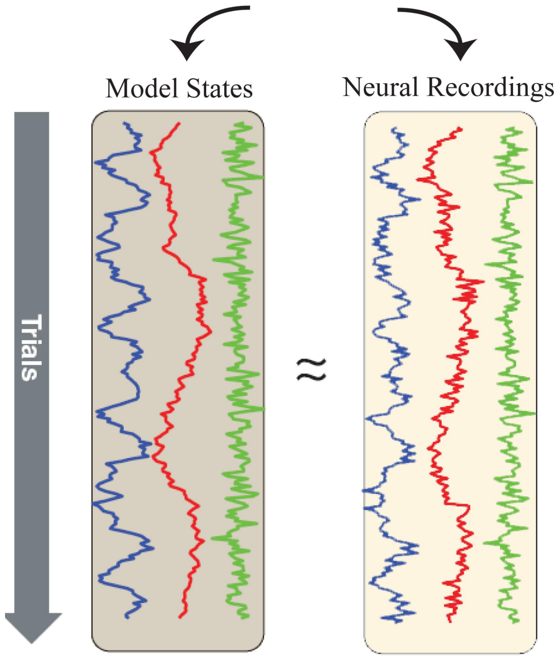


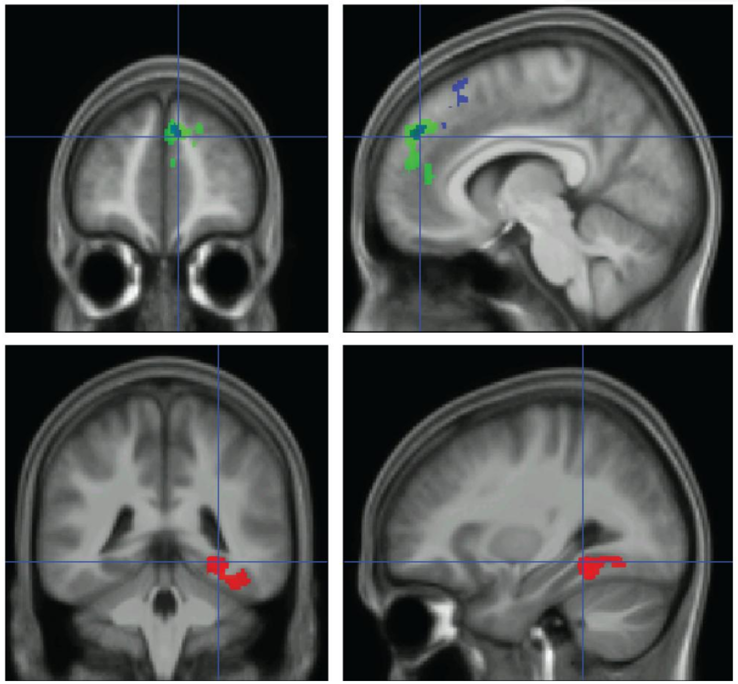
Adapted from O'Doherty et al., *Ann. N.Y. Acad. Sci.*, 2007

- In SPM 1st-level analysis

5. Regress against fMRI data

$$\mu_1^k = \mu_1^{(k-1)} + \sigma \cdot \delta$$



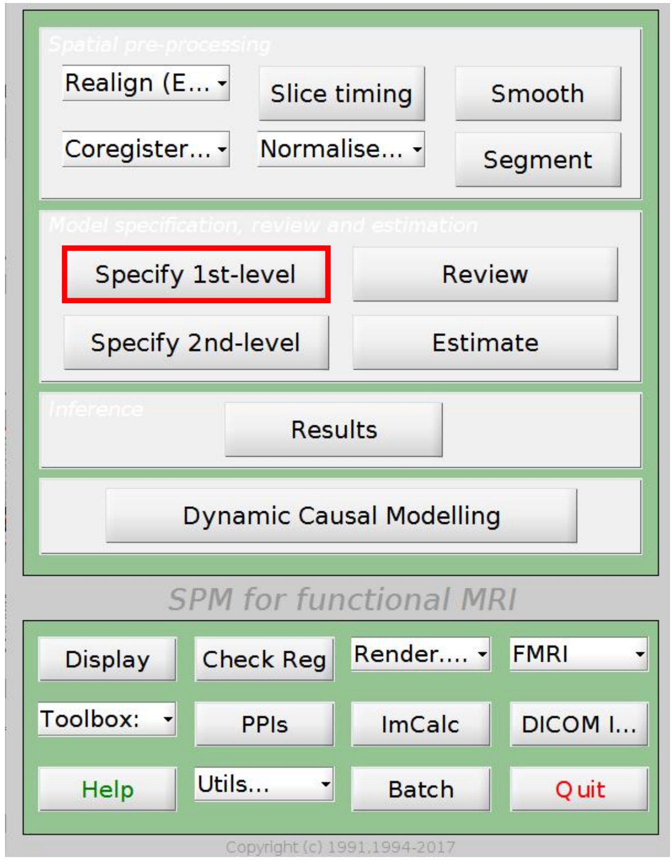


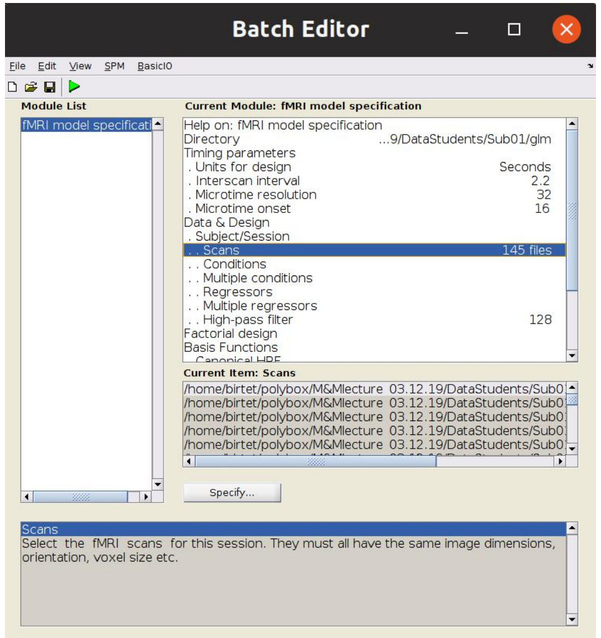
Adapted from Behrens et al., 2010

- In SPM 1st-level analysis

SPM 1st-level analysis

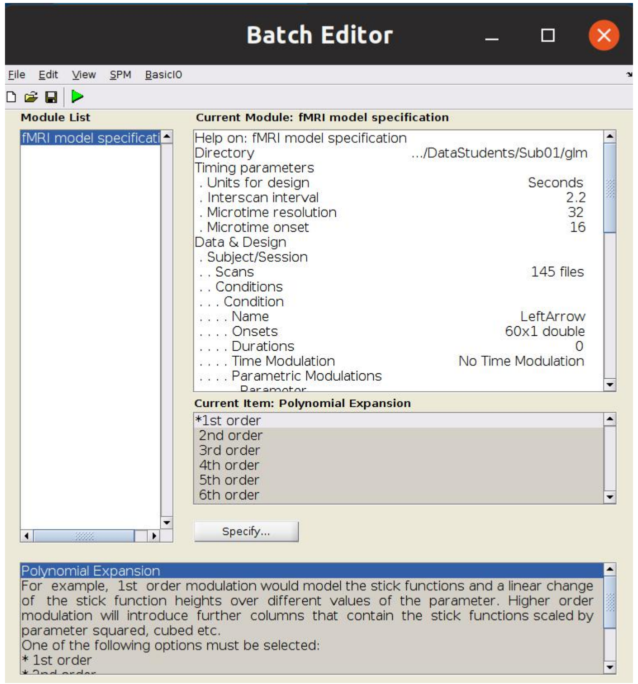
- SPM: Specify 1st-level
- Specify directory:
  - /Sub01/glm/modelbased
- Scanning parameters:
  - TR = 2.2 s
  - 32 slices
- Load data:
  - /Sub01/functional/ s8wafmri02.nii

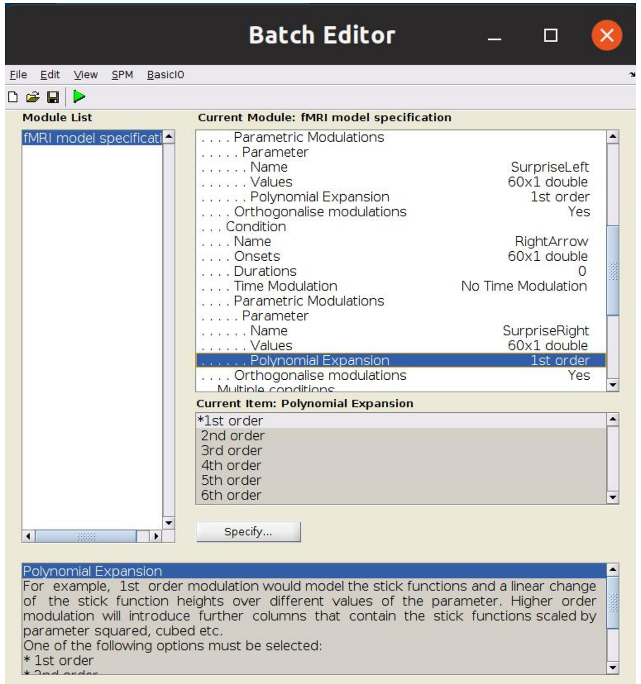


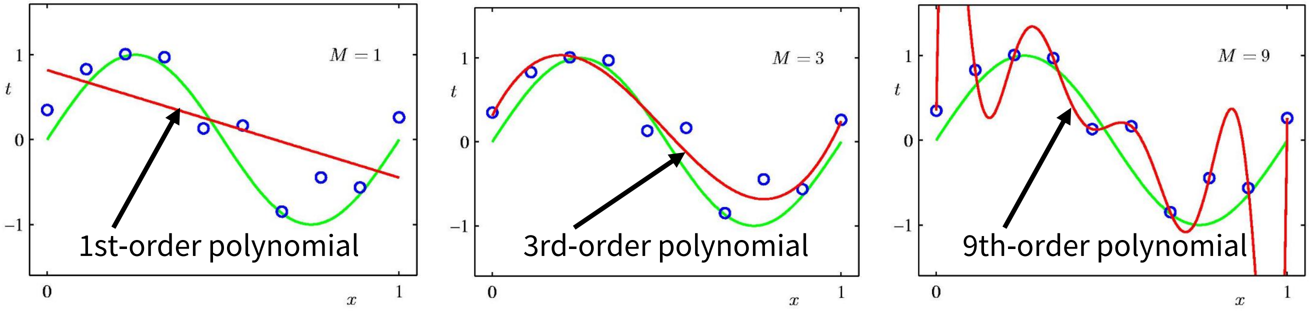


SPM 1st-level analysis

- Load regressors:
  - Condition 1: tLeftStim
    - Parametric modulation: SurpriseLeft
  - Condition 2: tRightStim
    - Parametric modulation: SurpriseRight
- Set Durations = 0
- Polynomial Expansion:
  - Choose 1st order

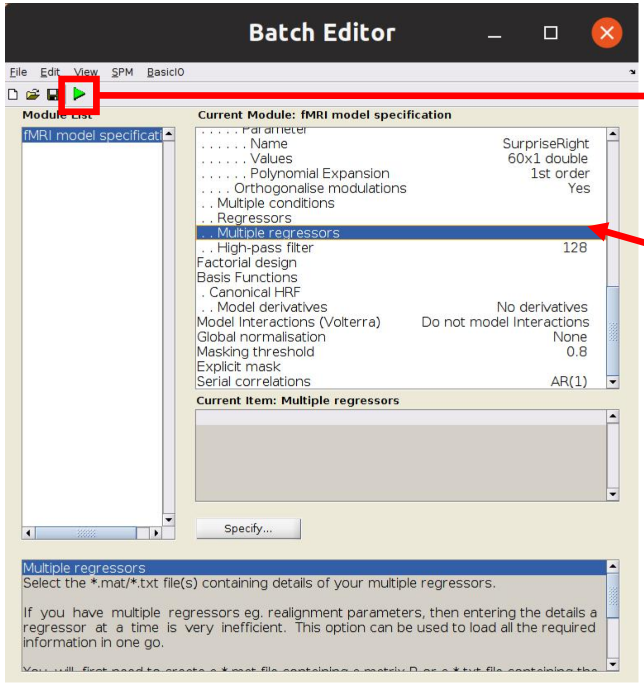






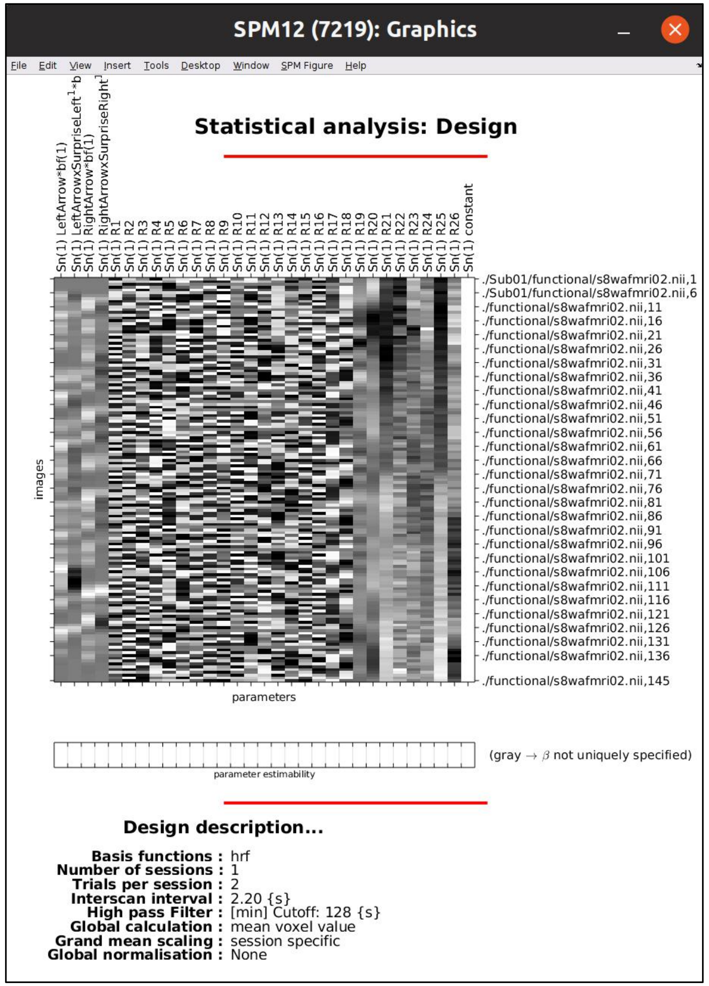
SPM 1st-level analysis

- Enter motion regressors
  - physio_regressors_run02.txt



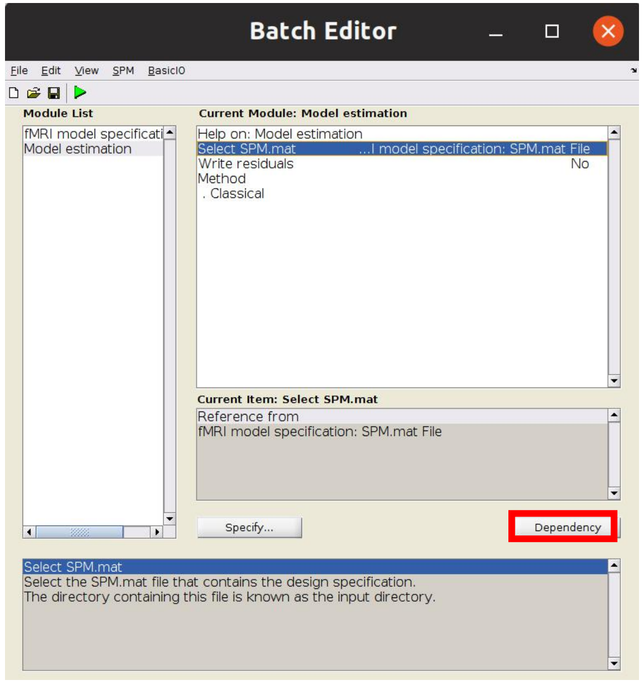
Run batch to view design matrix

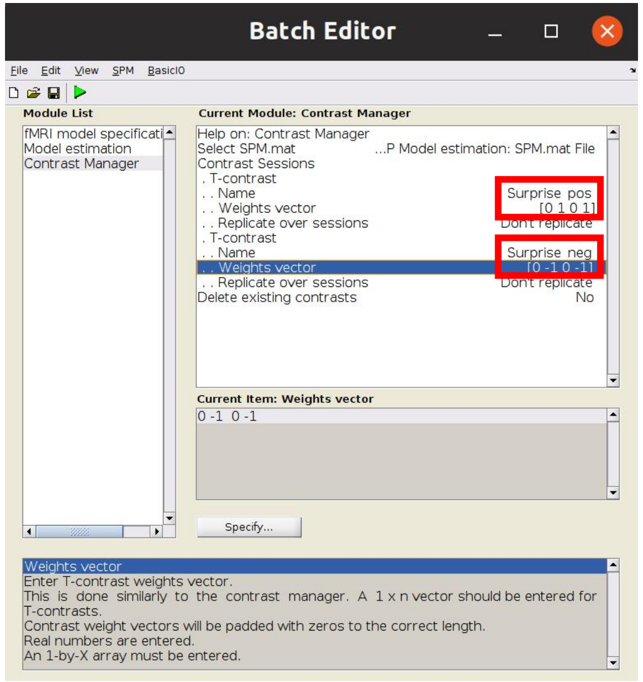
motion regressors



SPM 1st-level analysis

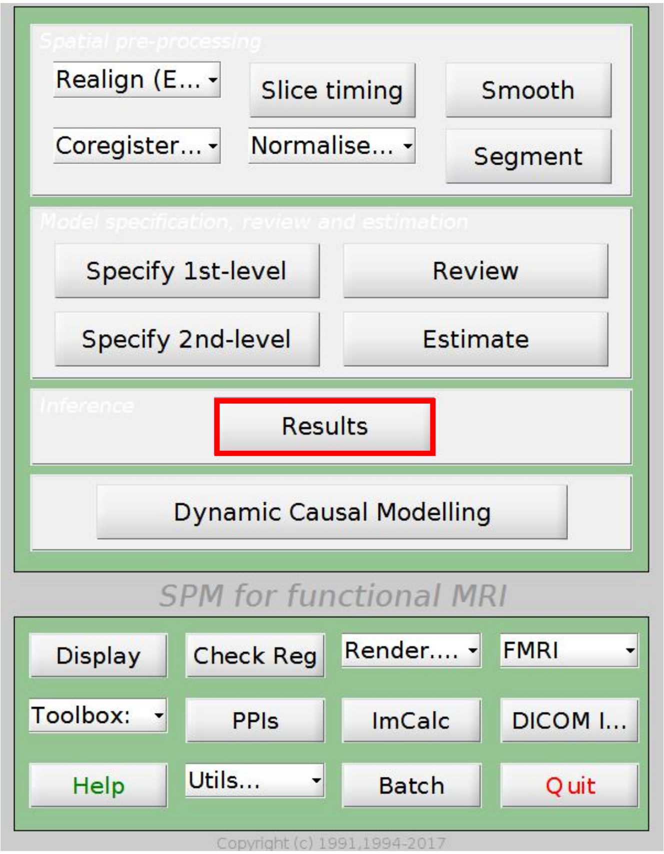
- Stats: Model estimation
  - Select SPM.mat
    - set Dependency
- Stats: Contrast manager
  - Select SPM.mat
    - set Dependency
  - Contrast Sessions
    - T-contrast: Surprise_pos
    - T-contrast: Surprise_neg
- Run batch





1st-level fMRI analysis

- View results
  - /Sub01/glm/SPM.mat

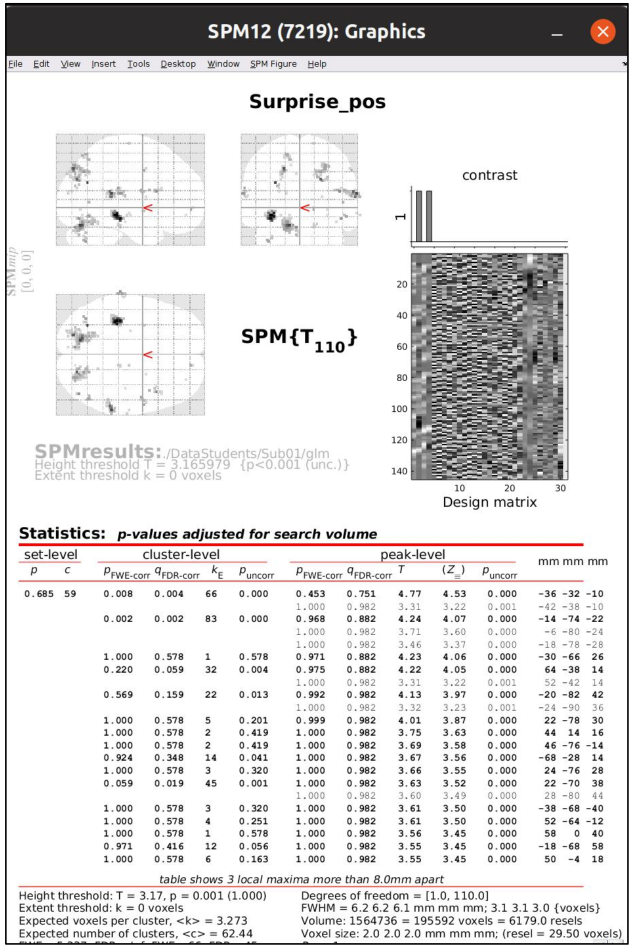


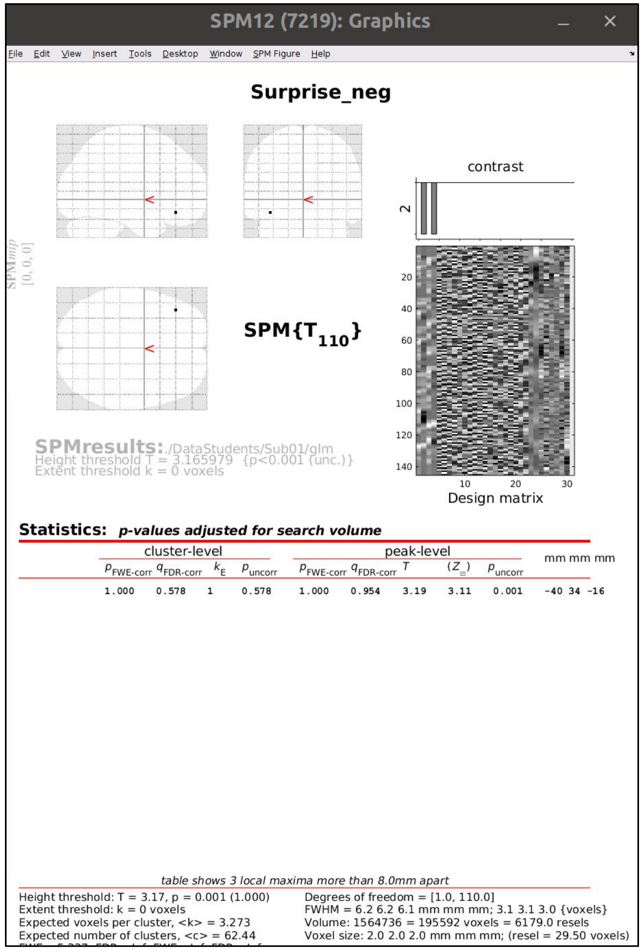
1st-level fMRI analysis

- Select contrast
  - apply masking: none
  - p val adjustment: none

→ Uncorrected results:

  - brain activity correlates with surprise about arrow direction

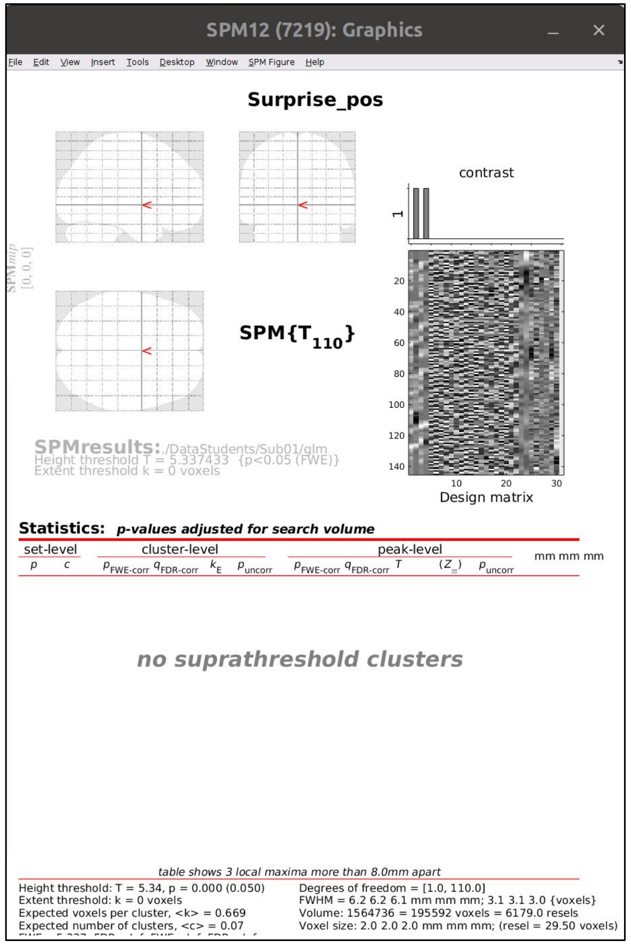


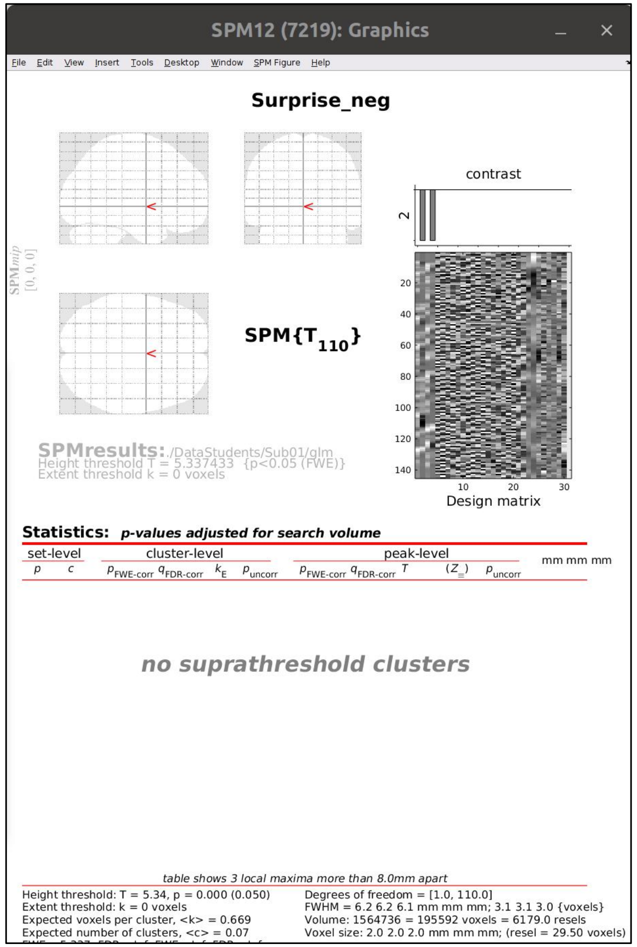


1st-level fMRI analysis

- Contrasts
  - Significance level
    - Set to 0.05 (FWE)

→ FWE-corrected results:
  - correlations not significant when we correct for multiple comparisons





2nd-level fMRI analysis

- What questions can we ask?

2nd-level fMRI analysis

- What questions can we ask?
  1. Where in the brain are PEs (about arrow direction) represented?

  2. Are there differences in brain activity depending on whether PEs occur in response to the arrow being presented on the left or right?

2nd-level fMRI analysis

- What questions can we ask?
  1. Where in the brain are PEs (about arrow direction) represented?

  2. Are there differences in brain activity depending on whether PEs occur in response to the arrow being presented on the left or right?

- Different 2nd-level design matrices:

2nd-level fMRI analysis

- What questions can we ask?
  1. Where in the brain are PEs (about arrow direction) represented?

  2. Are there differences in brain activity depending on whether PEs occur in response to the arrow being presented on the left or right?

- Different 2nd-level design matrices:
  1. Average over parametric regressors:
  - Set surpriseLeft = 1 & surpriseRight = 1, one-same t-test

  2. Difference between parametric regressors:
  - Set surpriseLeft = 1 & surpriseRight = -1, one-sample t-test