

# Discovering signals in fMRI data; a Bayesian nonparametric approach

Ahmed Bou-Rabee, Wanrong Zhu, Zheng Xu, Mo Zhou

STAT 30850  
University of Chicago

March 16, 2017

# Outline

Introduction

Method

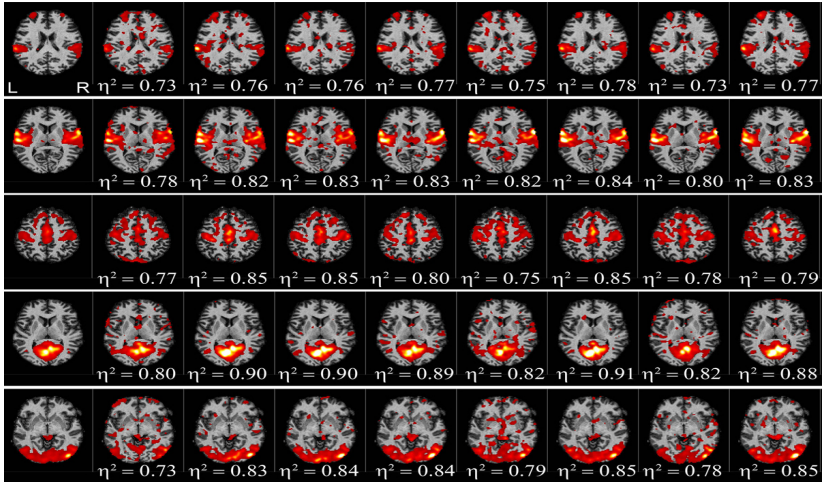
Example: simulated data

Example: fMRI data

# What is fMRI?

- ▶ fMRI: functional magnetic resonance imaging
- ▶ measures the change in brain blood flow while subject is stimulated
- ▶ data tends to look like (voxel, time, intensity of reading)
- ▶ e.g., to identify regions of the brain associated with hunger, fMRI readings can be taken while hungry subjects are shown pictures of food.

## Example



# Problems

- ▶ multiple comparison problem - thousands of readings over time and space
- ▶ want to incorporate location and time information - want continuous significant regions, not voxels
- ▶ want to determine these significant regions adaptively

# Outline

Introduction

Method

Example: simulated data

Example: fMRI data

# The idea

- ▶ focus on data with p-values rather than intensities
- ▶ assume that data is generated according to a prior which induces clustering of significant p-values
- ▶ get the posterior distribution (somehow)
- ▶ p-values which are in significant clusters are discoveries

## The prior we assumed

- ▶ number of signal clusters:  $k \sim \text{Truncated Poisson}(\lambda, 1, k_{max})$
- ▶ signal centers:  $c_j \sim \text{Uniform}$  over location-time space for  $j = 1, \dots, k$
- ▶ signal radius:  
 $r_j \sim \text{Truncated Normal}(\mu, \sigma, r_{min}, r_{max})$  for  $j = 1, \dots, k$ .
- ▶ signal strength:  $\beta_j \sim \text{Uniform}(\beta_{min}, \beta_{max})$  for  $j = 1, \dots, k$ .
- ▶ p-values in signal clusters:  $p_i \sim \text{Beta}(\frac{1}{\beta_j}, \beta_j)$ , when  $x_i$  is in cluster  $j$ .
- ▶ p-values not in signal clusters:  $p_i \sim \text{Uniform}(0, 1)$ .



## How do we write down the posterior??

- ▶ we don't - we sample from it
- ▶ we invented a Markov chain, inspired by Stephens (2000), with posterior distribution given by the likelihood of the data
- ▶ the type of chain is a birth-death chain (over an infinite dimensional state space!)

## The rough algorithm

- ▶ initialize clusters and labels randomly by sampling from the prior
- ▶ repeat the following 10,000 times
  - flip a weighted coin to determine if a birth or death occurs
  - if birth, add a new cluster randomly
  - if death, delete a cluster which doesn't explain the data well
- ▶ take the average of the last 5,000 labels to determine if a p-value is signal or null.

# Outline

Introduction

Method

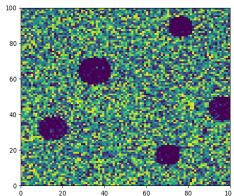
Example: simulated data

Example: fMRI data

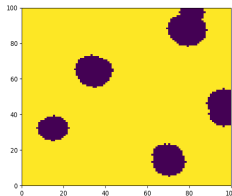
## Toy Data: Preliminaries

- ▶ 100-by-100 grid with  $k=5$  clusters of signals and the rest is null.
- ▶ the centers  $C_k \sim$  uniform from the grid while being distinct for  $k = 1, 2, \dots, 5$
- ▶ the radius  $R_k \sim TN(7, 2, 5, 10)$  for  $k = 1, 2, \dots, 5$
- ▶ signals in clusters  $p_{ki} \sim Beta(1, \beta_k)$  where  $\beta_k \sim TN(8, 5, 2, 200)$  for  $k = 1, 2, \dots, 5$

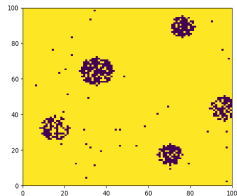
## Toy Data (continued): Performance



(a) Actual grid



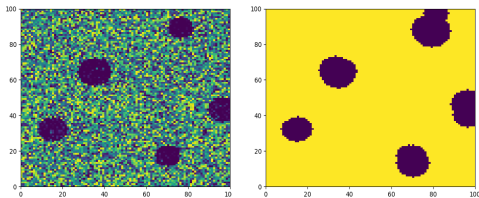
(b) Birth death chain



(c) BH

Figure: Simulated data

## Toy Data (continued): More on Priors



(a) Actual grid

(b)  $k \sim$

$$\text{Trun}(\text{Poi}(50), 1, 100);$$

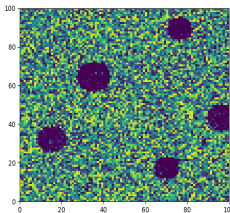
$$r \sim \text{TN}(7, 2, 5, 10);$$

$$\beta \sim \text{TN}(8, 5, 2, 200)$$

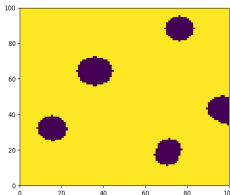
Figure: Simulated data

## Toy Data (continued): More on Priors

First, let's compare different priors on beta

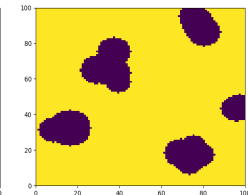


(a) Actual grid



(b)  $k \sim$

$Trun(Poi(50), 1, 100);$   
 $r \sim TN(7, 2, 5, 10);$   
 $\beta \sim TN(30, 3, 2, 200)$



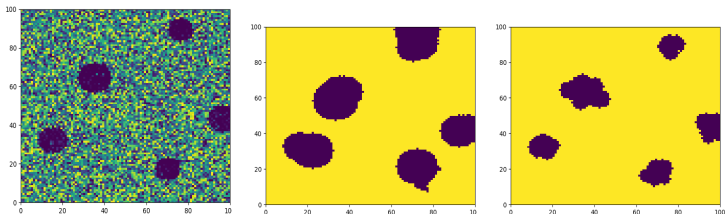
(c)  $k \sim$

$Trun(Poi(50), 1, 100);$   
 $r \sim TN(7, 2, 5, 10);$   
 $\beta \sim TN(2, 1, 2, 200)$

Figure: Simulated data

## Toy Data (continued): More on Priors

Now, let's compare different priors on radius.



(a) Actual grid

(b)  $k \sim$

$Trun(Poi(50), 1, 100);$   
 $r \sim TN(10, 2, 7, 13);$

(c)  $k \sim$

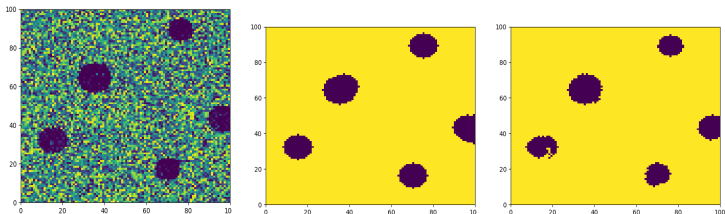
$Trun(Poi(50), 1, 100);$   
 $r \sim TN(2, 2, 1, 10);$   
 $\beta \sim TN(8, 5, 2, 200)$

Figure: Simulated data



## Toy Data (continued): More on Priors

What if we make signal stronger?



(a) Actual grid

(b)  $k \sim$

$Trun(Poi(50), 1, 100);$   
 $r \sim TN(10, 2, 7, 13);$   
 $\beta \sim TN(30, 3, 2, 200)$

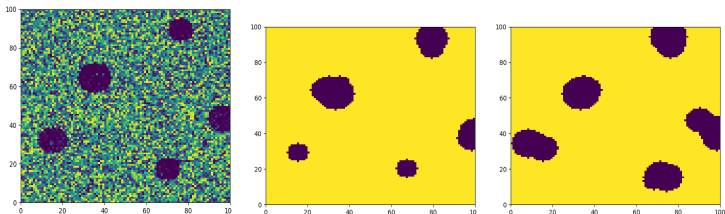
(c)  $k \sim$

$Trun(Poi(50), 1, 100);$   
 $r \sim TN(2, 2, 1, 10);$   
 $\beta \sim TN(30, 3, 2, 200)$

Figure: Simulated data

## Toy Data (continued): More on Priors

Next, we change the priors on number of clusters.



(a) Actual grid

(b)  $k \sim$

$\text{Trun}(\text{Poi}(3), 1, 10);$

$r \sim \text{TN}(7, 2, 5, 10);$

$\beta \sim \text{TN}(8, 5, 2, 200)$

(c)  $k \sim$

$\text{Trun}(\text{Poi}(300), 1, 1000);$

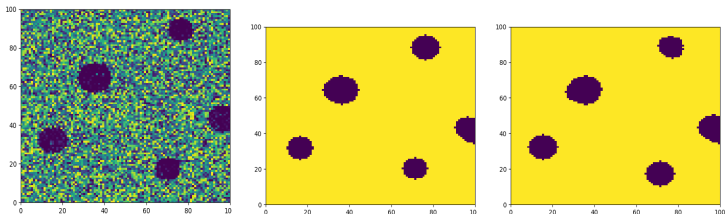
$r \sim \text{TN}(7, 2, 5, 10);$

$\beta \sim \text{TN}(8, 5, 2, 200)$

Figure: Simulated data

## Toy Data (continued): More on Priors

Again, let's make signal stronger



(a) Actual grid

(b)  $k \sim$

$$\text{Trun}(\text{Poi}(3), 1, 10);$$

$$r \sim \text{TN}(7, 2, 5, 10);$$

$$\beta \sim \text{TN}(30, 3, 2, 200)$$

(c)  $k \sim$

$$\text{Trun}(\text{Poi}(300), 1, 1000);$$

$$r \sim \text{TN}(7, 2, 5, 10);$$

$$\beta \sim \text{TN}(30, 3, 2, 200)$$

Figure: Simulated data

## Toy Data 2 (3-D)

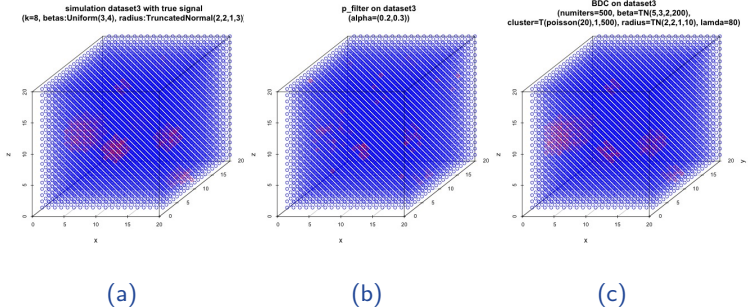


Figure: Simulated data

## Toy Data 2 (3-D)

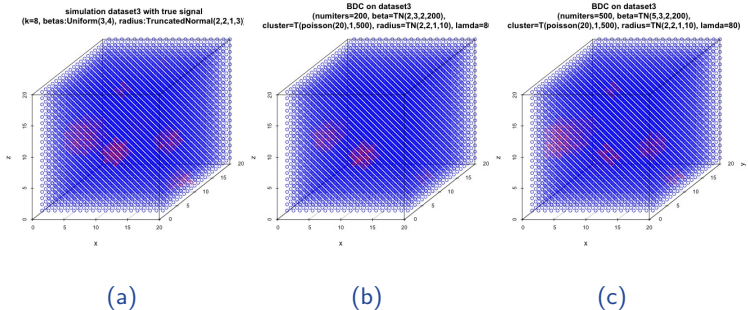


Figure: Simulated data

## Toy Data 2 (3-D)

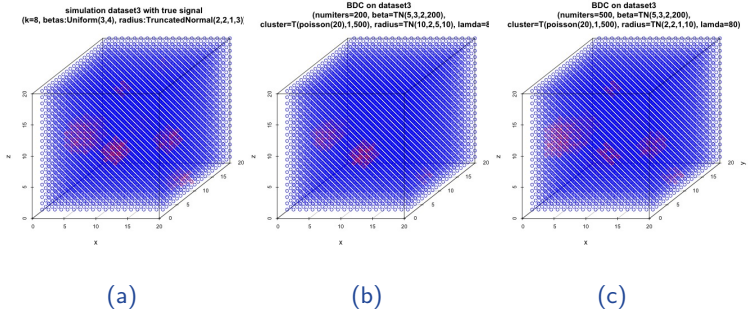


Figure: Simulated data

## Toy Data 2 (3-D)

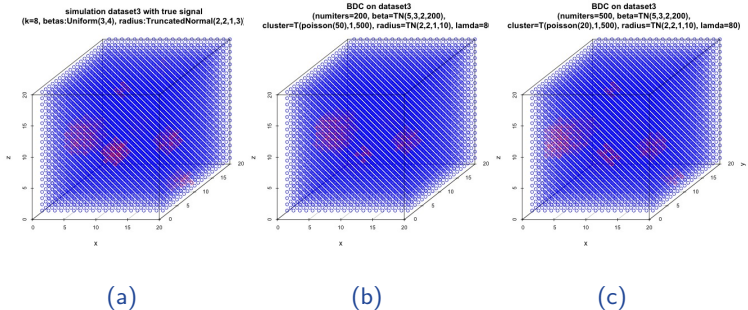


Figure: Simulated data

# Outline

Introduction

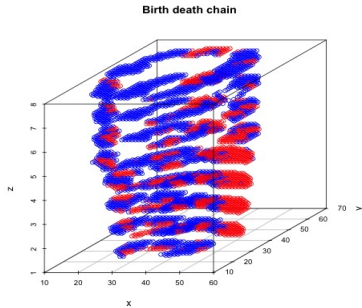
Method

Example: simulated data

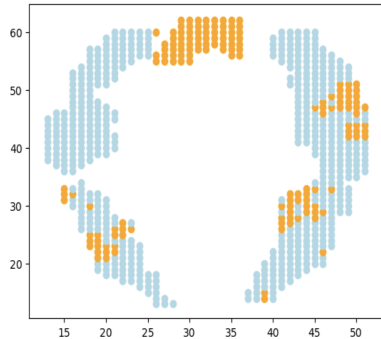
Example: fMRI data



## Performance on real fMRI data



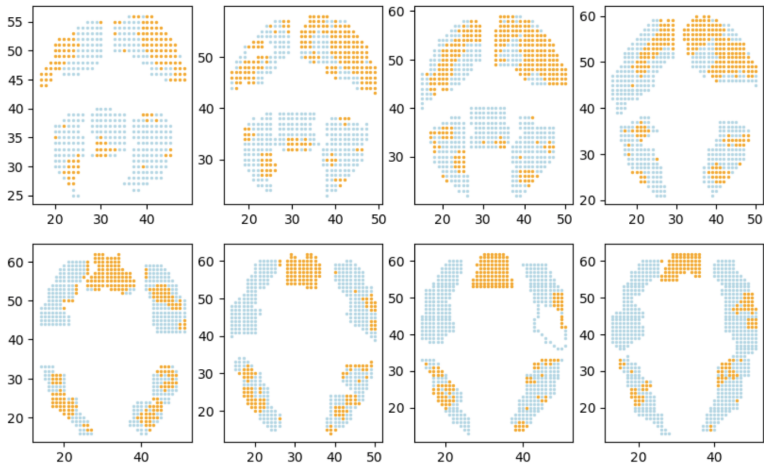
(a) 3-D view



(b) 2-D view: one slice

Figure: fMRI data

## Performance on real fMRI data (continued)



## Performance on real fMRI data (continued): Comparison to p-filter

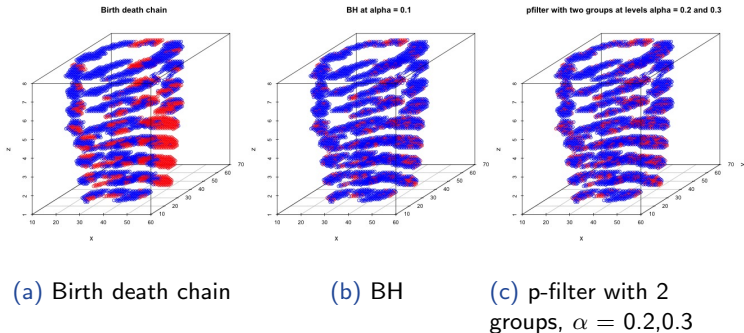


Figure: fMRI data

## Conclusion and Future Work

- ▶ presented a nonparametric-bayesian method to adaptively identify clusters of signals.
- ▶ it showed promising results on both simulation and real fMRI data.
- ▶ possible extensions
  - work with intensities directly by specifying appropriate priors for null distributions and for signal distributions.
  - put priors on the hyper-parameters and maximize these priors using EM

**Thanks!**