

# Notes: Markov Random Fields and fMRI

Wei Liu

August 19, 2010

## Contents

<b>1 Ising Model</b>	<b>2</b>
1.1 Generate Ising model by Gibbs Sampling . . . . .	2
1.2 Jan 19 Generate 1D fMRI image and compute posterior distribution . . . . .	3
<b>2 New Reading on Functional Connectivity</b>	<b>7</b>
<b>3 Apr 6 Meeting</b>	<b>7</b>
<b>4 Apr 20</b>	<b>7</b>
<b>5 GPUsampling code change memo</b>	<b>8</b>
<b>6 may 18</b>	<b>12</b>
<b>7 May 25</b>	<b>13</b>
<b>8 Jun 9</b>	<b>14</b>
<b>9 Jun 16</b>	<b>15</b>
<b>10 Junn 22</b>	<b>15</b>
<b>11 June 28: Review of ICA</b>	<b>15</b>
<b>12 Aug 19 fMRI clustering review</b>	<b>15</b>

# 1 Ising Model

In this section I did test on Ising model. Following example 1 in [Perez \[1998\]](#), I sampled the prior the distribution of hidden variable  $\mathbf{x}$  on a  $255 \times 255$  binary images. Results at fig. 1.

Also apply Ising prior on a image denoising application: 1) Get a original image and apply 10% noise. 2) denoise with no spatial prior, with little and much prior. Results at fig. 2.

**Figure 1:** Initial image with uniform random -1 an +1. Update each pixel in sequence by the conditional probability. And repeat this many 1000 times. Here is the iteration for 1st time, 300, 600 and 900.

**Figure 2:**  $\beta$  controls how strong the prior is, compared to the conditional likelihood  $P(\mathbf{y}|\mathbf{x})$ . Greater  $\beta$  means the energy function prefer prior rather than data.

## 1.1 Generate Ising model by Gibbs Sampling

In this test I want to generate a Ising model using Gibbs sampling. The possible gray level for each data point is -1, 1 and I'm going to use Gibbs sampling method to generate the prior image in Gibbs distribution.

The Gibbs distribution is defined by

$$P(\mathbf{x}) = \frac{1}{Z} \exp\{-U(x)\} \quad (1)$$

The conditional probability of data point  $x_s$  is given by

$$p(x_s|x_{i \neq s}) = \frac{\frac{1}{Z} \exp\{-\sum_{C \in \mathcal{C}} V_C(\mathbf{x}_C)\}}{\sum_{x_s=0}^{M-1} \frac{1}{Z} \exp\{-\sum_{C \in \mathcal{C}} V_C(\mathbf{x}_C)\}} \quad (2)$$

$$= \frac{\frac{1}{Z} \exp\{-\sum_{C:s \in C} V_C(\mathbf{x}_C)\}}{\sum_{x_s=0}^{M-1} \frac{1}{Z} \exp\{-\sum_{C:s \in C} V_C(\mathbf{x}_C)\}} \quad (3)$$

$$= \frac{\exp\{-\sum_{C:s \in C} V_C(\mathbf{x}_C)\}}{\sum_{x_s=0}^{M-1} \exp\{-\sum_{C:s \in C} V_C(\mathbf{x}_C)\}} \quad (4)$$

(5)

So we only need to compute the potential function for those  $V_C$  that include  $x_s$ . Define  $V_C$  as

$$V_C(x_i, x_j) = \begin{cases} 1 & \text{when } x_i \neq x_j \\ 0 & \text{when } x_i = x_j \end{cases} \quad (6)$$

**Figure 3:** Gibbs sampling to generate Gibbs distribution. Upper-left is the random generated original 255x255 image with five gray level intensity. Upper-right is after 300 iterations of Gibbs sampling, with each iteration having raster scan through each pixel. Lower images are iterations 400 and 600 times.

**Figure 4:** Generate N time series. Some are sin wave + Gaussian noise, others are purely Gaussian noise. Visualize them in a single plot by shifting the mean of each time series according to its label (from 1 to N)

We don not need to have bigger potential function value for those  $|x_i - x_j|$  is large, because  $x_i$  and  $x_j$  as labeling, should make no difference on potential as long as they are different.

The difference between Gibbs sampling and Metropolis Sampling is, Gibbs sampling need to compute conditional probability of  $p(x_s|x_{i \neq s})$ , which take much computation when the possible value of  $x_s$  is large. For binary image like Ising model, possible value of  $x_s$  is just two, and Gibbs Sampling has no problem computing that.

Figure 3 is the simulation result

Question: When the possible gray level is more than two, how do I generate them in Gibbs distribution (suppose I can compute the probability of each possible value, i.e.  $P(\mathbf{x} = x_0)$ ,  $P(\mathbf{x} = x_1)$ , ... (Solved))

## 1.2 Jan 19 Generate 1D fMRI image and compute posterior distribution

Two questions about Fisher transformation: 1) When data point  $x$  and  $y$  are not independent, is  $z = 0.5 * \ln(\rho + 1/\rho - 1)$  still approximately Gaussian distribution? 2) No matter if data point  $x$  and  $y$  are independent, to assume  $z$  is approximately Gaussian, we must make sure sample correlation  $r_i$  are independent over all  $i = 1, \dots, N$ . But in MRF, connectivity  $c_i$  is not independent, i.e.  $P(c_i|c_{j \neq i} \neq p(c_i))$ . Does this mean the  $p(r|c)$  is not independent?

I think even  $c$  is not independent,  $r_i$  is still independent given  $c_i$ , by the theory of conditional independence. That is,  $p(r_1, r_2|c_1, c_2) = p(r_1|c_1) \cdot p(r_2|c_2)$ .

**Figure 5:** The empirical distribution of the correlation between pair of  $x_i$  and  $x_j$  given they are 'connected'. To make them connected I generate all  $x$  from same sin wave plus independent Gaussian noise with different variance.  $A$  is amplitude of sin wave, and  $\sigma$  is standard deviation of the Gaussian noise. I define  $A^2/\sigma^2$  as the signal-to-noise ratio. On the bottom row is the empirical distribution of the  $z$  after Fisher Transformation. From Wikipedia, Fisher transformation  $z$  is approximately normal only when 1)  $x_i$  and  $x_j$  are normal distribution, and 2)  $x_i$  and  $x_j$  are independent. Both conditions are not true in our case. But I just do the transformation and see the  $z$ 's distribution.

Now to compute posterior probability of connectivity  $c$ , I can use Gibbs Sampling(or Simulated Annealing) and EM algorithm to iteratively compute the posterior probability and model parameter. Here is the notations:

- $S$  is the set of lattice points.
- $s$  is a lattice point,  $s \in S$ .
- $X_s$  is the value of  $X$  at  $s$ . In our case,  $X_s$  is the connectivity between two voxels.  $X_s \in \{-1, 1\}$ . It is also the latent variable we're interested in.
- $\partial s$  is neighboring points of  $s$ .
- $x_c$  is the value of  $X$  at the points in clique  $c$ .
- $V_c(x_c)$  is potential function.
- $Y_s$  is correlation at site  $s$ .  $y_s$  is sample correlation between two voxels.

We do not know the likelihood. There are two options: 1) Assume a data model used to generate  $Y$ . For example, we can assume two data point  $\tilde{d}_i$  and  $\tilde{d}_j$  and assume they are clear signal without any noise. Further assume they are perfectly correlated. Then generate two noised signal  $d_i = \tilde{d}_i + N_i$  and  $d_j = \tilde{d}_j + N_j$ .  $N_i$  and  $N_j$  are additive Gaussian noise term with zero mean. Then we try to compute probability  $\text{corr}(d_i, d_j)$  given the fact  $\tilde{d}_i$  and  $\tilde{d}_j$  are perfectly correlated. 2) We can generate some sample correlation  $y_s$  from data points  $d_i$  and  $d_j$  given  $\tilde{d}_i$  and  $\tilde{d}_j$  have correlation one. This can be seen as Monte Carlo method. From figure 5 we see even  $d_i$  and  $d_j$  are correlated, their sample correlations are approximately Gaussian distribution after Fisher Transformation.

So I assume the likelihood function  $p(y_s|x_s = 1)$  is Gaussian with unknown  $\mu_1$  and  $\sigma_1^2$ . We already know  $p(y_s|x_s = 0)$  is Gaussian with known  $\mu_0 = 0$ . To compute both the posterior  $p(X|Y)$  and the parameters  $\mu_1$  and  $\sigma_1^2$  and  $\sigma_0^2$  try to use EM algorithm as below:

**Issue 1:** If there is negative correlation in the sample correlation, we need to model this component. We need to look at the histogram of sample correlation on real data, and see if there is much negative correlation. If there is, we need to add another state for connectivity, i.e.  $c_{ij} = -1$  to model this component.

**Issue 2:** Need to assume the latent variable 'connectivity' is continuous in  $[0, 1]$ , instead of the discrete  $\{0, +1\}$  for current model.

**Issue 3:** How to change the temperature  $T$  in annealing of E step? Two options 1) Keep  $T$  unchanged in each iteration of EM. That is,  $T$  is constant over the annealing. Over all E step,  $T$  decreases. 2)  $T$  decrease over annealing in a single E step of EM. At the beginning of annealing of each E step,  $T$  begins with a high value.

**Issue 4:** In terms of 'neighbors', we can define two types of neighbors: 'local neighbors' and 'remote neighbors'. Local neighbors are those data points spatially close to the data we're interested in. Remote neighbors are data that are not necessarily spatially close, but are

---

**Algorithm 1** EM-Annealing

---

**Require:** Sample correlation matrix  $\mathbf{Y}$  with  $y_s$  the sample correlation between voxel  $i$  and  $j$ .

Init posterior matrix  $\mathbf{X} : x_s = \text{argmax} \ln p(Y_s|x_s; \theta)$ .

**while** Some Condition **do**

**E step:**

(1) Based on the current parameters  $\boldsymbol{\theta} = \{\mu_0, \mu_1, \sigma_0^2, \sigma_1^2, \beta\}$ , compute the posterior probability as

$$p(X_s|Y_s = y_s) = \frac{p(X_s) \cdot p(Y_s = y_s|X_s, \boldsymbol{\theta})}{p(Y_s = y_s|\boldsymbol{\theta})} \quad (7)$$

(2) Repeatedly Do Gibbs Sampling from (7) until the field stabilize.

(3) Based on current value of  $X_s$ , iteratively compute the mean field

$$p(X_s| < X_{\partial s} >) = \frac{1}{Z_s} \exp\{-\beta U_s(X_s| < X_{\partial s} >)\} \quad (8)$$

$$Z_s = \sum_{X_s \in \{-1, 1\}} \exp\{-\beta U_s(X_s| < X_{\partial s} >)\} \quad (9)$$

$$< X_s > = \sum_{X_s \in \{-1, 1\}} X_s \cdot p(X_s| < X_{\partial s} >) \quad (10)$$

**M step:**

(4) With compute data  $\{X, Y\}$ , estimate  $\beta$  by maximizing log-likelihood of posterior probability of  $X$ . Because likelihood  $p(Y|X)$  does not depend on  $\beta$ , we only maximize prior  $p(X)$ , which is Gibbs distribution. We use Newton's method. (To be added)

(5) Estimate  $\mu$  and  $\sigma^2$  by maximizing conditional log-likelihood  $p(Y| < X >)$ . (To be added)

**end while**

---

close according to some defined features (like DTI connection). And we can use Suyash's theory: to know current pixel, we not only look at its local neighbors (like MRF), and also look at its remote neighbors.

**issue 5:** When applying prior, we have to be careful not to smooth 'edges'. When we find edges, we should not assume the pixels on two sides of edges are neighbors (either local neighbors, or remote neighbors). the edges are not given, but to be computed in the iteration algorithm. (need to study this later)

**Issue 5:** If we can assume the posterior distribution is also a Gibbs distribution, we can use annealing on posterior Gibbs, instead of only on prior. In this method, there is an additional parameter  $\alpha$  to control the weights between the prior energy and likelihood energy (update: From Gelman's paper, seems there is no such  $\alpha$ , posterior energy just sum of prior and likelihood energy, not weighted sum.)

To prove the posterior distribution  $p(c_n|r_n)$  is also Gibbs distribution, we just need to prove it is Markov Random Field. Assume  $\partial c$  is the neighbors of  $c_n$ , and  $\tilde{C}$  is the set of other nodes that does not include  $c_n$  and  $\partial c$ . We need to prove

$$p(c_n|r_n, \partial c, \tilde{C}) = p(c_n|r_n, \partial c) \quad (11)$$

while we know that

$$p(c) \sim \text{Gibbs}, \quad p(c_n|\partial c_n, \tilde{C}) = p(c_n|\partial c_n) \quad (12)$$

$$p(r_n|c_n) = \mathcal{N}(\mu, \sigma^2) \quad (13)$$

$$(14)$$

Now rewrite the posterior as

$$p(c_n|r_n, \partial c_n, \tilde{C}) = \frac{p(c_n) \cdot p(r_n, \partial c_n, \tilde{C}|c_n)}{p(r_n, \partial c, \tilde{C})} \quad (15)$$

$$= \frac{p(c_n) \cdot p(r_n|c_n) \cdot p(\partial c_n, \tilde{C}|c_n)}{p(r_n) \cdot p(\partial c_n, \tilde{C})} \quad (16)$$

$$= \frac{p(r_n|c_n)}{p(r_n)} \cdot p(c_n|\partial c_n, \tilde{C}) \quad (17)$$

$$= \frac{p(r_n|c_n)}{p(r_n)} \cdot p(c_n|\partial c_n) \quad (18)$$

$$p(c_n|\partial c_n) = \frac{p(c_n)p(r_n, \partial c_n|c_n)}{p(r_n, \partial c_n)} \quad (19)$$

$$= \frac{p(c_n) \cdot p(\partial c_n|c_n) \cdot p(r_n|c_n)}{p(r_n) \cdot p(\partial c_n)} \quad (20)$$

$$= \frac{p(c_n, \partial c_n) \cdot p(r_n|c_n)}{p(r_n) \cdot p(\partial c_n)} \quad (21)$$

We see (18) and (21) are equal, so we prove the posterior is also Gibbs.

**SNR of functional MRI data:** It is better to view fMRI data as time series and define signal-to-noise ratio from signal processing standpoint, i.e.

$$SNR = \frac{P_{sig}}{P_{no}} = \left( \frac{A_{sig}}{A_{No}} \right)^2$$

According to this definition, the 'auditory' data set has SNR closet to 1.0. This is better than  $SNR = A_{sig}/\sigma_{no}^2$ . If  $SNR = A_{sig}/\sigma_{no}^2 = 1$ , we have at least one time point among 100 time point that can reach  $3\sigma = 3$ , and by signal processing definition, the SNR would be 1/3.

## 2 New Reading on Functional Connectivity

### 3 Apr 6 Meeting

- Add one clique potential energy function.
- Add edge process to preserved edges when annealing.
- When estimating parameters, again we can use simulated annealing (or Metroplis-Hasting) to find the global maximum of the psudo-likelihood with respect to parameters.
- Change 26 neighbors back to 12 neighbors. –

### 4 Apr 20

**Prerocessing:** The processing takes following steps:

- Slice timing correction. After this step, I get image `rawfmriS`.
- motion correction with middle volume as reference. After this step I get motion corrected image `rawfmriSC`.
- Remove skull and scalp of t1, t2 and middle volume of fmri image by fsl tool `bet`, and get results in `t1B` and `t2B`.
- Register scalp-removed `t2B` to `t1B`, and get `t21`.
- Segmentation of `t1B` using FSL tool `fast`, get gray matter mask image `t1B_pve_1`.
- Register fmri (scalp removed middle volume `midB`) to `t21`, and get transformation matrix `mid_t2B.mat`.
- Invere the matrix `mid_t2B.mat`, and get trans. matrix `t2B_mid`.

- Use the intervse matrix in previous to register gray matter mask image to middle fmri image, and get `mask`.
- Now register t1B (scalp removed t1 image) to MNI altas, only to get the transformation matrix.
- concatenate two transforms. `fmri` → `t21`, and `t1` → `MNI`. The goal is to get trasformation of `fmri` → `MNI`.
- Given the coordinates in MNI space and the transform matrix in previous step, use FSL tool `std2imgcoord` to find the coordinates(voxel) in fmri image. The reason of doing this step is, for fMRI image of different subject, we do not have to mannually pick CSF or white matter voxel. We just choose a coordinate of CSF/WM in standard MNI space, and the script automatically find the voxels in fMRI space.

After above steps, the `mask` image looks better than what I got from `t2` image. one thing nee mention is when removing scalp and skull from `t2`, the results is not good. But since we only use scalp remove `t2` for registration between `t1` and `fMRI`, this does not matter much.

**Number of Voxels in gray matter:** When thresholding `t1` mask image at 0.5, the number of voxels in gray matter is 15,325. Set threshold at 0.4 give the number 18,654.

## 5 GPUSampling code change memo

$$\begin{aligned}
V_c(x_i, x_j) &= -\beta x_i x_j \\
U &= \sum_C V_c \\
p(\mathbf{x}) &= \frac{1}{Z} \exp\{-U(\mathbf{x})\} \\
p(x_i) &= \frac{1}{Z_i} \exp\{\sum \beta x_i x_j\} = \frac{1}{Z_i} \exp\{\beta \sum x_i x_j\} \\
p(x_i|\cdot) &= \frac{\exp\{\beta x_i \sum_j x_j\}}{2 \cosh(\beta \sum_j x_j)} \\
p(x_i|y_i) &\propto p(x_i|\cdot) \cdot p(y_i|x_i) \\
p(x_i|y_i) &= \frac{\exp\{\beta x_i \sum_j x_j\} \cdot \frac{1}{\sigma_i} \exp\{-\frac{(y_i - \mu_i)^2}{2\sigma_i^2}\}}{\sum_{x_i=\{-1,1\}} \exp\{\beta x_i \sum_j x_j\} \cdot \frac{1}{\sigma_i} \exp\{-\frac{(y_i - \mu_i)^2}{2\sigma_i^2}\}} \\
&= \frac{\exp\{\beta x_i \sum_j x_j - \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log \sigma_i\}}{\sum_{x_i \in \{-1,1\}} \exp\{\beta x_i \sum_j x_j - \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log \sigma_i\}}
\end{aligned}$$

Define the posterior energy as

$$\begin{aligned}
U^p &= -\beta x_i \sum_j x_j + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} + \log \sigma_i \\
p(x_i|y_i) &= \frac{\exp\{-U^p\}}{\sum_{x_i \in \{-1,1\}} \exp\{-U^p\}} \\
p(x_i = -1|y_i) &= \frac{\exp\{-U_{-1}^p\}}{\exp\{-U_{-1}^p\} + \exp\{-U_1^p\}} \\
&= \frac{1}{1 + \exp\{U_{-1}^p - U_1^p\}} \\
\text{if } U_{-1}^p &>> U_1^p \quad p(x_i = -1|y_i) \approx 0 \\
\text{if } U_{-1}^p &<< U_1^p \quad p(x_i = -1|y_i) \approx 1 \\
\text{otherwise} \quad p(x_i = -1|y_i) &= \frac{1}{1 + \exp\{U_{-1}^p - U_1^p\}}
\end{aligned}$$

**Likelihood and its derivative:** Without adding single clique potential term, let us review the derivative. because

$$p(x_i|\cdot) = \frac{\exp\{\beta x_i \sum_j x_j\}}{2 \cosh(\beta \sum_j x_j)}$$

We compute the log-likelihood of single pixel given other points as

$$\ln(x_i; \beta) = \beta x_i \sum_j x_j - \ln\left(2 \cosh(\beta \sum_j x_j)\right)$$

The 1st and 2nd derivative of the log-likelihood is given as

$$\begin{aligned}
\frac{\partial \ln p(x_i; \beta)}{\partial \beta} &= x_i \sum_j x_j - \frac{2 \sinh(\beta \sum_j x_j)}{2 \cosh(\beta \sum_j x_j)} = \left(x_i - \tanh(\beta \sum_j x_j)\right) \sum_j x_j \\
\frac{\partial^2 \ln p(x_i|\cdot)}{\partial \beta^2} &= \left(\sum_j x_j\right)^2 \left(\tanh^2(\beta \sum_j x_j) - 1\right)
\end{aligned}$$

Now if we add  $\alpha$  term we have likelihood and its log as

$$\begin{aligned}
p(x_i|\cdot) &= \frac{\exp\{\alpha x_i + \beta x_i \sum_j x_j\}}{\dots} \\
&= \frac{\exp\{x_i(\alpha + \beta \sum_j x_j)\}}{2 \cosh\{\alpha + \beta \sum_j x_j\}} \\
\ln p(x_i|\cdot) &= x_i(\alpha + \beta \sum_j x_j) - \ln\left(2 \cosh(\alpha + \beta \sum_j x_j)\right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ln p(x_i|\cdot)}{\partial \alpha} &= x_i - \tanh(\alpha + \beta \sum_j x_j) \\
\frac{\partial^2 \ln p(x_i|\cdot)}{\partial \alpha^2} &= \tanh^2 \left( \alpha + \beta \sum_j x_j \right) - 1 \\
\frac{\partial \ln p(x_i|\cdot)}{\partial \beta} &= \beta x_i \sum_j x_j - \tanh(\alpha + \beta \sum_j x_j) \cdot \sum_j x_j = \sum_j x_j \left( x_i - \tanh(\alpha + \beta \sum_j x_j) \right) \\
\frac{\partial^2 \ln p(x_i|\cdot)}{\partial \beta^2} &= \left( \sum_j x_j \right)^2 \left( \tanh^2(\alpha + \beta \sum_j x_j) - 1 \right)
\end{aligned}$$

Now the posterior energy looks like

$$\begin{aligned}
U^p &= -\alpha x_i - \beta x_i \sum_j x_j + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} + \log \sigma_i \\
p(x_i|y_i) &= \frac{\exp\{-U^p\}}{\sum_{x_i \in \{-1,1\}} \exp\{-U^p\}}
\end{aligned}$$

**Joint log-likelihood:** To tell when EM stop iteration, we need to compute joint log-likelihood  $p(\mathbf{X}, \mathbf{Y})$ , where  $\mathbf{X}$  is the set of hidden variable (label) on all sites, and  $\mathbf{Y}$  is the observed sample correlation.

$$P(\mathbf{X}, \mathbf{Y}) = p(\mathbf{X}) \cdot p((\mathbf{Y}|\mathbf{X}) = \prod_i p(x_i|\cdot) \cdot \prod_i p(y_i|x_i) \quad (22)$$

$$\log p(\mathbf{X}, \mathbf{Y}) = \sum_i \log(p(x_i|\cdot)) + \sum_i \log p(y_i|x_i) \quad (23)$$

$$p(x_i|\cdot) = \frac{\exp\{\beta x_i \sum_j x_j\}}{2 \cosh(\beta \sum_j x_j)} \quad (24)$$

$$\log(x_i; \beta) = \beta x_i \sum_j x_j - \log \left( 2 \cosh(\beta \sum_j x_j) \right) \quad (25)$$

$$p(y_i|x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left\{ \frac{(y_i - \mu_i)^2}{2\sigma_i^2} \right\} \quad (26)$$

$$\log p(y_i|x_i) = \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log \sigma_i - \log(\sqrt{2\pi}) \quad (27)$$

$$\begin{aligned}
\log p(\mathbf{X}, \mathbf{Y}) &= \sum_i \left( \beta x_i \sum_j x_j - \log \left( 2 \cosh(\beta \sum_j x_j) \right) \right) + \sum_i \left( \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log \sigma_i - \log(\sqrt{2\pi}) \right) \\
&\quad (28)
\end{aligned}$$

$$= \sum_i \left( \beta x_i \sum_j x_j - \log \left( 2 \cosh(\beta \sum_j x_j) \right) + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log \sigma_i - \log(\sqrt{2\pi}) \right) \quad (29)$$

(30)

To compute the joint log-likelihood of (29) we need to know the latent variable, i.e. the labeling variable  $\mathbf{X}$ . But since we only know the posterior distribution of  $x_i$ , we can use the expected value of  $\mathbf{X}$  in computing (29) and compute the expectation of (29) with regard to  $p(\mathbf{X}|\mathbf{Y})$ .

$$\begin{aligned} \mathcal{Q}(\theta) &= \mathbb{E}_{p(\mathbf{X}|\mathbf{Y})} [\log p(\mathbf{X}, \mathbf{Y})] \\ &= E_{p(\mathbf{X}|\mathbf{Y})} \left[ \sum_i \left( \beta x_i \sum_j x_j - \log \left( 2 \cosh(\beta \sum_j x_j) \right) + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log \sigma_i - \log(\sqrt{2\pi}) \right) \right] \\ &= \sum_i E_{p(\mathbf{X}|\mathbf{Y})} \left[ \left( \beta x_i \sum_j x_j - \log \left( 2 \cosh(\beta \sum_j x_j) \right) + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log \sigma_i - \log(\sqrt{2\pi}) \right) \right] \\ &= \sum_i p(x_i = 1|y_i) \cdot \left( \beta \sum_j x_j - \log \left( 2 \cosh(\beta \sum_j x_j) \right) + \frac{(y_i - \mu_1)^2}{2\sigma_i^2} - \log \sigma_i - \log(\sqrt{2\pi}) \right) + \\ &\quad p(x_i = -1|y_i) \cdot \left( -\beta \sum_j x_j - \log \left( 2 \cosh(\beta \sum_j x_j) \right) + \frac{(y_i - \mu_{-1})^2}{2\sigma_i^2} - \log \sigma_i - \log(\sqrt{2\pi}) \right) \end{aligned}$$

**Choose Seed voxel in std space:** To guarantee we choose same seed region in all data set, we need to define the seed coordinates in standard MNI space. Because in previous section, we get the registration of fMRI  $\rightarrow$  t2  $\rightarrow$  t1  $\rightarrow$  MNI space, we can inverse transform the coordinates in MNI space to the image space (R1, R2 ...).

**Linear indexing in a triangular matrix:** The original  $N$  square matrix  $C$  and  $R$  are used for saving connectivity and correlation value. If we keep only the upper triangular part, we can reduce the size to  $N(N + 1)/2$  (including the diagonal). The mapping from  $(i, j)$  coordinates to linear index (row major) would be:

$$M(i, j) \rightarrow \frac{(2N - i + 1)i}{2} + (j - i)$$

That is, the  $i$ th and  $j$ th column in matrix  $M$  is located at  $\frac{(2N - i + 1)i}{2} + (j - i)$  in memory.

**Compute posterior probability from expected value:** We can save much memory space by computing posterior probability  $p(x_i = 1|y)$  from expected value  $E(p(x_i))$ . This is because

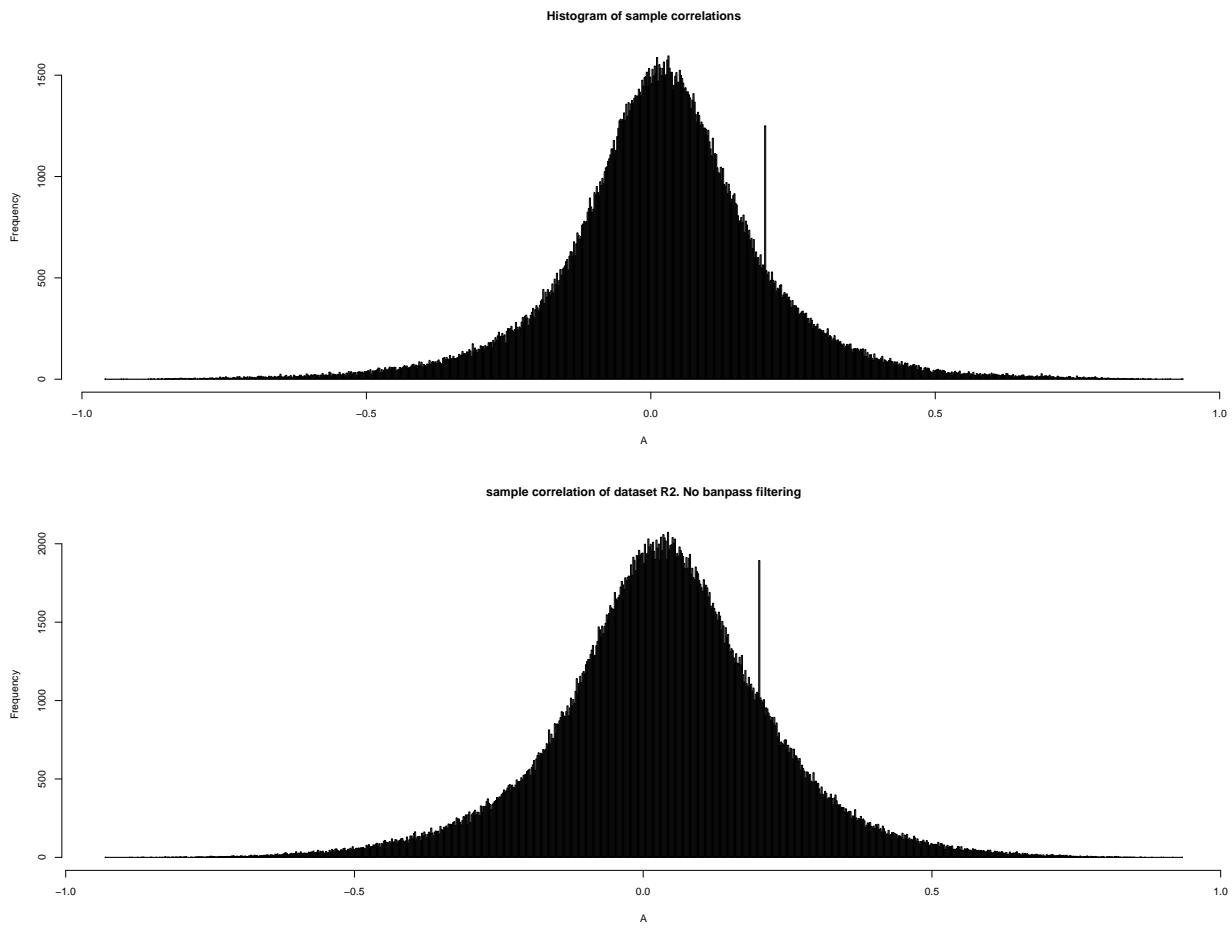
$$\begin{aligned} p(x_i = 1) + p(x_i = -1) &= 1 \\ p(x_i = 1) \cdot 1 + p(x_i = -1) \cdot (-1) &= c_i \end{aligned}$$

So we can solve them and get

$$p(x_i = 1) = \frac{1 + c_i}{2} \quad p(x_i = -1) = \frac{1 - c_i}{2}$$

## 6 may 18

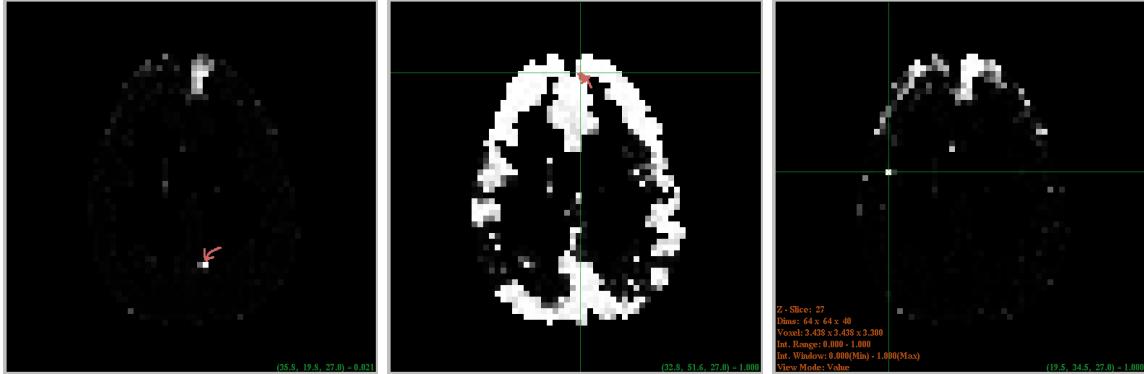
**Histogram of sample correlation of the fMRI data without filtering:** See figure 6.



**Figure 6:** histogram of sample correlation of fMRI dataset R1 and R2. top is R1. Bottom is R2. It is unimodal, with mean slightly away from zero. The sharp bump around 0.2 is the connection of a voxel with itself, and I set it manually to 0.2, a empirical value that falls into the mixture model.

**connectivity map with different seed voxels:** See figure 7.

**bandpass filtering:** our data set  $TR = 2s$ . In 'functional connectivity premier', the cutoff frequency  $f_H = 0.08Hz$ ,  $f_L = 0.009Hz$ . for low pass filtering,  $0.08Hz \rightarrow 12.5s \approx 6TR$ . For high pass filtering,  $0.009Hz \rightarrow 111s \approx 55TR$ .



**Figure 7:** connectivity map with different seed voxels.

**estimation of parameters  $\mu$  and  $\sigma$ :** See log files.

**Histogram of sample correlation after bandpass filtering:** See figure 8. There are more positive correlated and negative correlated connectivity than before filtering. To see if this change of histogram is caused by high pass filtering, or low pass filtering. I need to filter time series by either of them and see the results. A simple test found even only with low pass filter, the histogram of filtered data is much like bandpass filtered data. That mean removing high frequency component of time series significantly changed the histogram of correlation.

**Time and frequency domain response before and after bandpass filtering:** See figure 9.

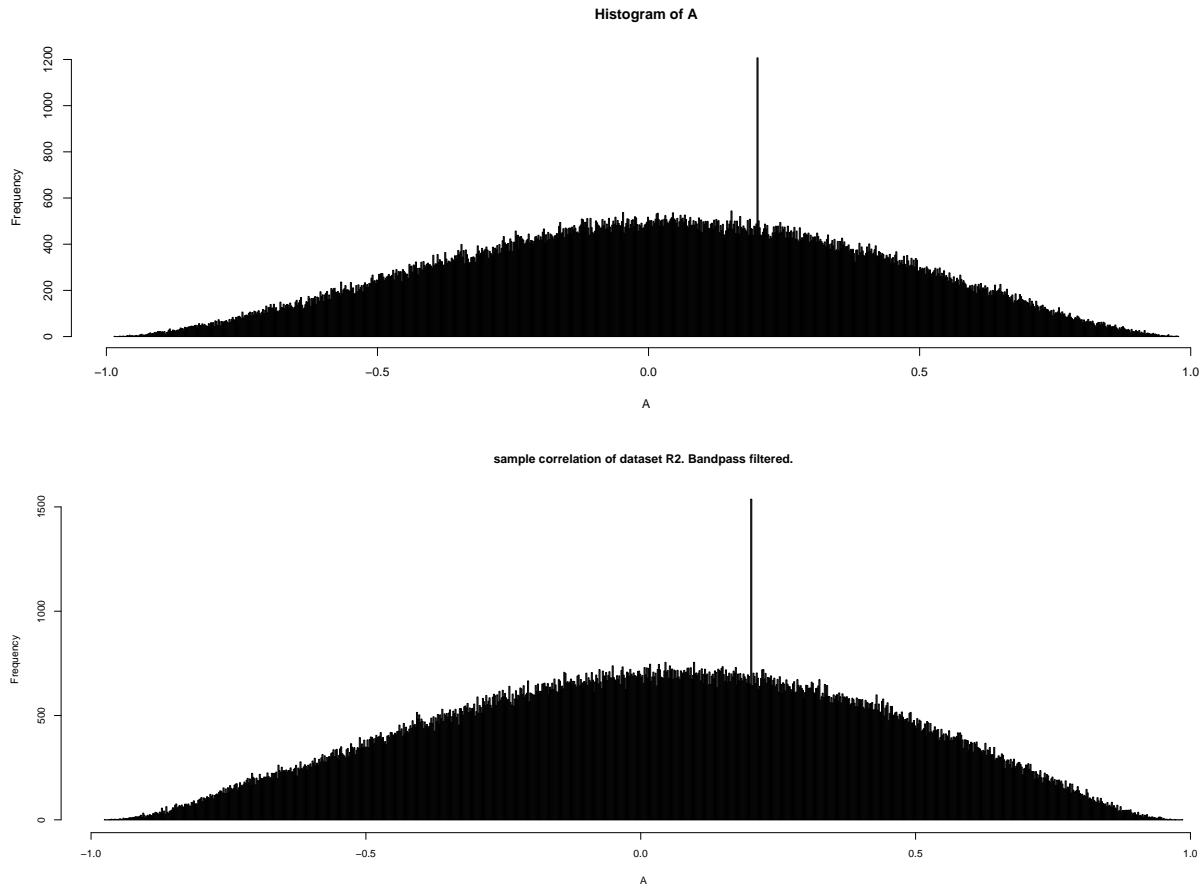
**corrlation map before and after bandpass filtering;** See figure 10.

## 7 May 25

**Standard SPM method:** In Greicius paper, a ROI is selected as a covariate (regressor), and linear regression (SPM) method is used to get the contrast map of whole brain. The linear regression parameter  $\hat{\beta}$  is not equivalent to the sample correlation between regressor time sequence and other time sequence. To see that sample correlation  $r_{xy} = \sum_i x_i y_i / (N - 1)s_x s_y$ . But in linear regression  $\mathbf{Y} = \mathbf{X}\beta + \epsilon$ , the estimator of  $\beta$  is  $\hat{\beta} = \sum_i x_i y_i / \sum_i x_i^2$ , and we see  $\hat{\beta} = (s_y/s_x)r_{xy}$ . That means, even if two voxels have small correlation, the linear regression parameter  $\beta$  may be large.

When we divide  $x$  and  $y$  by their standard deviation  $s_x$  and  $s_y$ ,  $\hat{\beta}$  will be equal to sample correlation  $r_{xy}$ . So, linear regression in this case is amount to compute sample correlation btween two data points on a sphere.

- Preprocessing: motion correction by SPM.
- Spatial smoothing by SPM.



**Figure 8:** Histogram of sample correlation after bandpass filtering. Top is dataset R1, bottom is dataset R2.

- Use `fslmaths` to convert to float type. `fslmaths srfmri.nii -add 0 srfmri_f.nii -odt float`
- Temporal filtering by FSL. `fslmaths srfmri -bptf 55 6 fsrfmri`
- Choose seed voxel, get regressor by averaging around seed voxel.
- Linear regression, and estimate coefficients  $\beta$  for the whole volume.
- Also compute correlation between seed regressor and all other voxels.

## 8 Jun 9

Use `conn` for functional connectivity analysis. Processing: motion correction, slice timing, registration with T1 and T2, normalization, spatial smoothing. I found (1). Adding motion parameters as confounds improve the results a little? (2) more PCA components and derivatives of covariates improves results. (3) Spatial smoothing makes a big difference.

## 9 Jun 16

## 10 Junn 22

By tuning  $\alpha$  term in MRF's clique potential function, we can choose the relative proportion between the 'connected' cluster and 'un-connected' cluster. The  $\alpha$  term is similar to the  $p$  value in standard SPM method.

By tuning  $\beta$  term in MRF's clique potential function we can choose how the connectivity map is partitioned into (i.e. big chunks or small chunks). The  $\beta$  term is similar to the spatial smoothing kernel size in SPM.

So what's the sense of our MRF since what  $\alpha$  and  $\beta$  can do are also can be done by spatial smoothing. The answer is MRF give a concept of neighborhood. And by re-defining neighborhood, we can incorporate inter-subject information, or multi-modality information (DTI) into this neighborhood system.

**Summary of talk with Tom:** For group analysis, we need  $p(X|D_1, D_2, \dots) = p(D_1, \dots, |X)$ .  
 $p(X) = p(D_1|X) \cdot p(D_2|X) \cdot p(D_3|X)$ .

## 11 June 28: Review of ICA

for Gaussian distribution, uncorrelatedneww and independence become the same thing.

For PCA, we can derive the projection matrix without knowing the data distribution. PGA does not use data distribution either? But for ICA, one of the two assumptions is non-Gaussian, which is related to the data distribution. How to add this information in Riemannian Geometry?

Consider probabilistic PCA. If we can generalize PPCA into Riemannian space, we may also generalize ICA. This is because PPCA has distribution information in it, and is more similar to PCA.

## 12 Aug 19 fMRI clustering review

abcd

## 13 Scratch Pad

spontaneous brain activity. Spontaneous neuronal activity consume 20% of total energy while task related only take 5%.

One strategy to account for these non-neuronal noise is high sampling rate, which prevents aliasing of higher frequency of cardiac or respiratory activity. But this reduce spatial coverage.

Regions that have similar functionality in task paradigms, tends to be correlated in spontaneous activity.

Resting-state activity serves as a priori hypothesis that brain response to task conditions.

Temporal properties:  $1/f$ . Only frequency below 0.1 Hz is good. (do we need a low pass filter?)

The spontaneous activity can increase SNR of task by correcting. Maybe it explains many inter-trail variability.

The decrease/deactivation arise from unrecognized increases/activations? No enough information of the control state to judge this.

real data - $\zeta$  *in vivo* data

red blood cell carry hemoglobin (xuehongsu), which store oxygen molecule. hemoglobin can release and absorb oxygen. At lungs, red blood contain hemoglobin, which is oxygenated, then deoxygenated. Oxygenated and de-oxygenated hemoglobin have different magnetic properties:

If we just use band-pass filter before calculating correlation, is there any difference with frequency method, i.e., spectral coherence averaging at different frequency point.

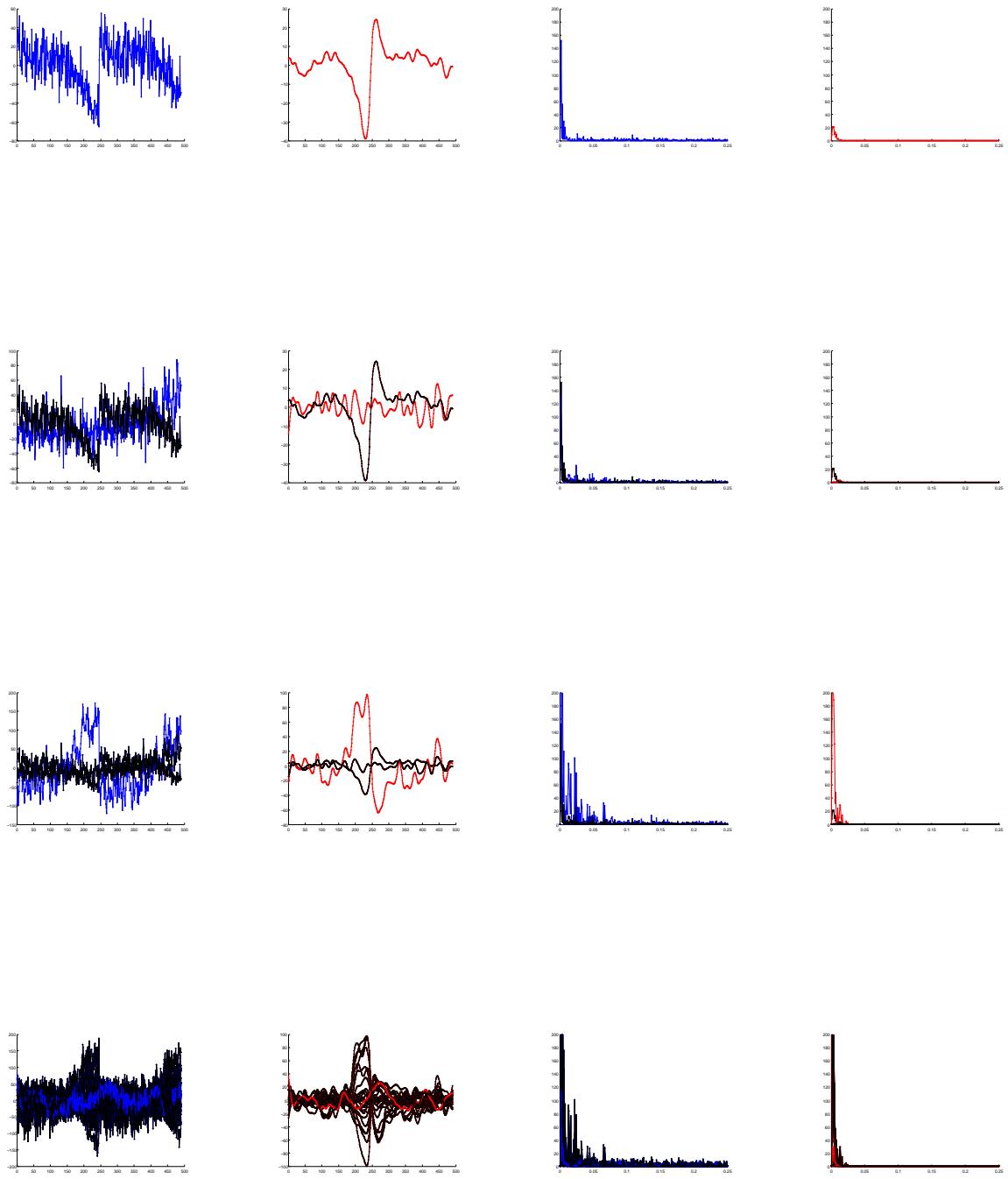
Is there any need for slice-timing correction. In SPM, they do not recommend to do it because there is probably a regressor(1st derivative of HRF) that can take care of slice timing issue. But for resting state connectivity we do not have such feature. Furthermore, for inter-leaved EPI sequence, there is even more need for this.(mindhive.mit.edu/node/109)

Global normalization. Remove the difference between volumes in different time point??? search spm manual for 'global normalization'.

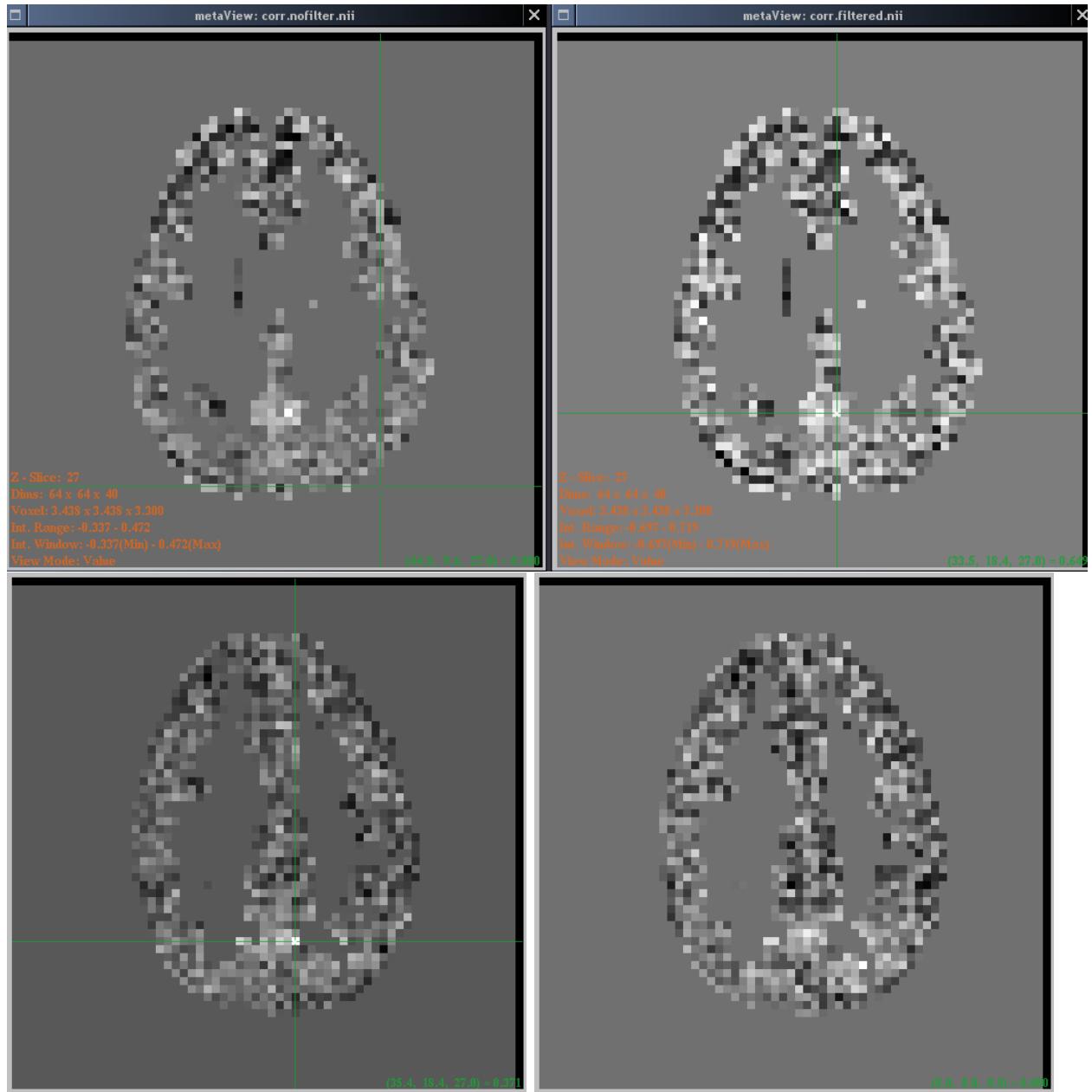
3/25: Found high pass filter make the detection worse. (high pass filter sigma = 20 TR). Also found low pass filter make detection worse, though it's slightly better than applying high pass filter. (low pass sigma = 2 TR).

## References

P. Perez. Markov random fields and images. *CWI Quarterly*, 11(4):413437, 1998.



**Figure 9:** Time sequence before and after bandpass filtering. each row is a time course.



**Figure 10:** correlation map. Left is before bandpass filtering, and right is after bandpass filtering. First row is dataset R1, second row is dataset R2. Seed voxel is [34 28 27].

