MDS-MCMC February 2017

Tue Herlau

Technical University of Denmark

tuhe@dtu.dk

$s_k(t)$ the neural signal in region k
$u_k(t)$ the stimulus in region k (box functions)
$y_k(t)$ Observed (fMRI) signal in region k (convolution of s_k)

Underlying neural dynamic is:

$$\frac{\partial \boldsymbol{s}(t)}{\partial t} = C\boldsymbol{s}(t) + \text{Stimulus} + \text{Noise}$$

Question: What is the right model for the stimulus? (U is diagonal)

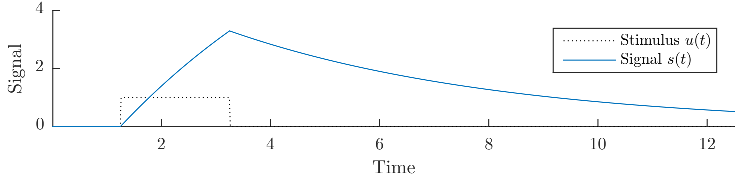
$$\text{Current: } \frac{\partial \boldsymbol{s}(t)}{\partial t} = C\boldsymbol{s}(t) + U\boldsymbol{u}(t)$$

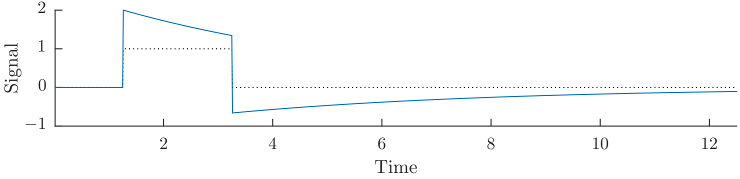
$$\text{Alternative: } \frac{\partial \boldsymbol{s}(t)}{\partial t} = C\boldsymbol{s}(t) + U\frac{\partial \boldsymbol{u}(t)}{\partial t}$$

What does this mean?

$$\text{Current (top): } \frac{\partial \boldsymbol{s}(t)}{\partial t} = C\boldsymbol{s}(t) + U\boldsymbol{u}(t)$$

$$\text{Alternative (bottom): } \frac{\partial \boldsymbol{s}(t)}{\partial t} = C\boldsymbol{s}(t) + U\frac{\partial \boldsymbol{u}(t)}{\partial t}$$





$C = \frac{1}{5}$, $U = 2$: Top has slow rampup, slow decline. Bottom allows recovery of boxfunction ($C = 0$)

Start with:

$$\frac{\partial \boldsymbol{s}(t)}{\partial t} = C\boldsymbol{s}(t) + U\frac{\partial \boldsymbol{u}(t)}{\partial t} + \text{Noise}$$

Formally:

$$ds_t = Cs_t dt + Udu_t + D^{\frac{1}{2}} w(dt)$$

(w is a Wiener noise process). Using finite-time discretization:

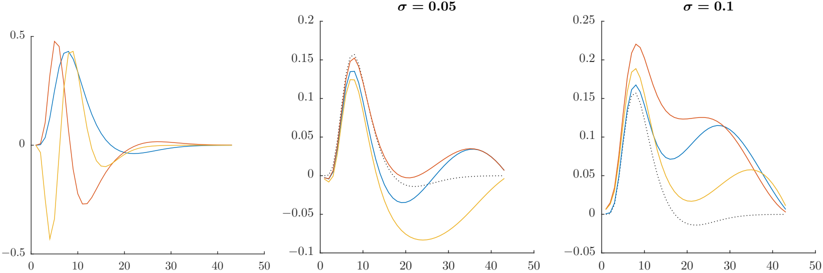
$$s_{t+1} = (I + dtC)s_t + U(u_{t+1} - u_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, Ddt)$$

$$y_{kt} = B_k^T \Phi \begin{bmatrix} s_{k,t} & s_{k,t-1} & \cdots, & s_{k,t-L} \end{bmatrix}^T$$

$$s_{t+1} = (I + dtC)s_t + U(u_{t+1} - u_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, Ddt)$$
$$y_{kt} = B_k^T \Phi \begin{bmatrix} s_{k,t} & s_{k,t-1} & \cdots, & s_{k,t-L} \end{bmatrix}^T$$

Two options for Φ_l:

- Use fixed functions for Φ_l (MDS):
- Let $\Phi_l \sim \mathrm{GP}(\mu_0, \kappa_\sigma)$ conditional on $\Phi_l(0) = \Phi_l(L) = 0$



Prior for C. The important bit is a prior that promotes sparsity. Tried various choices:

$$\text{ARD: } x_j \sim \mathcal{N}(0, \sigma_j^2),\ \sigma^{-2} \sim \mathrm{Gam}(a, b)$$

$$\text{Student half-}t\text{: } x_j \sim \mathcal{N}(0, \sigma_j^2),\ \sigma \sim p_{\frac{1}{2}t}(\nu, A)$$

$$p_{\frac{1}{2}t}(x|\nu, A) = \frac{2}{A\sqrt{v}B(\frac{1}{2}, \frac{\nu}{2})}\left(1 + \frac{1}{\nu}\left(\frac{\sigma}{A}\right)^2\right)^{-\frac{\nu+1}{2}}$$

$$\text{Dirichlet-Laplace shrinkage } x_j \sim \mathcal{N}(\mu_j, \sigma^2 = q_k \phi_k^2 \tau^2)$$

$$q_k \sim \mathrm{Exp}(\frac{1}{2})$$

$$\theta \sim \mathrm{Dir}(a, \ldots, a)$$

$$\tau \sim \mathrm{Gam}(Ma, \frac{1}{2})$$

$$\epsilon_{kt} \sim \mathcal{N}(0, \epsilon_{k,n}^2)$$

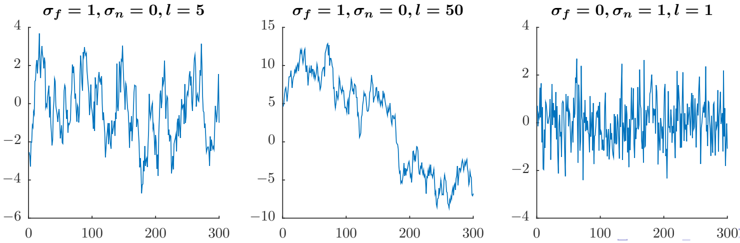
A more general prior for the noise is an $\mathrm{AR}(1)$ autoregressive prior

$$\begin{aligned} \epsilon_{k,t} &= \tilde{\epsilon}_k + \mu_{k,0} \\ \tilde{\epsilon}_{k,t+1} &= Q_k \tilde{\epsilon}_{k,t} + \delta_{k,t}, \quad \delta_{k,t} \sim \mathcal{N}(0, \sigma_k^2) \end{aligned}$$

The continious-time analogue is the Ornstein-Uhlenbeck (OU) process. That's just a fancy word for a Gaussian process with a particular kernel:

$$\epsilon_k \sim \mathrm{GP}(\mu_{k0}, \kappa)$$

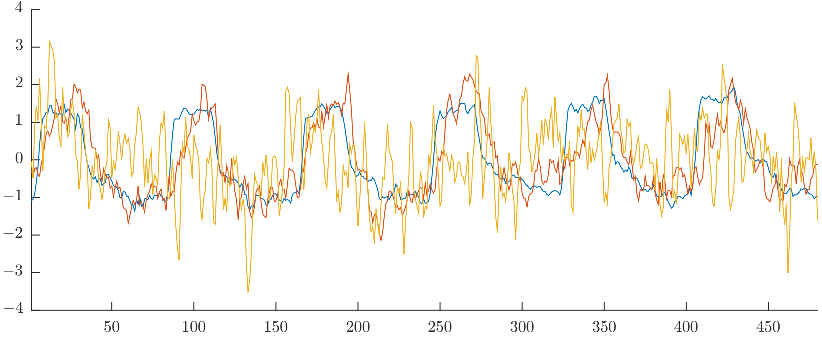
$$\kappa(x, x') = \sigma_f^2 e^{-\frac{\|x-x'\|}{l}} + 1_{x-x'} \sigma_n^2$$



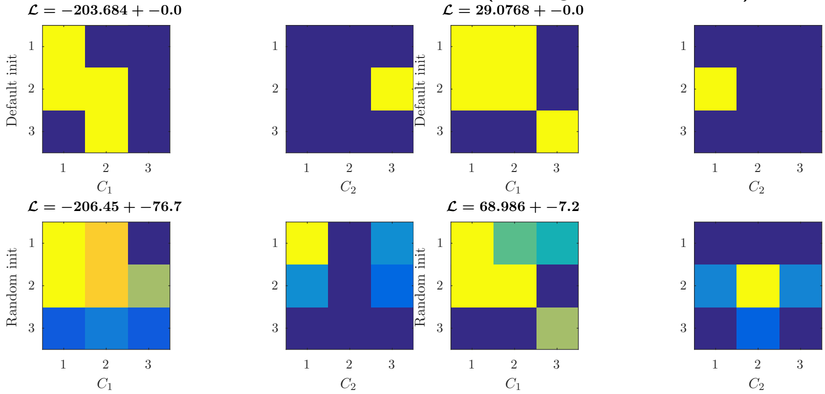
$$
\begin{aligned}
s_{t+1} &= (I + dtC)s_t + U(u_{t+1} - u_t) + \epsilon_t \\
y_{kt} &= B_k^T \Phi \begin{bmatrix} s_{k,t} & s_{k,t-1} & \cdots, & s_{k,t-L} \end{bmatrix}^T
\end{aligned}
$$

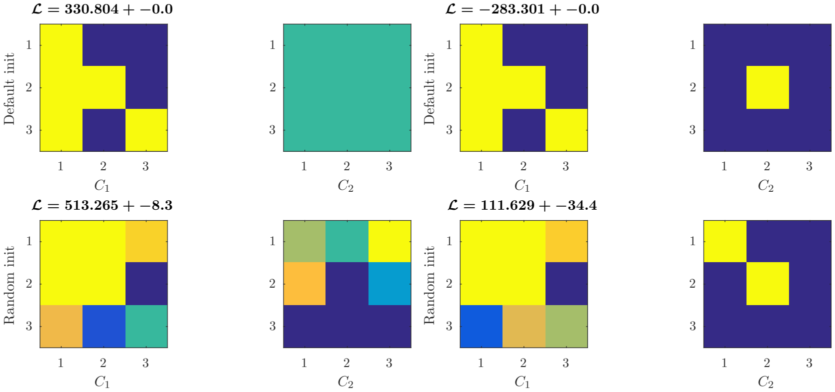
- Tried Gibbs sampling for each time slice ϵ_t, too slow
- Instead: Parameterize using $\boldsymbol{s}$ and sample $\boldsymbol{s} \sim \mathcal{N}(\mu_{TK\times 1}, \Sigma_{TK\times TK})$.
- Reasonably fast/numerically stable implementation possible using Kronecher products and Cholesky factorizations

Rat dataset (3 rats, 3 regions, 480 time points)

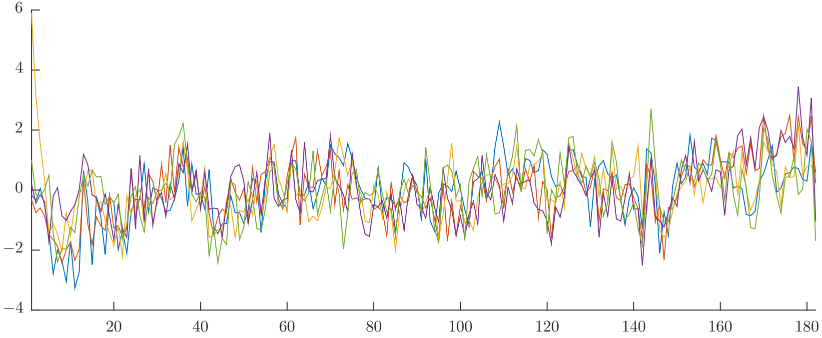


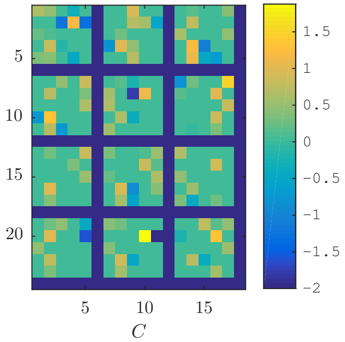
For each rat, find the two matrices C_1, C_2 with MDS using standard initialization & random initialization (Averaged over 10 runs)

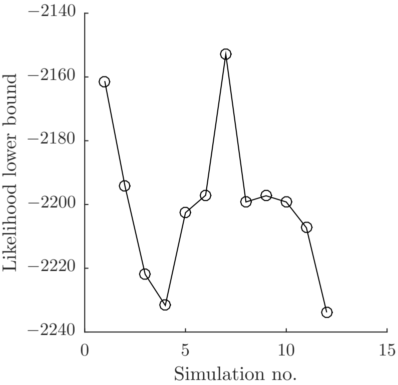



$\mathcal{L} = 330.804 + -0.0$
$\mathcal{L} = -283.301 + -0.0$
$\mathcal{L} = 513.265 + -8.3$
$\mathcal{L} = 111.629 + -34.4$
Default init
Random init
1
2
3
C_1
C_2

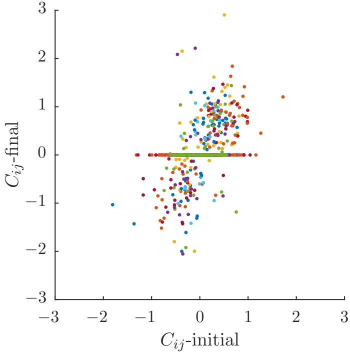
OpenfMRI dataset (19 subjects, 5 regions, 182 time points)

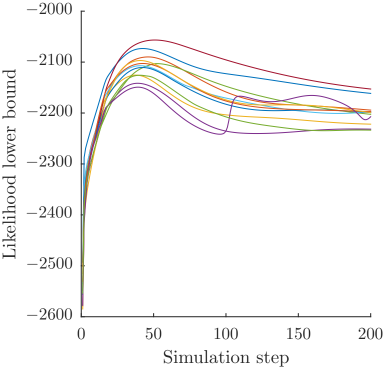






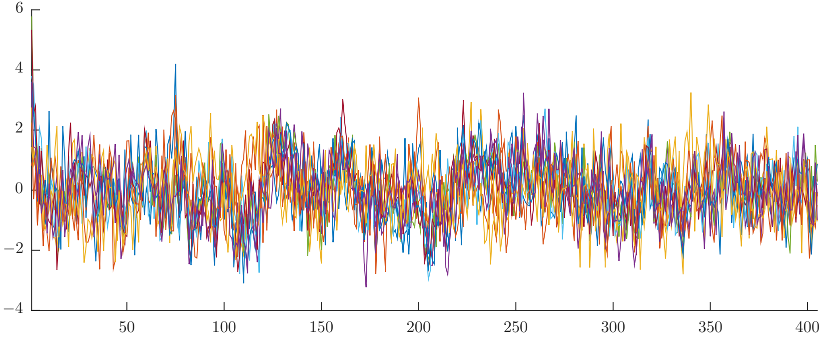
- Left: Inferred matrices C for 11 random initializations (upper-left is default initialization)
- Right: Lower-bound of log-likelihood after convergence. Higher is better

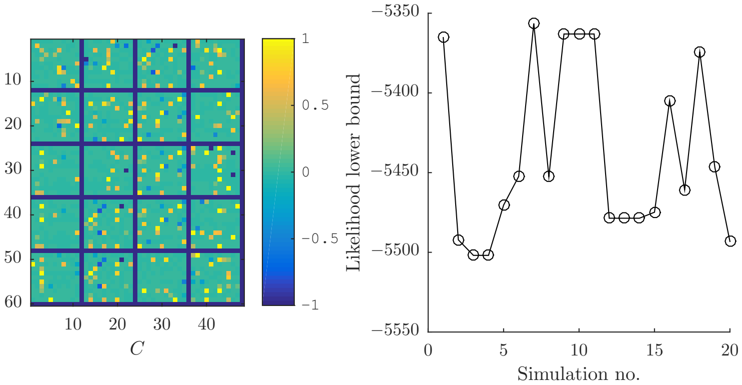




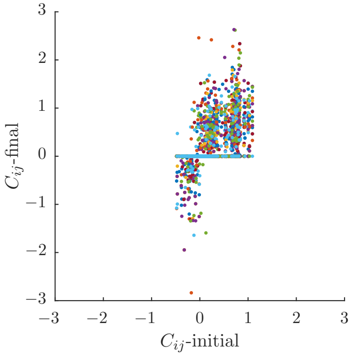
- Left: Initial vs. final value of matrix entries C for all simulations
- Right: Log-likelihood as a function of VB update

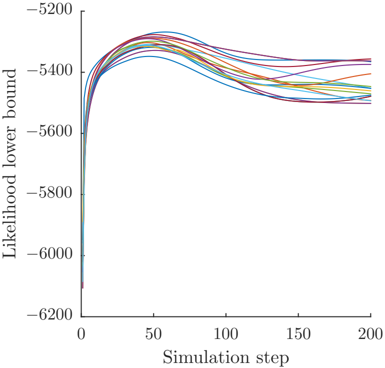
HW dataset (21 subjects, 11 regions, 405 time points)



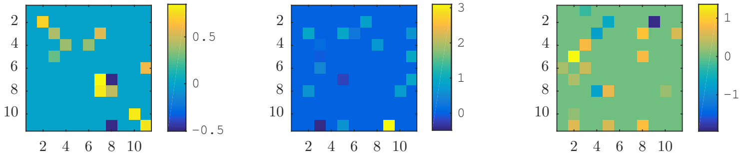


- Left: Inferred matrices C for 19 random initializations (upper-left is default initialization)
- Right: Lower-bound of log-likelihood after convergence. Higher is better





- Left: Initial vs. final value of matrix entries C for all simulations
- Right: Log-likelihood as a function of VB update

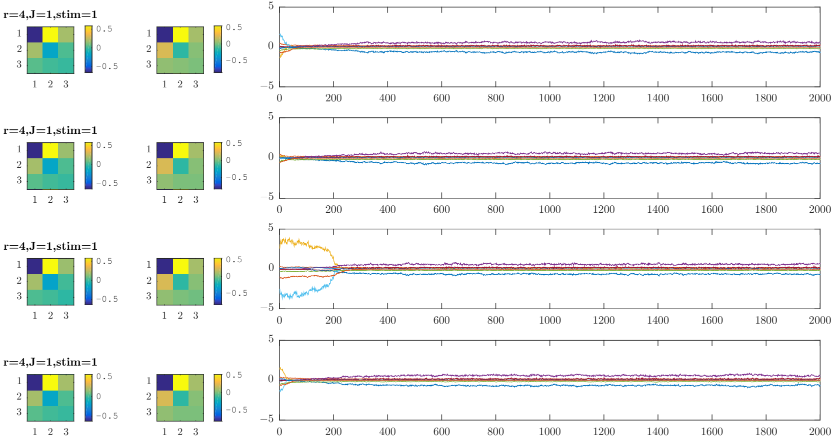


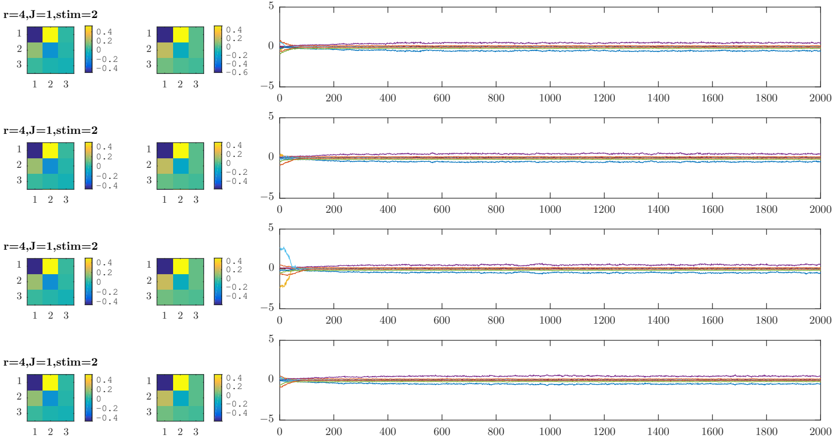
- Comparison of 3 matrices with highest likelihood
- If we focus on the highest-valued entries, we would draw fully opposite conclusions for each of the runs. Yet they represents the same model on the same data..

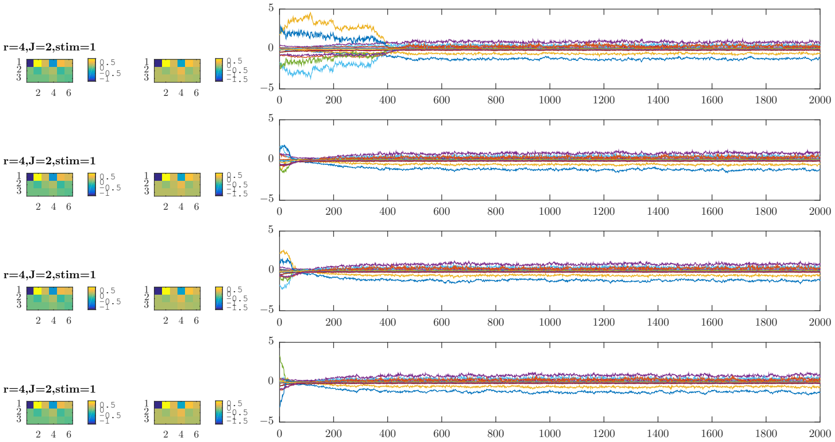
Things that can be varied:

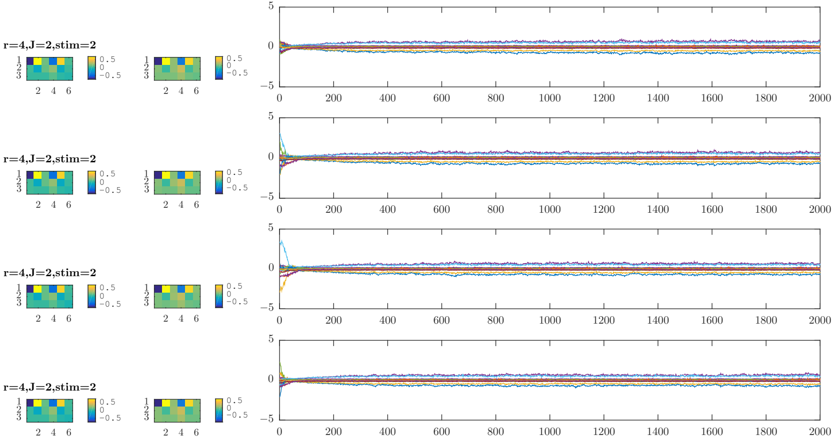
- J (number of C-matrices)
- If stimuli is used or not $u_t = 0, 1$
- How the noise is handled $(\mathrm{AR}(1)$ or iid)
- Priors

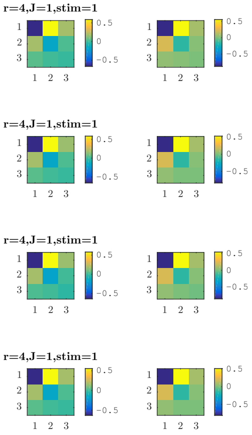
Numerical instability (i.e. very large values of C and s) were a common feature when Φ, B was inferred and only one rat was used. The canonical bold response function will therefore be fixed in these experiments

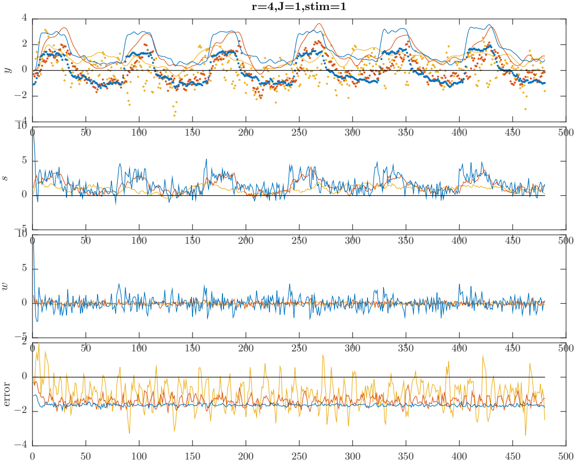

r=4,J=1,stim=1
r=4,J=1,stim=1
r=4,J=1,stim=1
r=4,J=1,stim=1

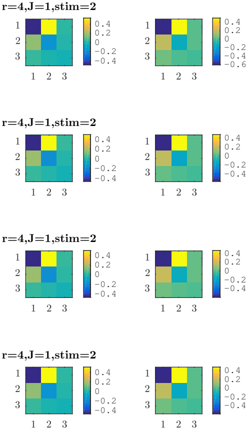

r=4,J=1,stim=2
r=4,J=1,stim=2
r=4,J=1,stim=2
r=4,J=1,stim=2
0.4
0.2
0
−0.2
−0.4
−0.6
5
0
−5
0
200
400
600
800
1000
1200
1400
1600
1800
2000

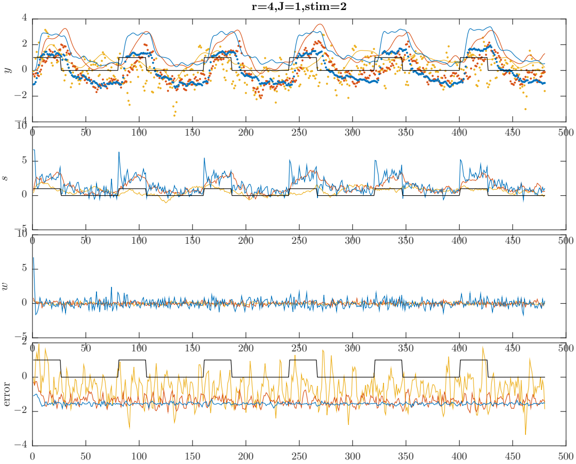

r=4,J=2,stim=1
r=4,J=2,stim=1
r=4,J=2,stim=1
r=4,J=2,stim=1

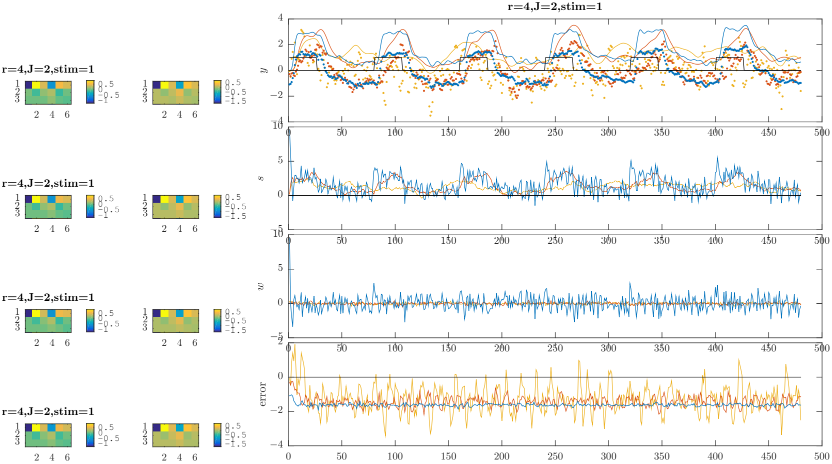

r=4,J=2,stim=2
r=4,J=2,stim=2
r=4,J=2,stim=2
r=4,J=2,stim=2
0.5
0
-0.5
5
0
−5
0
200
400
600
800
1000
1200
1400
1600
1800
2000

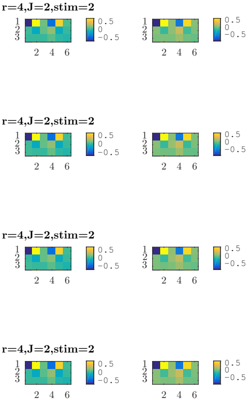

r=4,J=1,stim=1
r=4,J=1,stim=1
r=4,J=1,stim=1
r=4,J=1,stim=1
0.5
0
-0.5

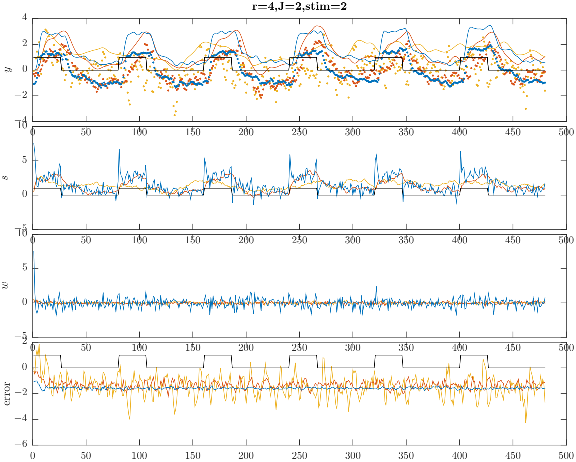

r=4,J=1,stim=1
y
s
w
error

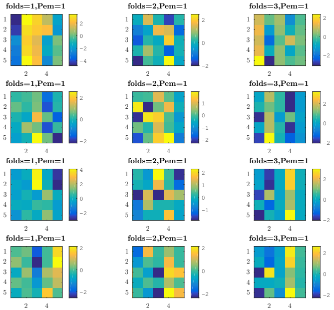

r=4,J=1,stim=2
r=4,J=1,stim=2
r=4,J=1,stim=2
r=4,J=1,stim=2

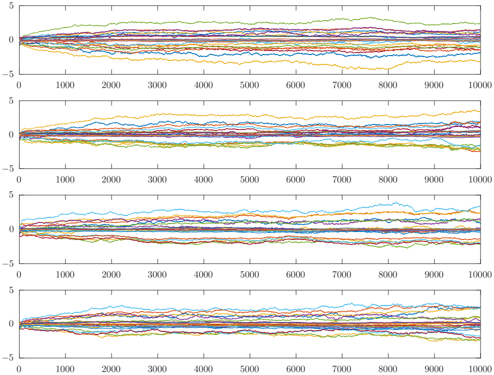

r=4,J=1,stim=2
y
s
w
error

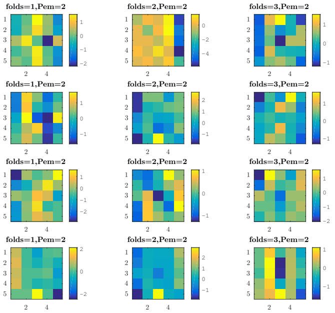

r=4,J=2,stim=1
r=4,J=2,stim=1
r=4,J=2,stim=1
r=4,J=2,stim=1
r=4,J=2,stim=1
y
s
w
error

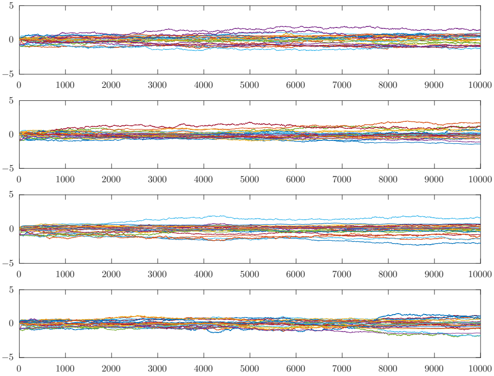

r=4,J=2,stim=2
r=4,J=2,stim=2
r=4,J=2,stim=2
r=4,J=2,stim=2

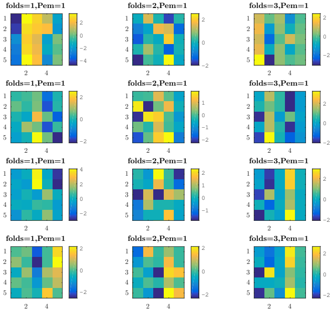

r=4,J=2,stim=2
y
s
w
error

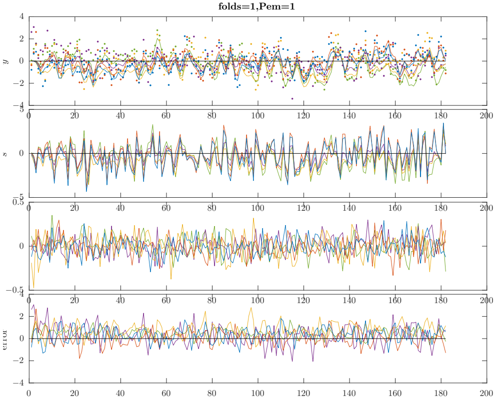

folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1

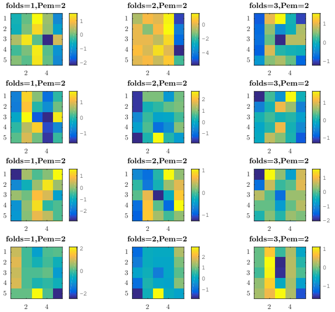


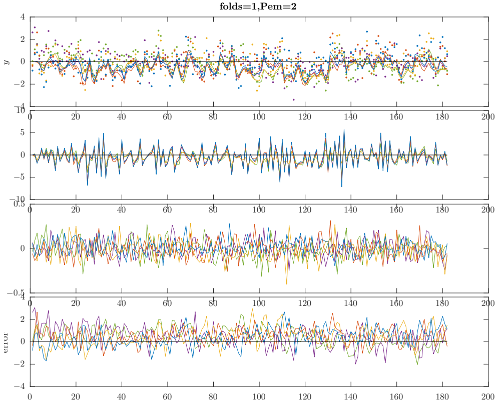

folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2

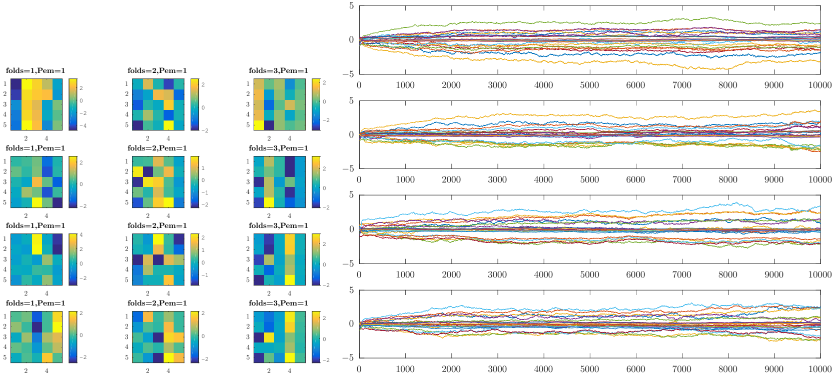


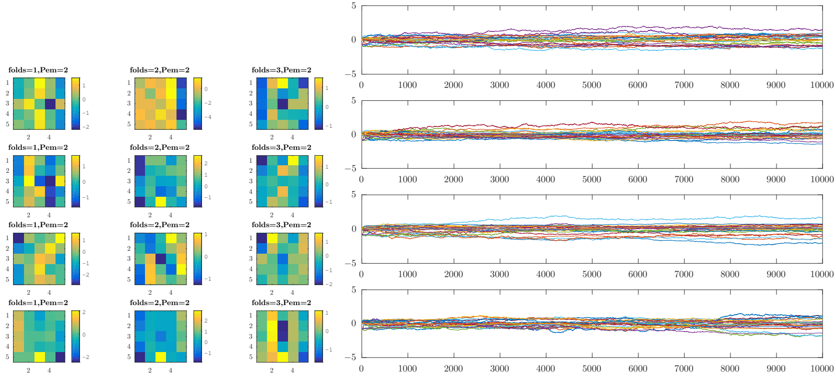

folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1

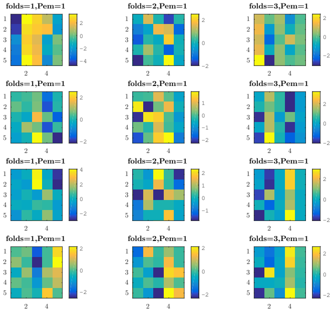

folds=1,Pem=1
y
s
error

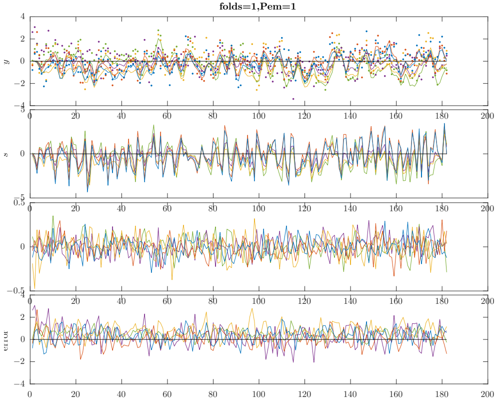

folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2

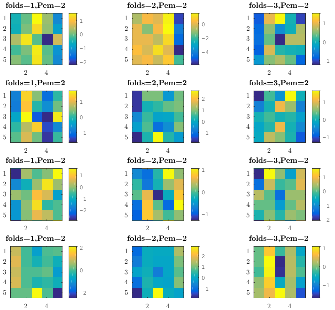

folds=1,Pem=2
y
error

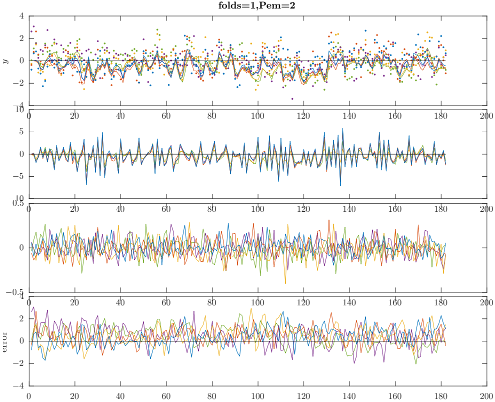

folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1
folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1


folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

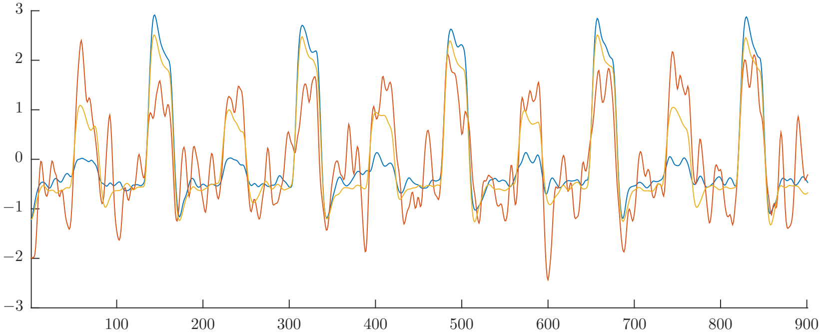

folds=1,Pem=1
folds=2,Pem=1
folds=3,Pem=1

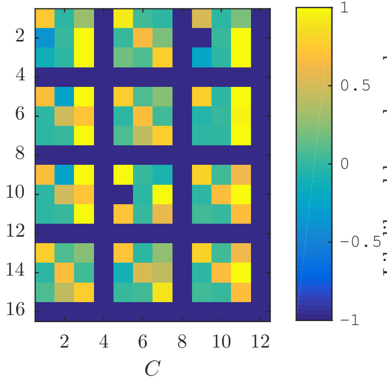

folds=1,Pem=1
y
s
error

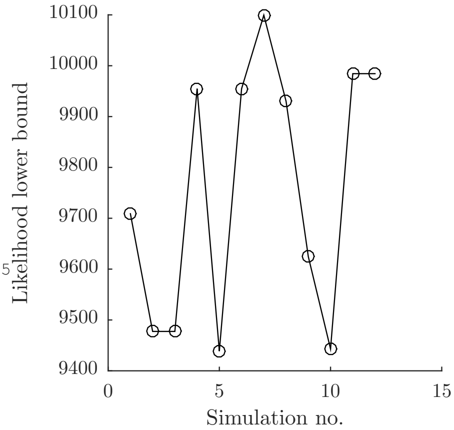

folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2
folds=1,Pem=2
folds=2,Pem=2
folds=3,Pem=2


folds=1,Pem=2
y
error

MDS on artificial data set (using 3 regions, 10 subjects, 410 data points)




2
4
6
8
10
12
14
16
2
4
6
8
10
12
C
1
0.5
0
−0.5
−1


Likelihood lower bound
10100
10000
9900
9800
9700
9600
9500
9400
0
5
10
15
Simulation no.

Inferred C matrices with highest likelihood.