

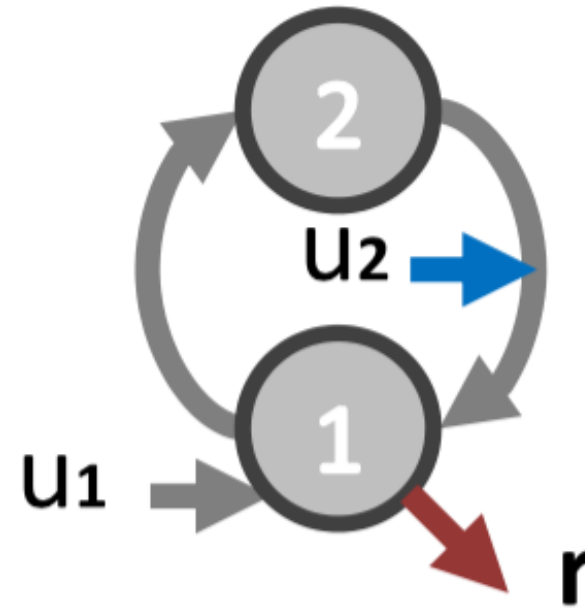


Behavioural DCM *in practice*

Lionel Rigoux

Outline

- Preparing the data
 - Extracting ROI time series
 - Behavioural responses and resampling
 - Defining the inputs
- Specifying the model
 - DCM connectivity
 - Behavioural mapping
 - Behavioural response transformation
- Model simulation & inversion
 - Sources and hyperpriors
 - Wrapping up
- Model Simulation and analysis
 - Artificial lesions
 - Susceptibility analysis



Preparing the data

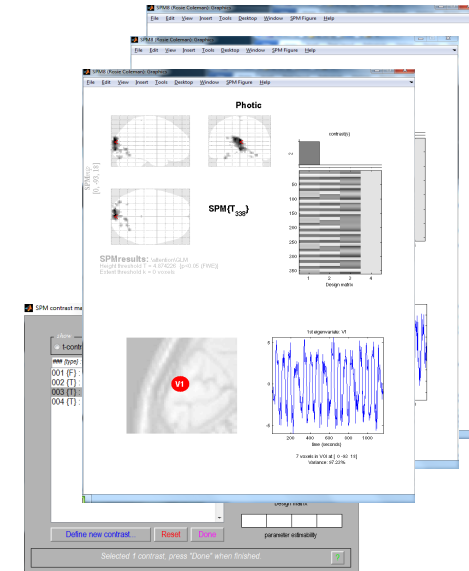
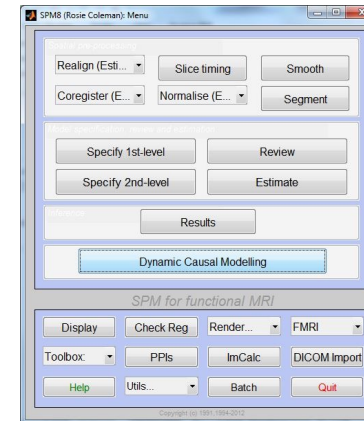
Extracting timeseries

From a DCM specified in SPM:

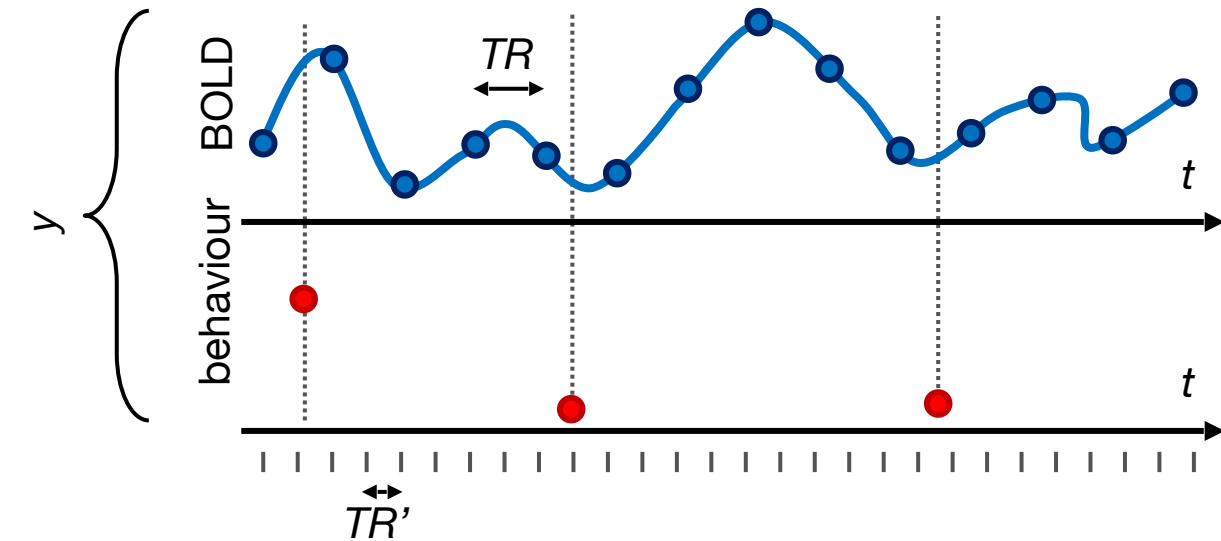
- DCM.y'

Using SPM:

- Define your ROI
- Extract 1st eigenvariate
- Adjust for effect of interest
- Save



Preparing the data



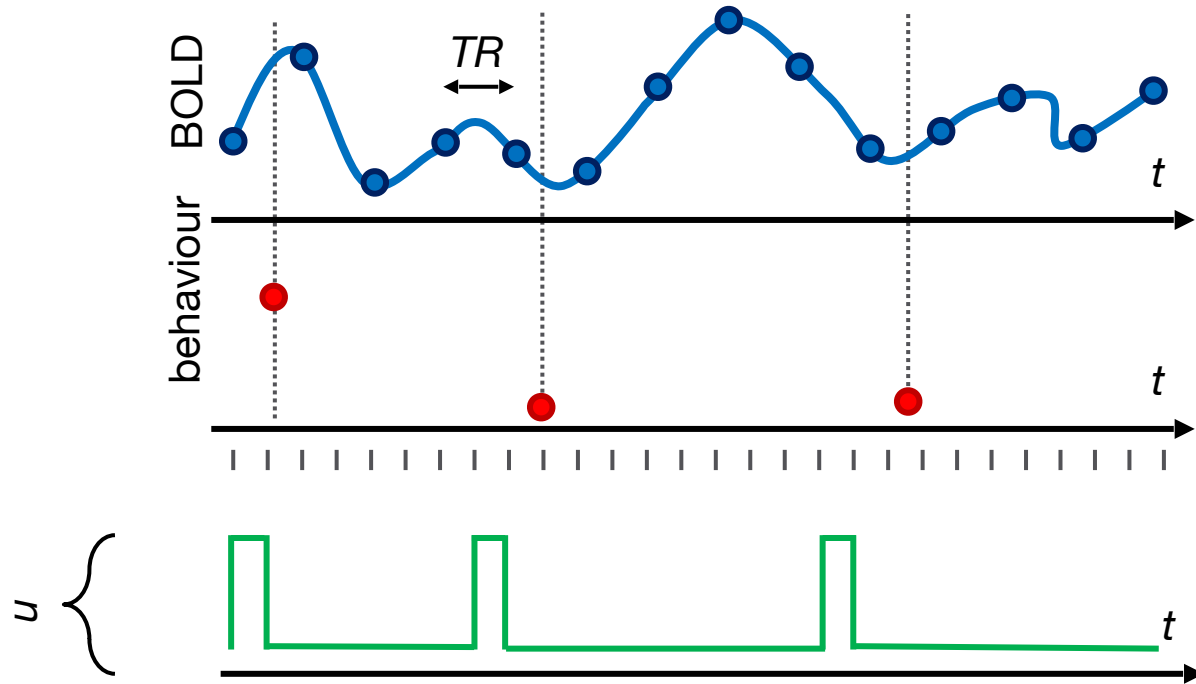
In practice

- Resample BOLD and behaviour at the same frequency (eg. $TR/4$ or $TR/20$)
- Fill in missing value with NaNs
- Store in one matrix $y = [y_bold ; y_resp]$

Tips

- Increasing the frequency will slow down the inversion
- Response predictor is smoothed by a kernel: round to the next timestep

Preparing the data



In practice

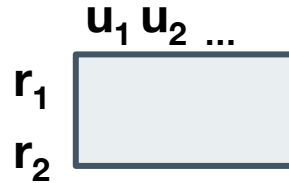
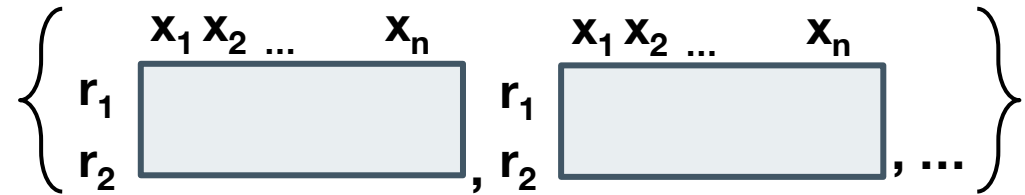
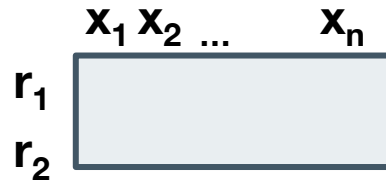
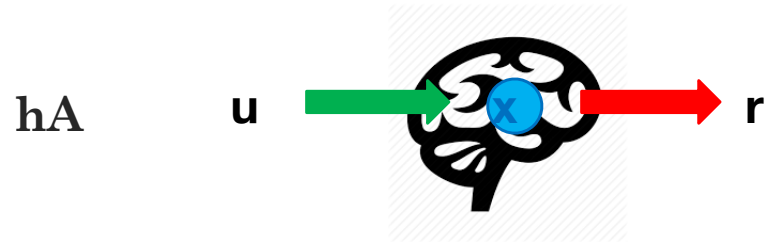
- You can use `DCM.U.u'`
- Resample the inputs:
 - at the same frequency as y
 - at the microtime resolution `DCM.U.dt`

Specifying the model

Define the brain connectivity model

A = DCM.a	intrinsic connectivity
B = DCM.b	modulatory influences
C = DCM.c	inputs
D = DCM.d	quadratic effects

Specifying the model



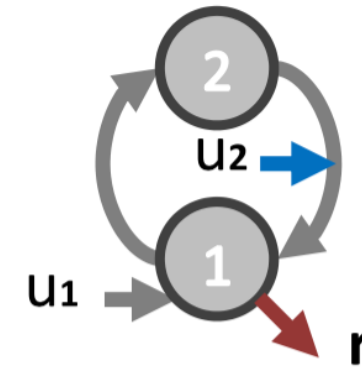
Specifying the model

Define the brain connectivity model

A = DCM.a intrinsic connectivity
B = DCM.b modulatory influences
C = DCM.c inputs
D = DCM.d quadratic effects

Define the behavioural mapping

hA neuro-behavioural mapping
hB neural modulation of inputs
hC direct input



$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$hA = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Specifying the model

Define the behavioural transformation

- Dynamics of the bDCM is computed by the evolution function $f_DCMwHRFext$

$$\begin{bmatrix} \dot{x} \\ \dot{r} \end{bmatrix} = f\left(\begin{bmatrix} x \\ r \end{bmatrix}, u, \theta\right)$$

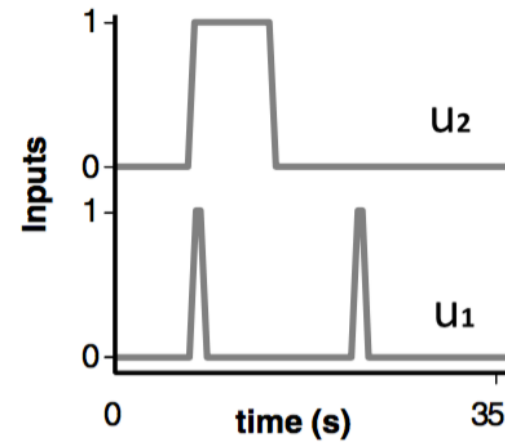
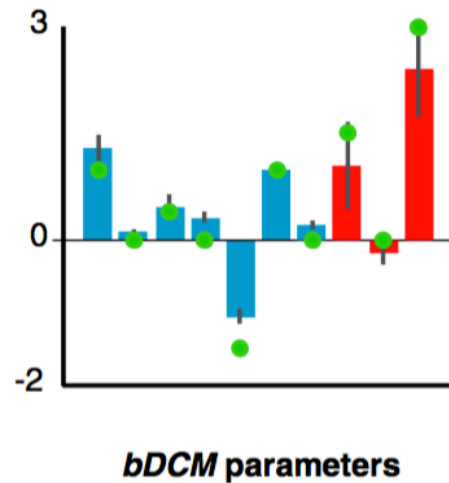
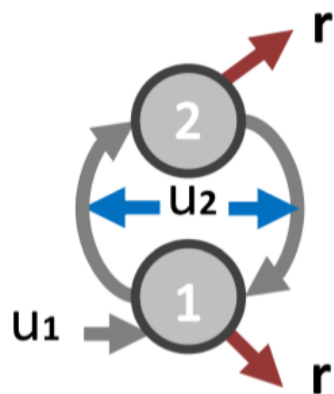
- Need to map the response predictor(s) r onto the response
 - sigmoid for binomial response
 - exponential for RT
 - softmax for multinomial
 - etc.
- The HRF and the response transformation must be combined into one observation function, eg. $g_DCMwHRFext$

$$\begin{bmatrix} y_{mri} \\ y_{beh} \end{bmatrix} = g\left(\begin{bmatrix} x \\ r \end{bmatrix}, u, \phi\right)$$

Model simulation

Model simulation

- Set the parameters of the connectivity and the mapping
- $y = \text{simulateNLSS}(n_t, f_fname, g_fname, \theta, \phi, u, \alpha, \sigma, \text{options});$



Model inversion

Source specification

- Need to use the different data types differently
 - Different distributions
 - Different noise variances
- > *options.sources* to split the observations *y* into separate distributions

Hyperpriors

- All sources are influencing the inversion according to their respective precision
- Hyperpriors define the (prior) precision of Gaussian sources
- > *getHyperpriors()* to set priors as a function of expected explained variance

Model inversion

You need

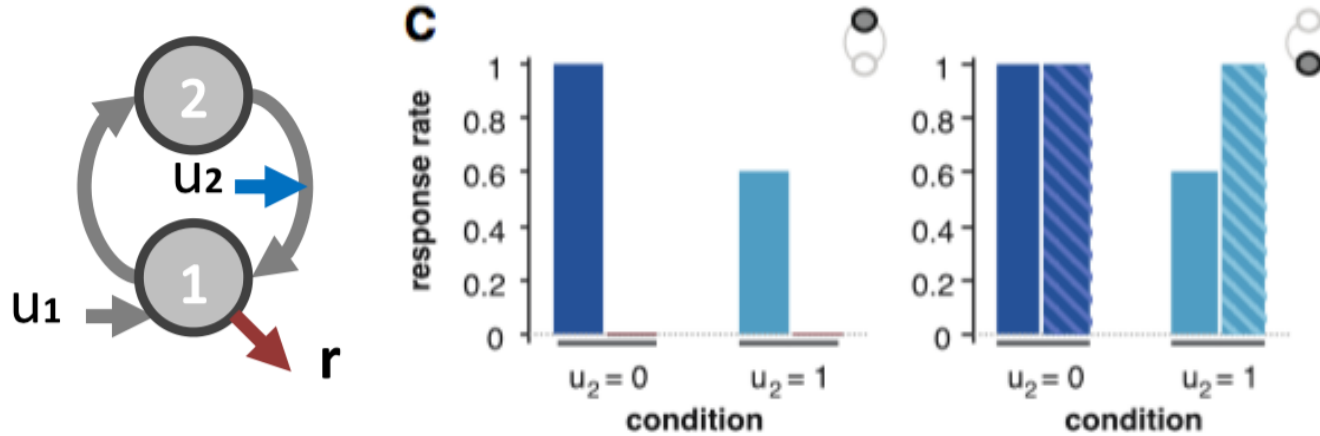
- Combined observations y and inputs u
- *options* structure containing the model definition: *prepare_fullDCM*
- Combined observation function
- Add priors to the *options*: *getPriors*
- Specify how to split the data: *options.sources*

Then run

[posterior,out] = VBA_NLStateSpaceModel(y,u,f_fname,g_fname,dim,options);

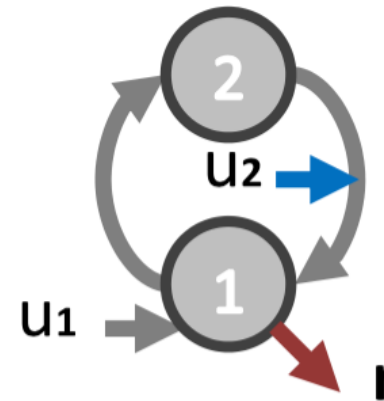
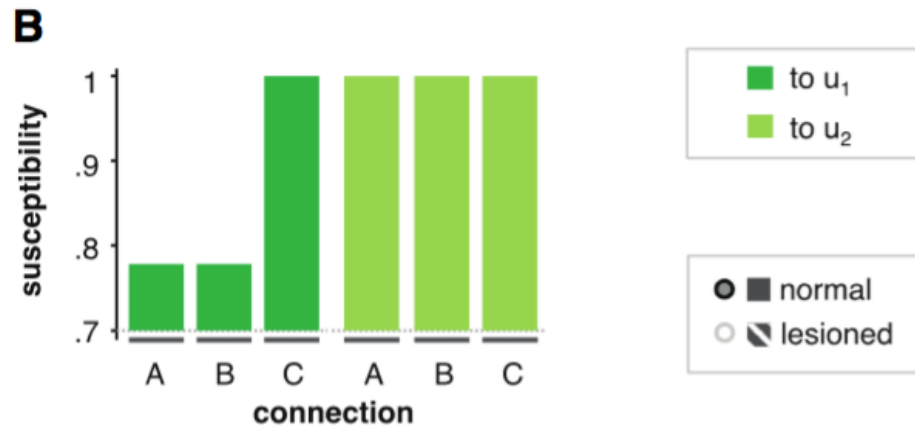
Virtual lesions

- Switch off all the afferences to a ROI and simulate the reduced model
- $res = VBA_bDCM_lesion(posterior, out);$



Susceptibility analysis

- $res = VBA_susceptibility(posterior, out);$
- res



Thank you