

Fast Discovery of Pairwise Interactions in High Dimensions using Bayes

Tamara Broderick

Associate Professor
EECS, MIT

Raj Agrawal, Jonathan H. Huggins, Brian L. Trippe



Gene expression levels

Person 1

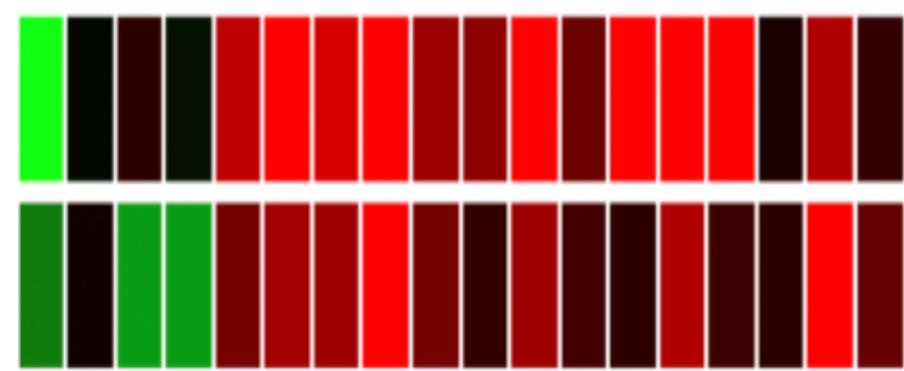


Person 2



:

Person N



Environmental factors

Gene expression levels

Person 1

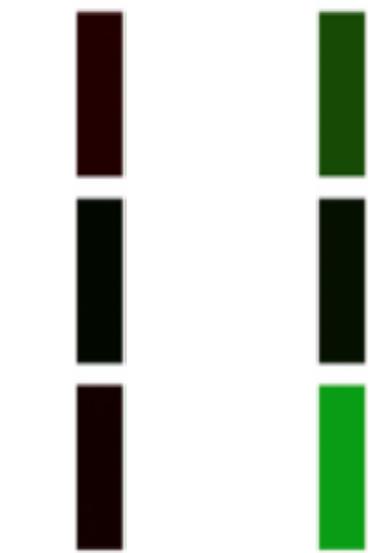
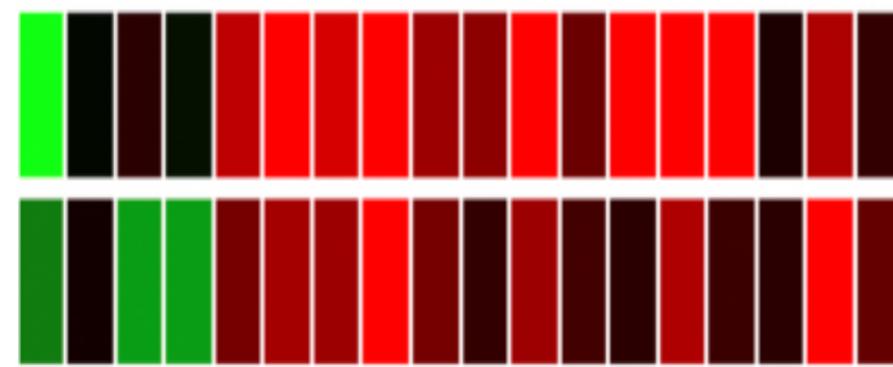


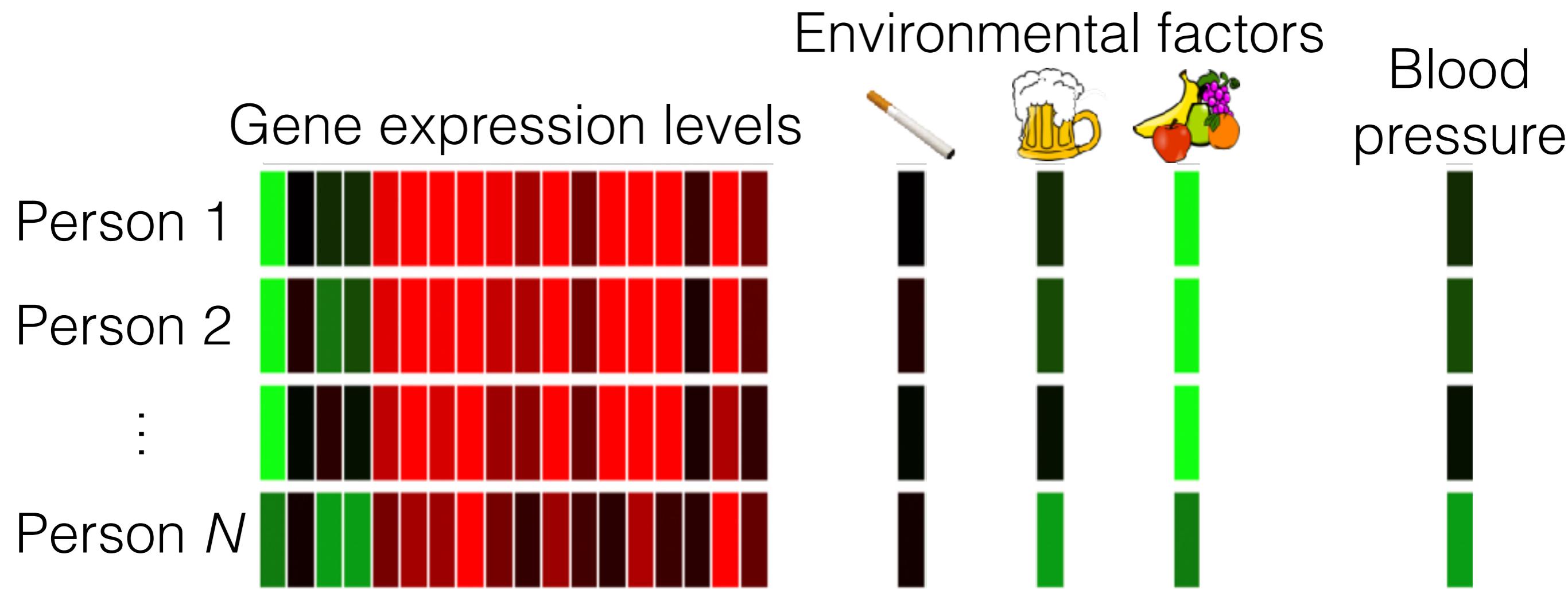
Person 2

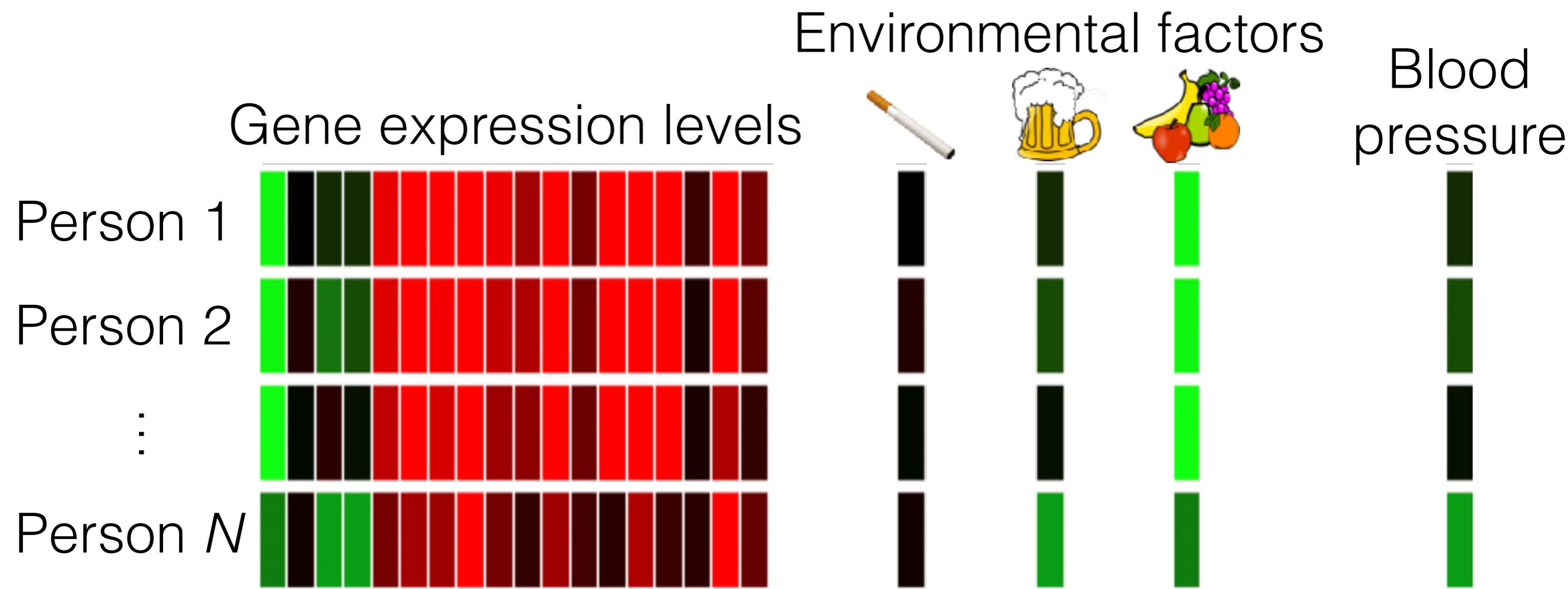


:

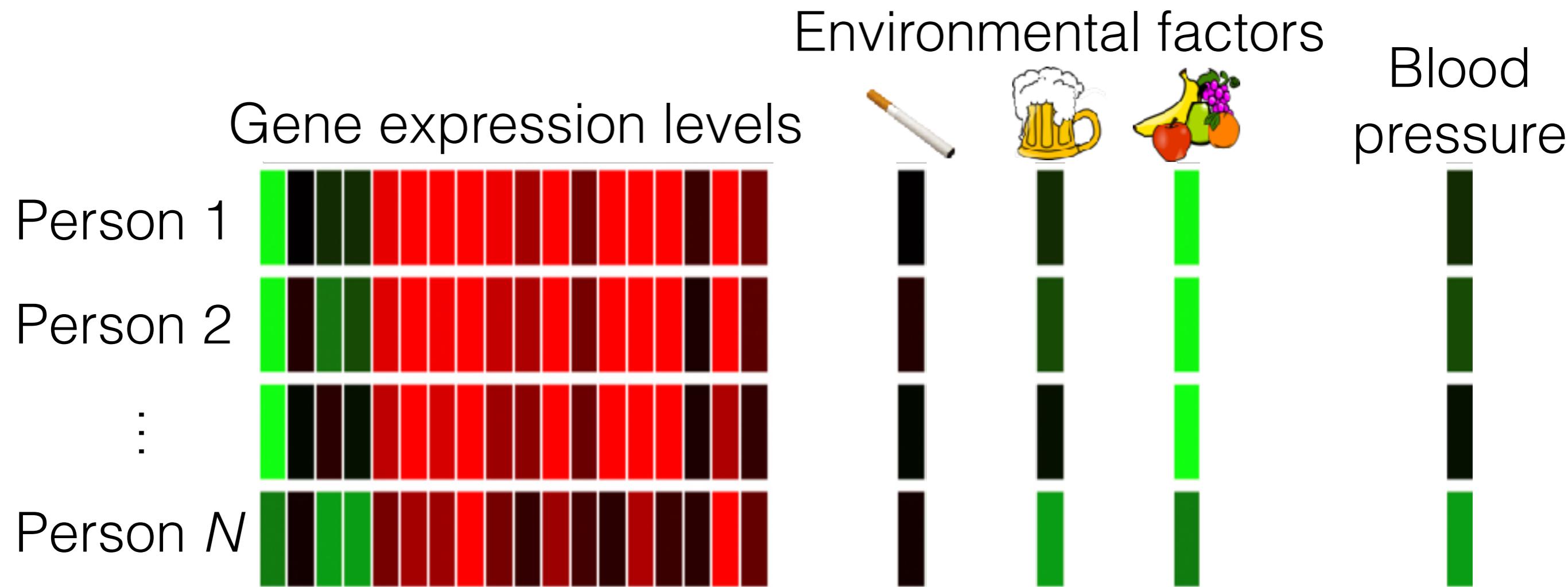
Person N



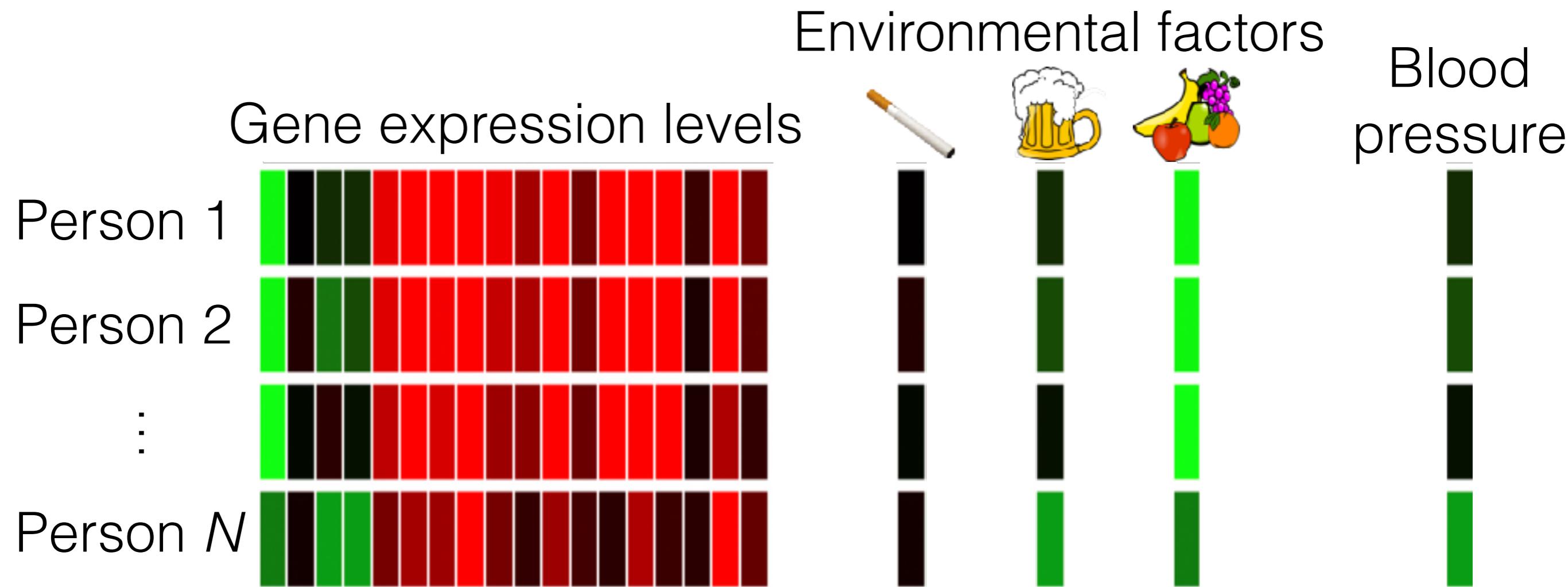




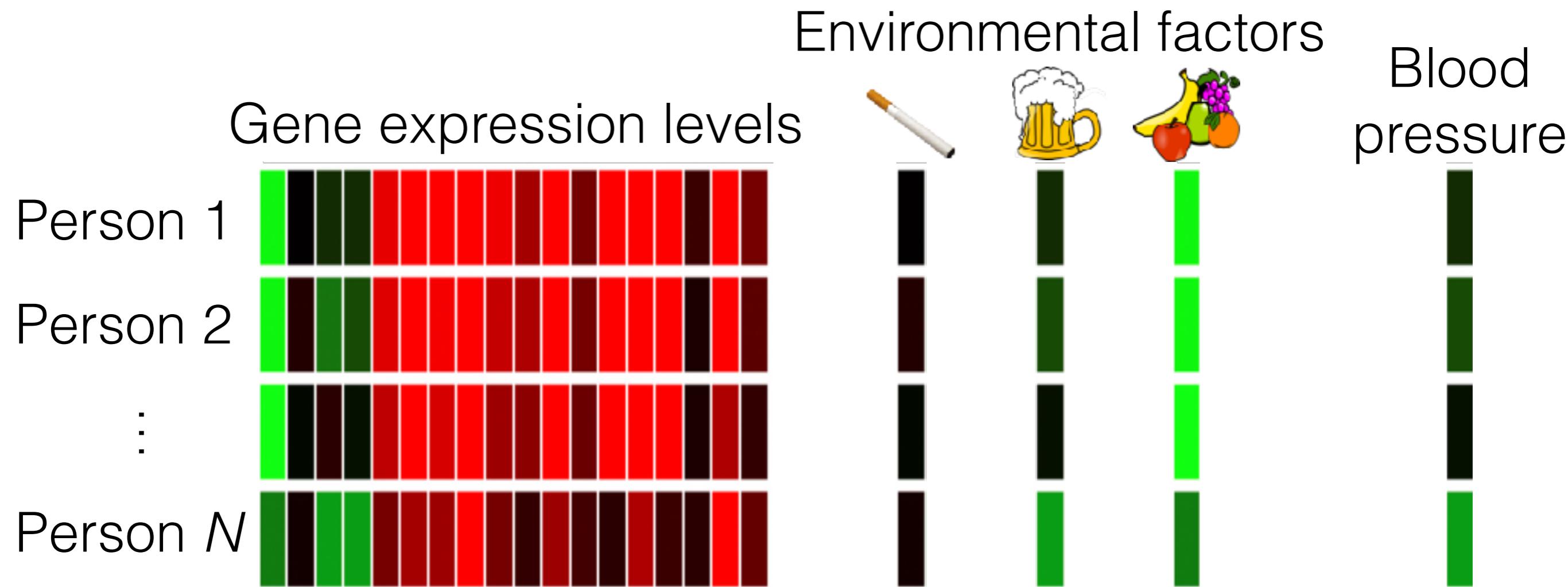
- Which genes/factors are associated with a health issue?



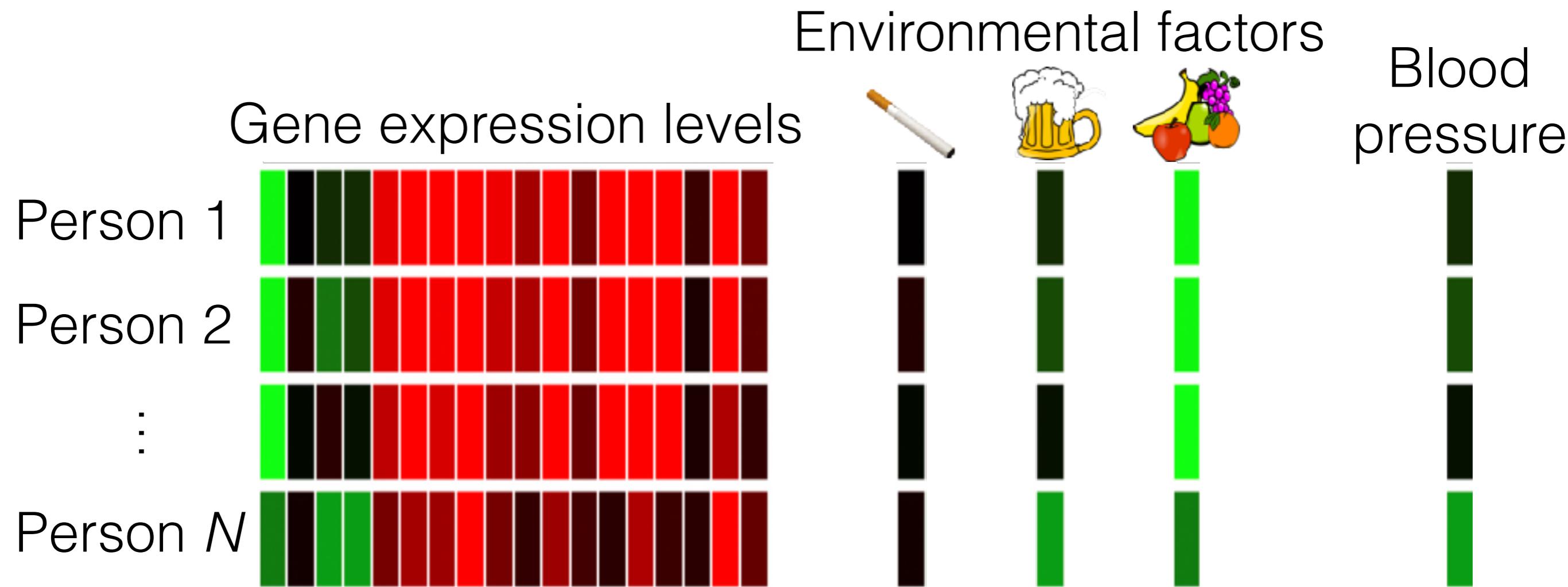
- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates



- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)

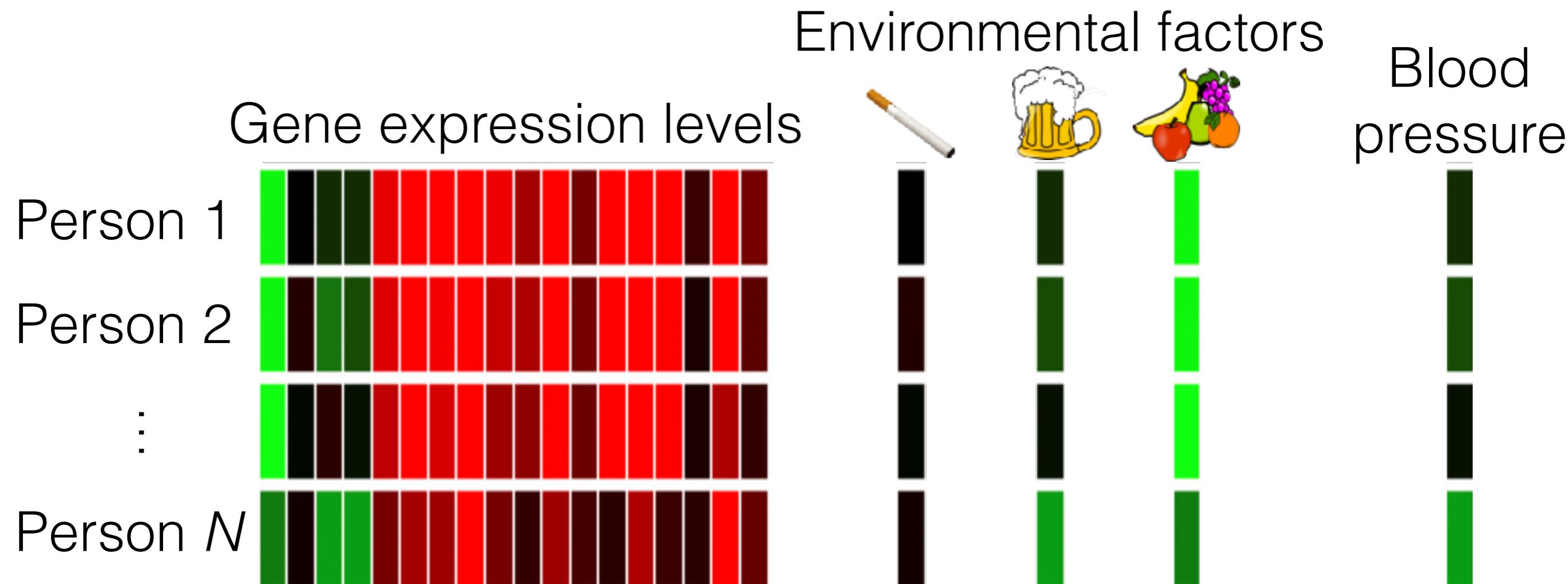


- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions



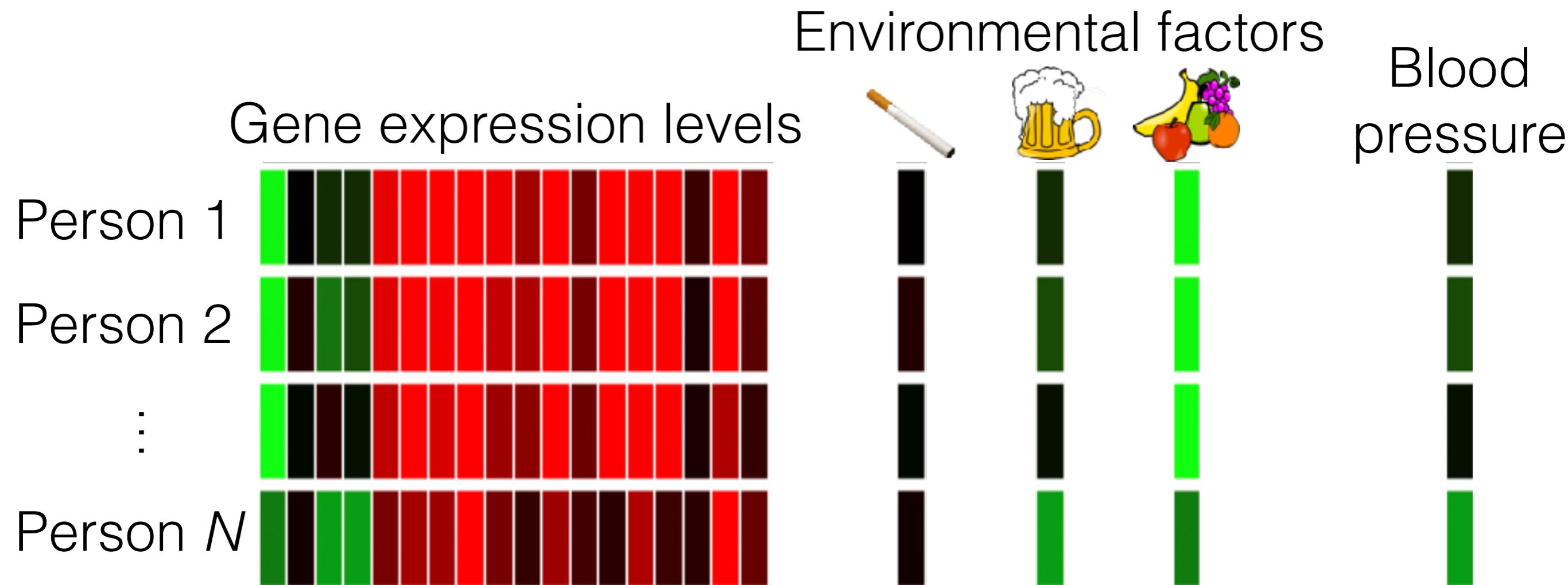
- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions (now p^2 dims!)

Pairwise interactions in high dimensions



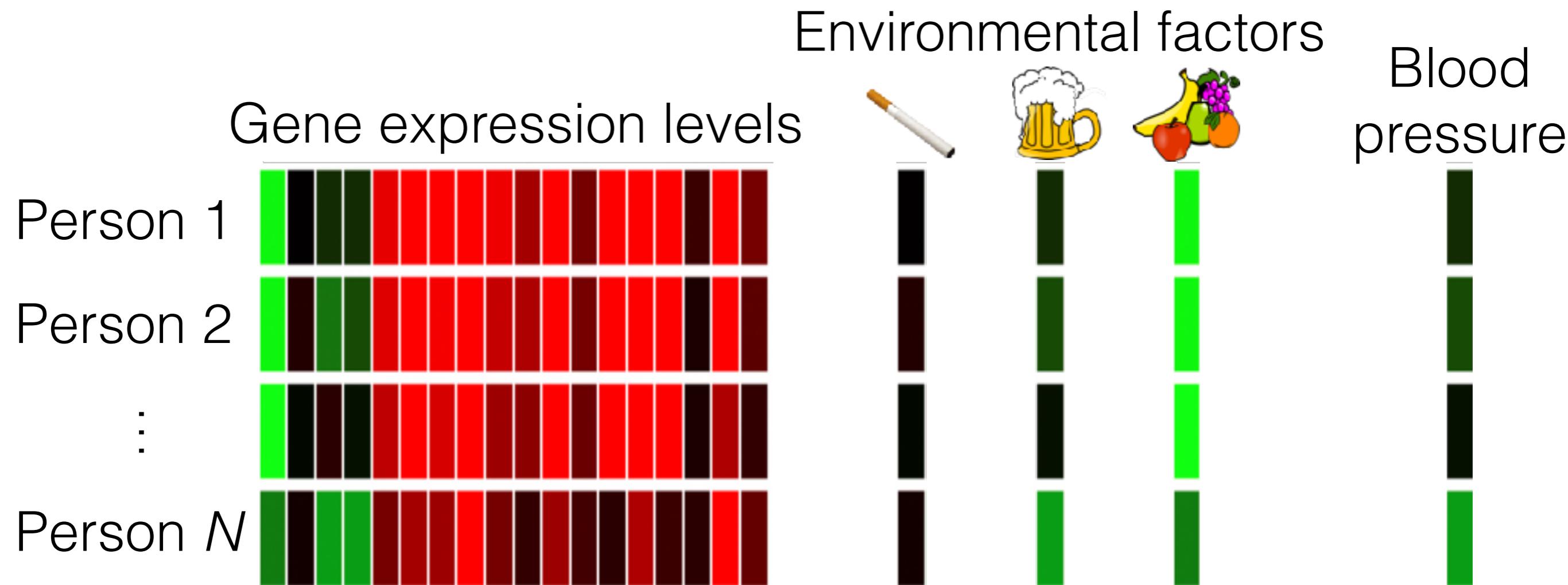
- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions (now p^2 dims!)

Pairwise interactions in high dimensions



- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions (now p^2 dims!)
- **We provide:** Fast, accurate (Bayes) method for interaction discovery

Pairwise interactions in high dimensions



- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions (now p^2 dims!)
- **We provide:** Fast, accurate (Bayes) method for interaction discovery
 - Better scaling in p & better accuracy than LASSO-based methods.
Orders of magnitude faster than naive Bayesian inference

Roadmap

Roadmap

- Setup: Discovering main and interaction effects

Roadmap

- Setup: Discovering main and interaction effects
- Our method

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference
 - Fast reporting of results

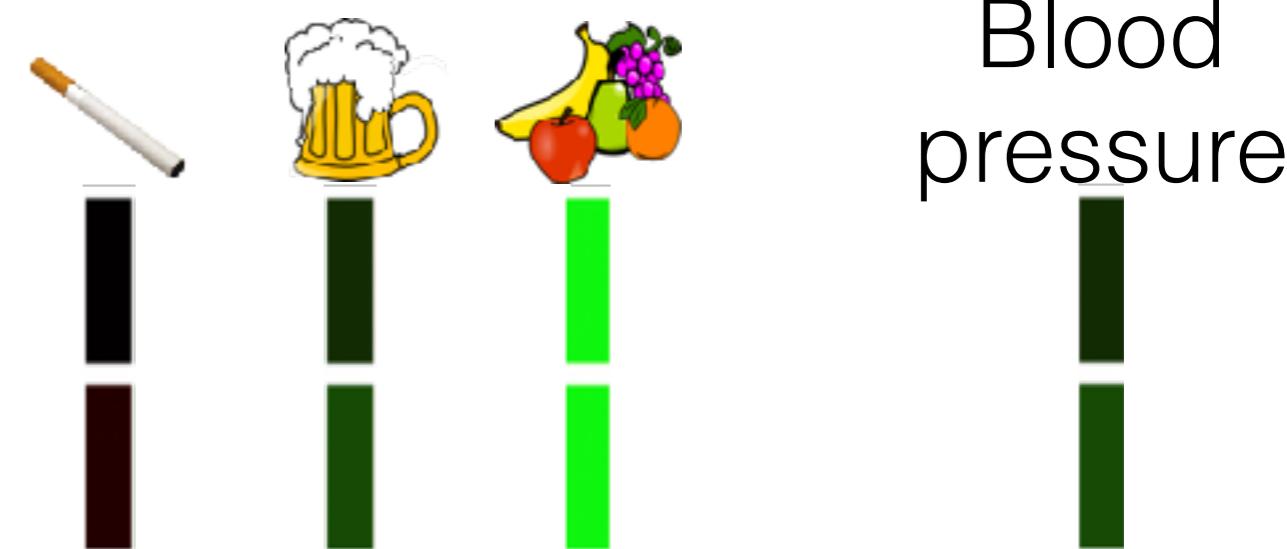
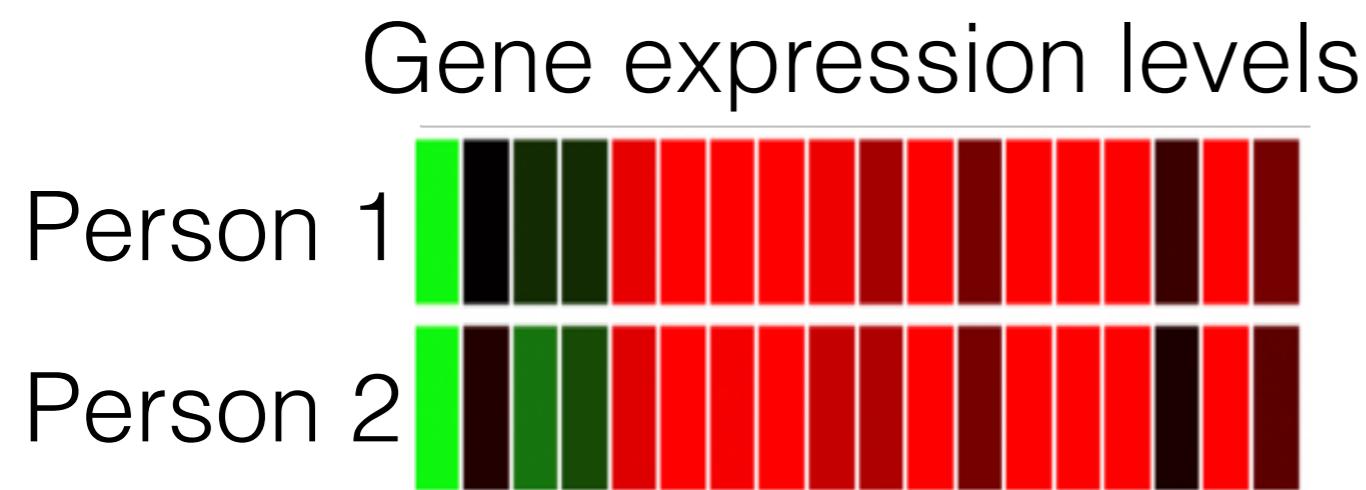
Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference
 - Fast reporting of results
- Experiments on simulated and real data

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference
 - Fast reporting of results
- Experiments on simulated and real data

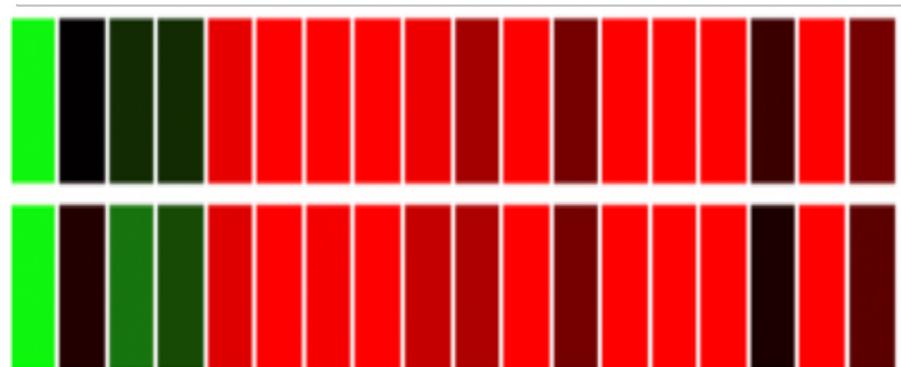
Discovering main and interaction effects



Discovering main and interaction effects

Gene expression levels

Person 1



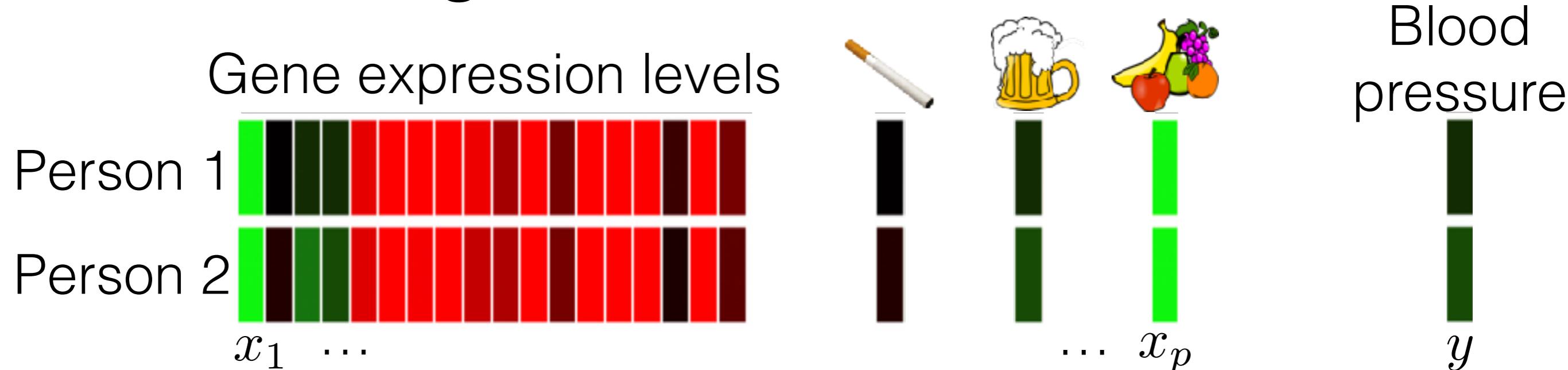
$x_1 \dots$



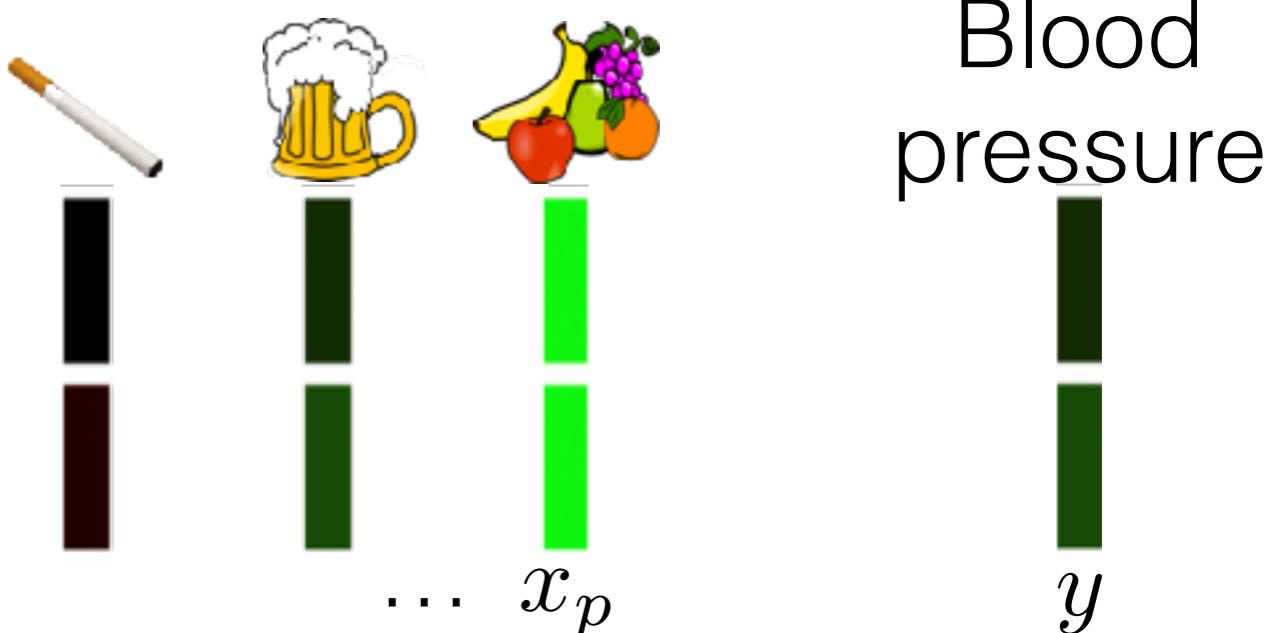
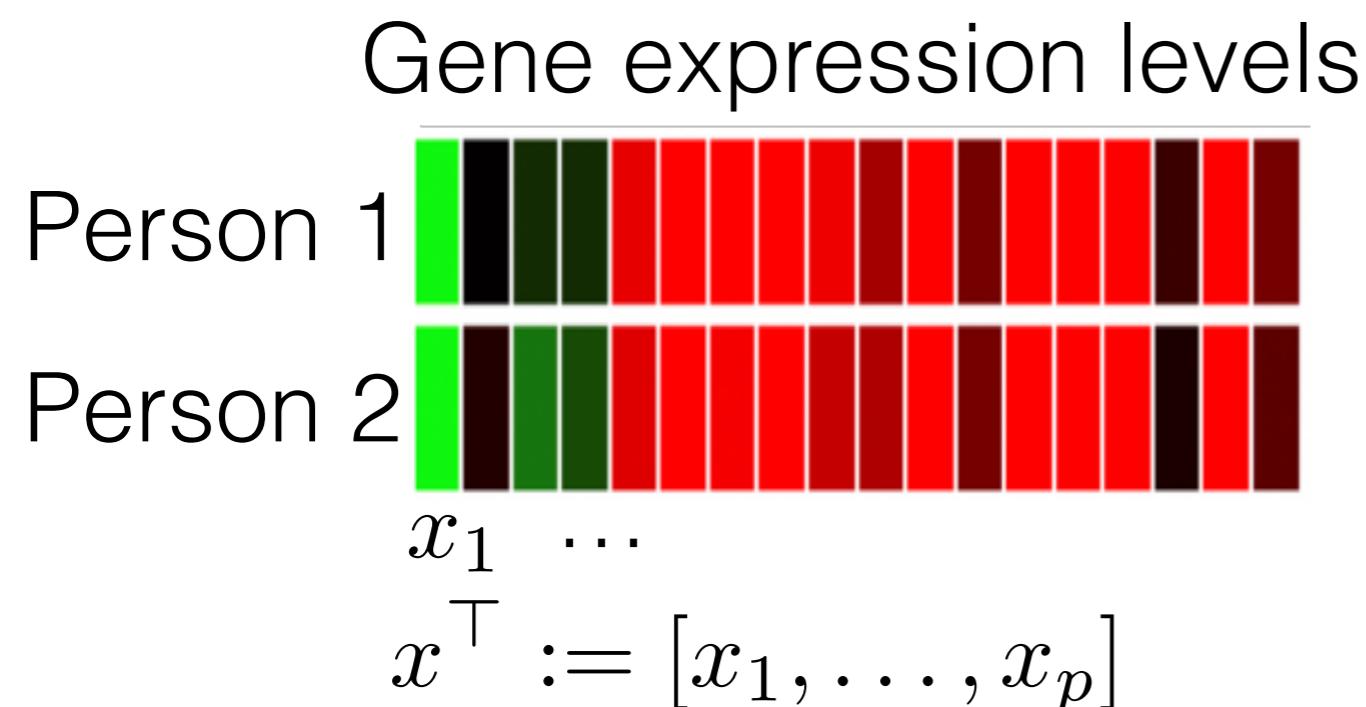
Blood pressure

... x_p

Discovering main and interaction effects



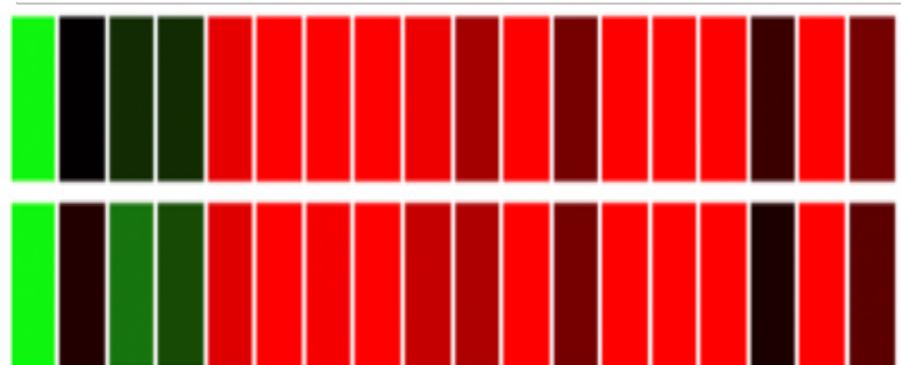
Discovering main and interaction effects



Discovering main and interaction effects

Gene expression levels

Person 1



Person 2

$x_1 \dots$

$$x^\top := [x_1, \dots, x_p]$$

$$y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$



Blood
pressure

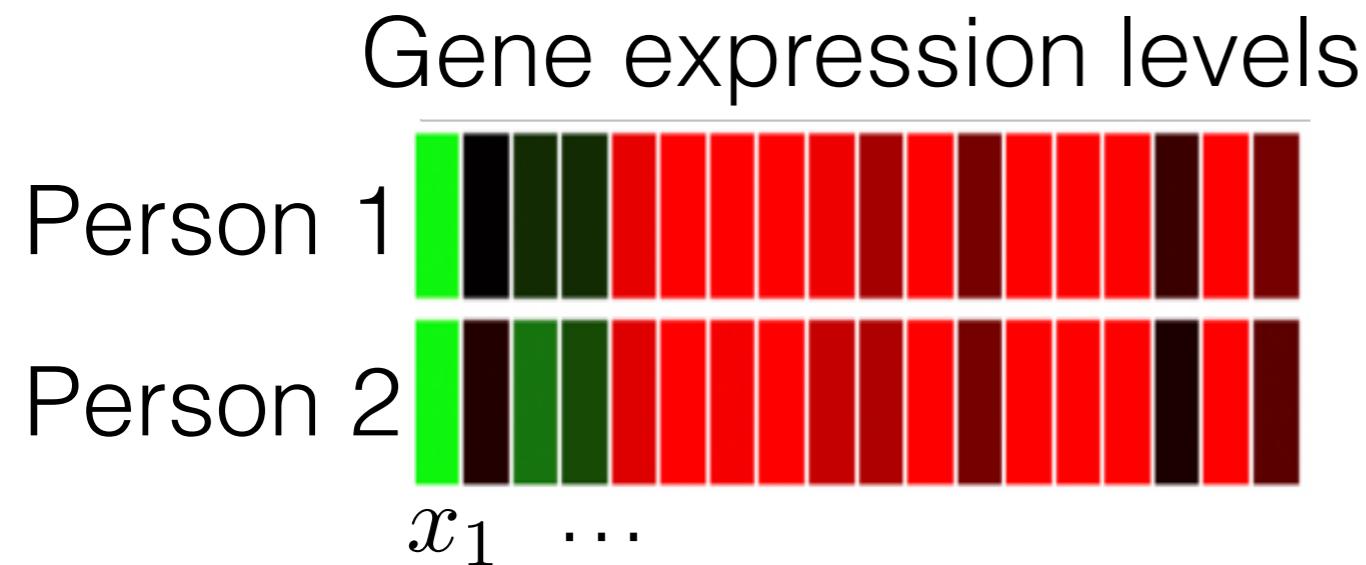


$\dots x_p$



y

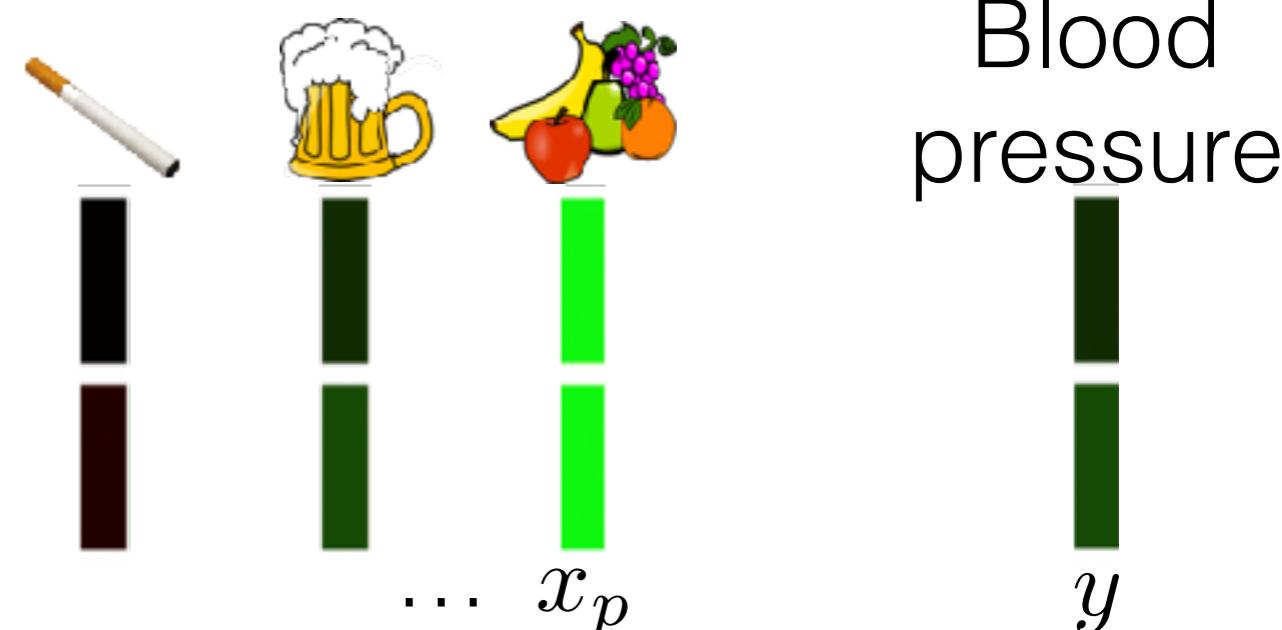
Discovering main and interaction effects



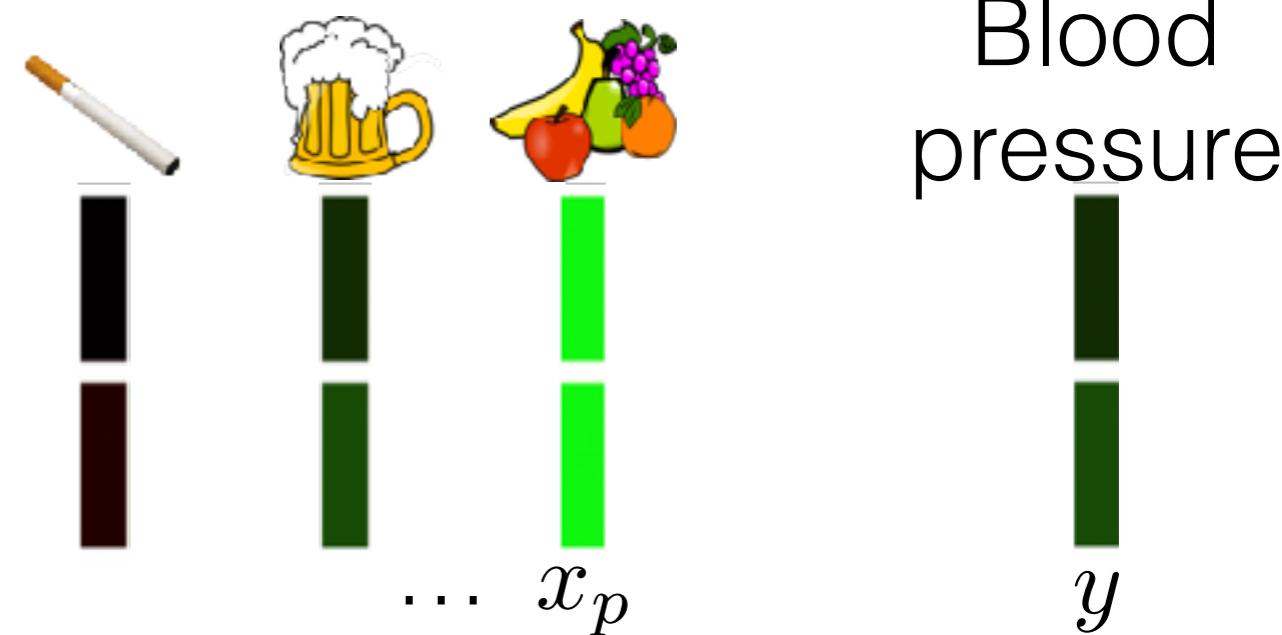
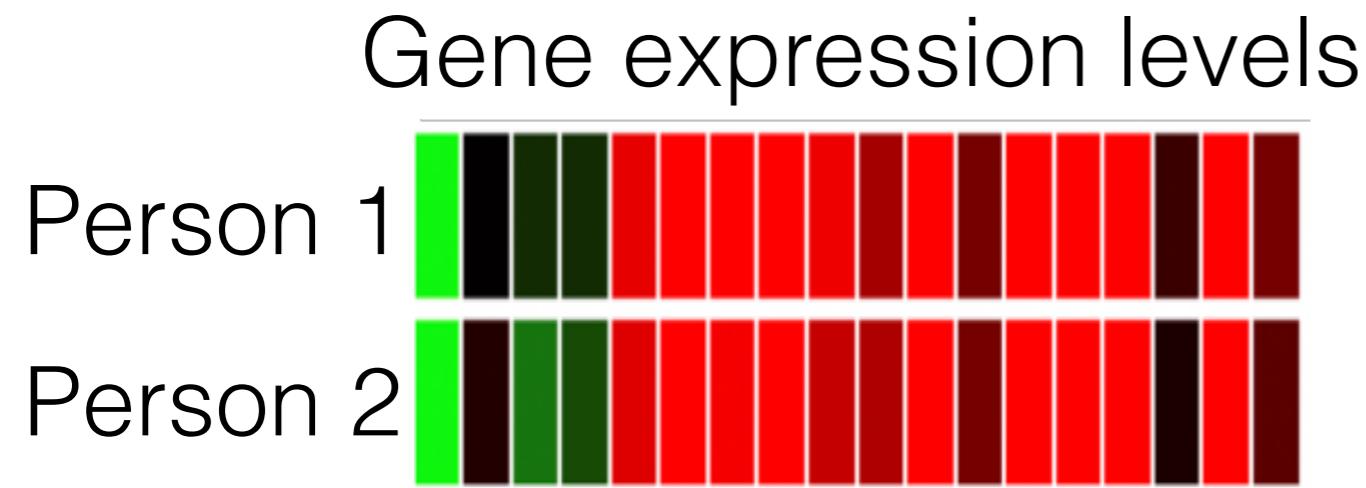
$x_1 \dots$

$$x^\top := [1, x_1, \dots, x_p]$$

$$y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$



Discovering main and interaction effects



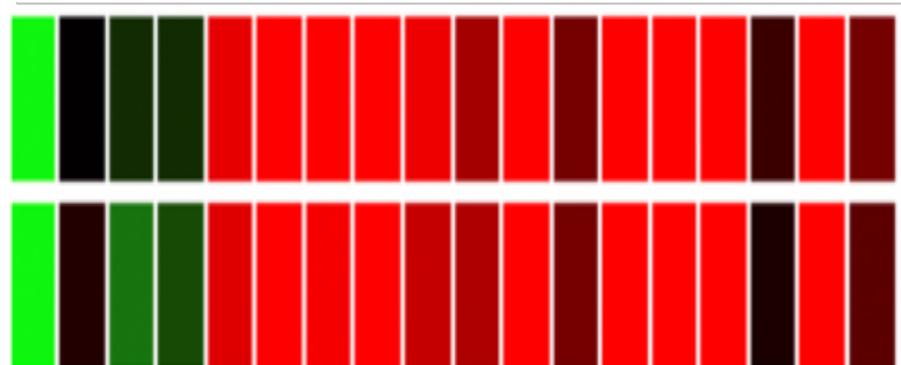
$$x^\top := [1, x_1, \dots, x_p]$$

$$y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

Discovering main and interaction effects

Gene expression levels

Person 1



Person 2



$x_1 \dots$



Blood pressure



y

$\dots x_p$

$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1x_2, \dots, x_{p-1}x_p, x_1^2, \dots, x_p^2]$$
$$y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

Discovering main and interaction effects

Gene expression levels

Person 1



Person 2



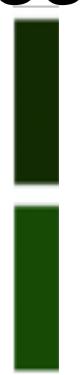
$x_1 \dots$



Blood
pressure



$\dots x_p$



y

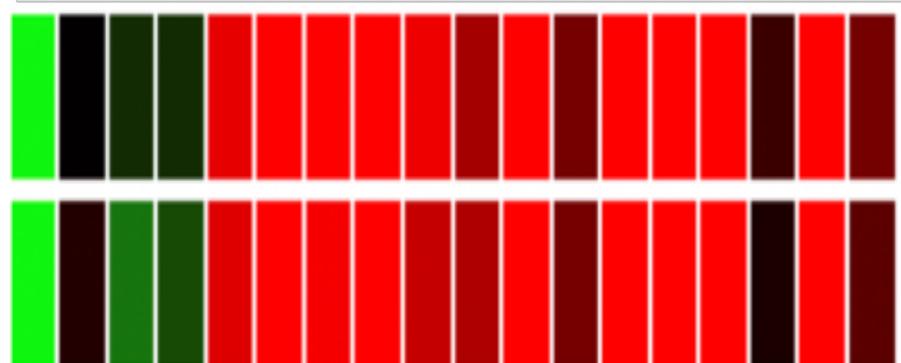
$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1 x_2, \dots, x_{p-1} x_p, x_1^2, \dots, x_p^2]$$

$$y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

Discovering main and interaction effects

Gene expression levels

Person 1



Person 2



$x_1 \dots$



Blood
pressure



$\dots x_p$

y

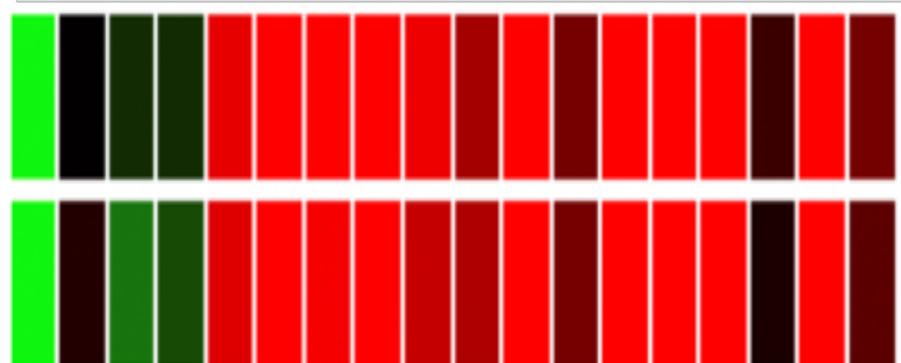
$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1x_2, \dots, x_{p-1}x_p, x_1^2, \dots, x_p^2]$$

$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

Discovering main and interaction effects

Gene expression levels

Person 1



Person 2



$x_1 \dots$



Blood pressure



$\dots x_p$



y

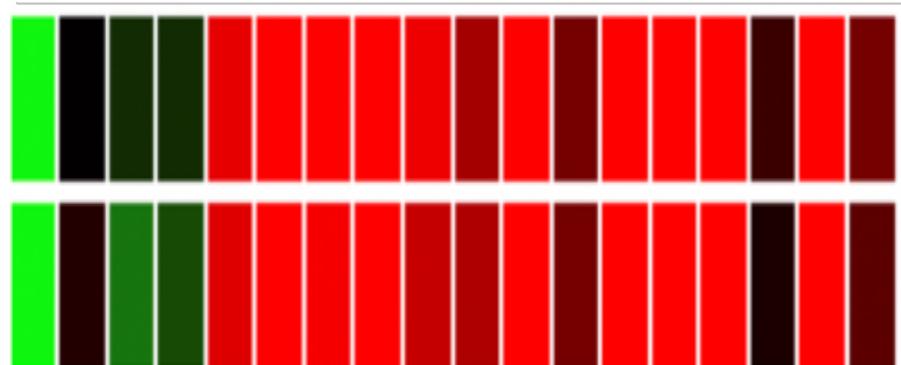
$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1 x_2, \dots, x_{p-1} x_p, x_1^2, \dots, x_p^2]$$

$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

Discovering main and interaction effects

Gene expression levels

Person 1



Person 2



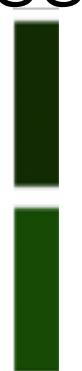
$x_1 \dots$



Blood
pressure



$\dots x_p$



y

$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1 x_2, \dots, x_{p-1} x_p, x_1^2, \dots, x_p^2]$$

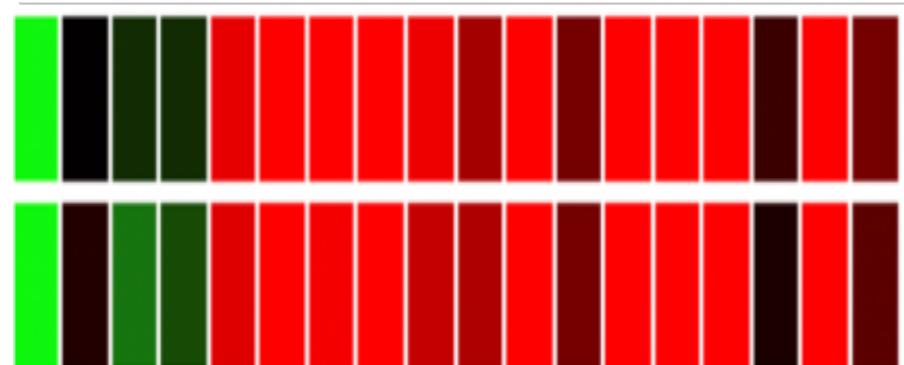
$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

- **Goal:** Parameter selection/estimation

Discovering main and interaction effects

Gene expression levels

Person 1



Person 2



$x_1 \dots$



Blood
pressure



$\dots x_p$



y

$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1 x_2, \dots, x_{p-1} x_p, x_1^2, \dots, x_p^2]$$

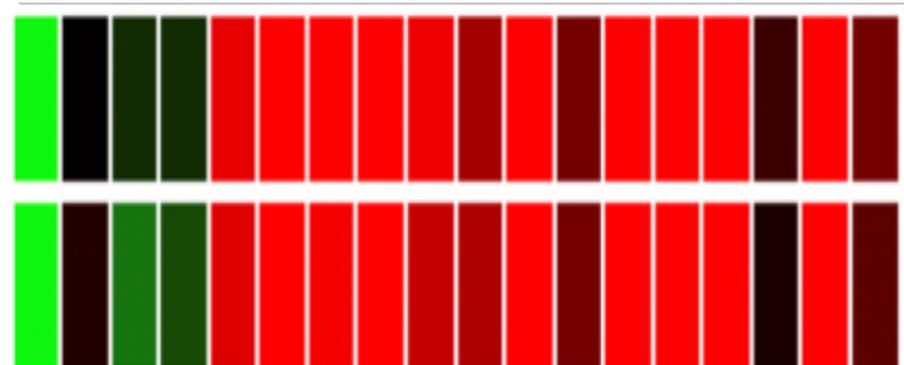
$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

- **Goal:** Parameter selection/estimation under assumptions:

Discovering main and interaction effects

Gene expression levels

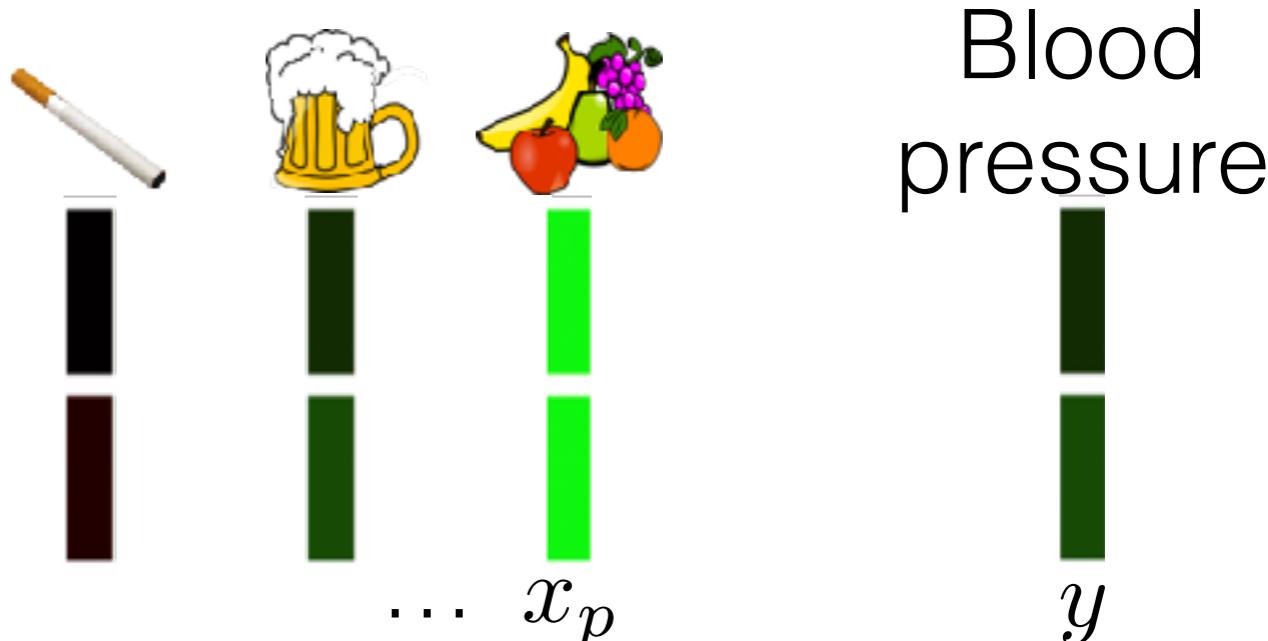
Person 1



Person 2



$x_1 \dots$

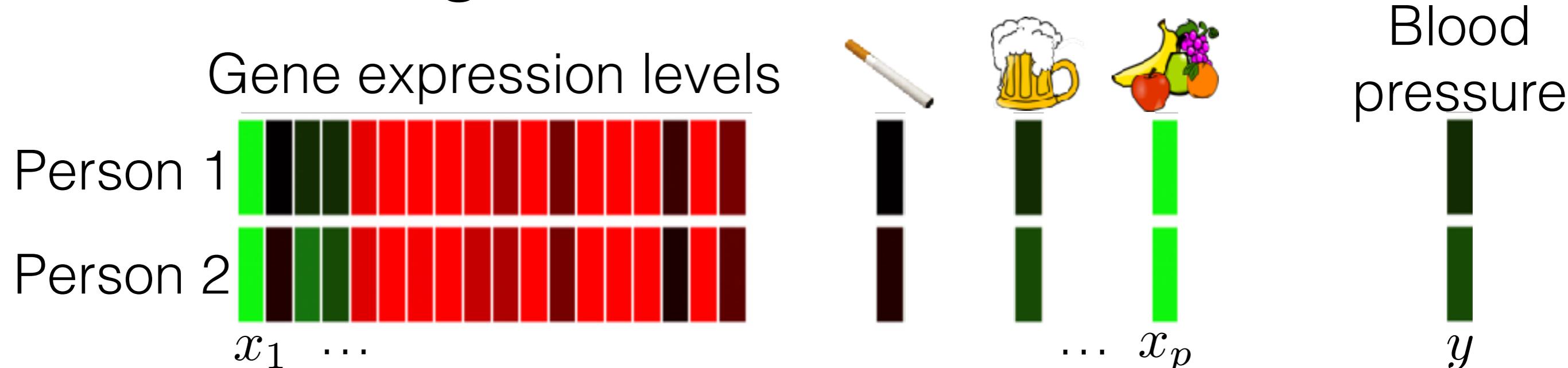


$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1x_2, \dots, x_{p-1}x_p, x_1^2, \dots, x_p^2]$$

$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

- **Goal:** Parameter selection/estimation under assumptions:
 - *Sparsity:* most main effects are negligible (interpretable)

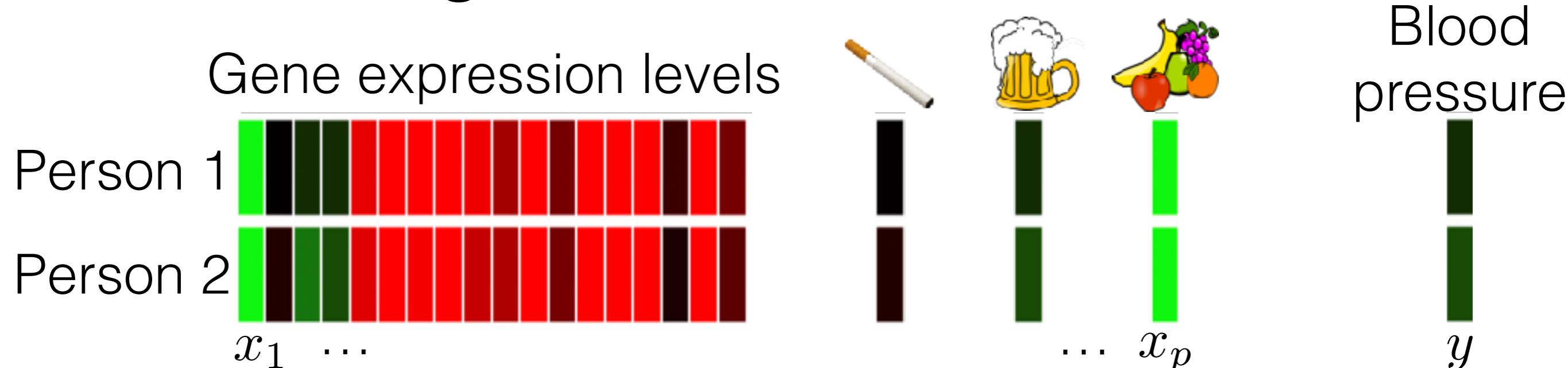
Discovering main and interaction effects



$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1 x_2, \dots, x_{p-1} x_p, x_1^2, \dots, x_p^2]$$
$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

- **Goal:** Parameter selection/estimation under assumptions:
 - *Sparsity:* most main effects are negligible (interpretable)
 - *Strong hierarchy:* Interaction only if main effects are present

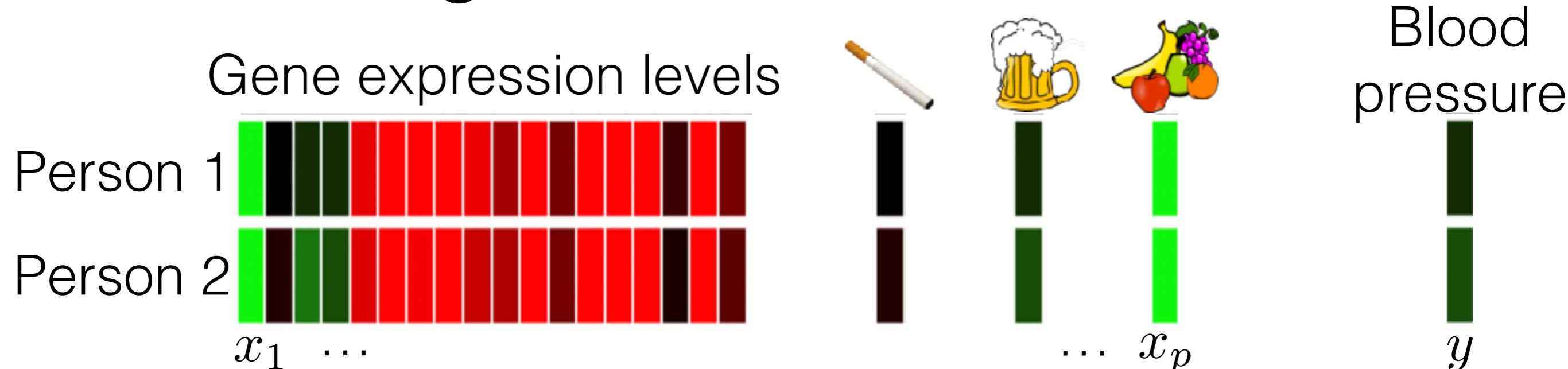
Discovering main and interaction effects



$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1x_2, \dots, x_{p-1}x_p, x_1^2, \dots, x_p^2]$$
$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

- **Goal:** Parameter selection/estimation under assumptions:
 - *Sparsity:* most main effects are negligible (interpretable)
 - *Strong hierarchy:* Interaction only if main effects are present
 - p^2 covariates: large $p \rightarrow$ statistical & computational challenge

Discovering main and interaction effects



$$\Phi_2^\top(x) := [1, x_1, \dots, x_p, x_1 x_2, \dots, x_{p-1} x_p, x_1^2, \dots, x_p^2]$$
$$y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

- **Goal:** Parameter selection/estimation under assumptions:
 - *Sparsity:* most main effects are negligible (interpretable)
 - *Strong hierarchy:* Interaction only if main effects are present
- p^2 covariates: large $p \rightarrow$ statistical & computational challenge
- **Our solution:** using structure in covariates + sparsity assumptions to reduce to a problem *linear* in p

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference
 - Fast reporting of results
- Experiments on simulated and real data

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference
 - Fast reporting of results
- Experiments on simulated and real data

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model
2. Compute posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model
2. Compute posterior
3. Report relevant summaries of the posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model
2. Compute posterior
3. Report relevant summaries of the posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy
2. Compute posterior
3. Report relevant summaries of the posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]
2. Compute posterior
3. Report relevant summaries of the posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]
2. Compute posterior
3. Report relevant summaries of the posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]
2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration
3. Report relevant summaries of the posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]
2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration
3. Report relevant summaries of the posterior

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]
2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration
3. **Kernel Interaction Trick (KIT)**: Use kernel trick to report *all* non-negligible main and interaction effects in $O(p)$ time

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]
2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration
3. **Kernel Interaction Trick (KIT)**: Use kernel trick to report *all* non-negligible main and interaction effects in $O(p)$ time

Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration

3. **Kernel Interaction Trick (KIT)**: Use kernel trick to report *all* non-negligible main and interaction effects in $O(p)$ time



Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample θ

Kernel Interaction Sampler vs. Naive MCMC

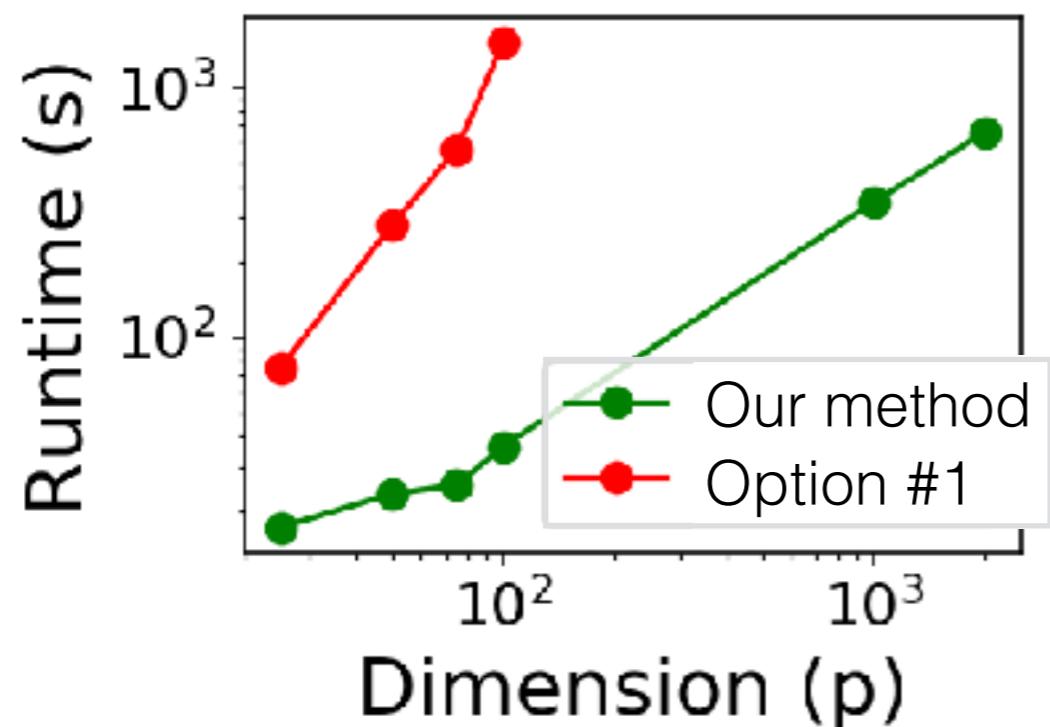
- MCMC option 1: sample θ (p^2 parameters)

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample θ (p^2 parameters)
 - Time cost: $O(p^2N)$

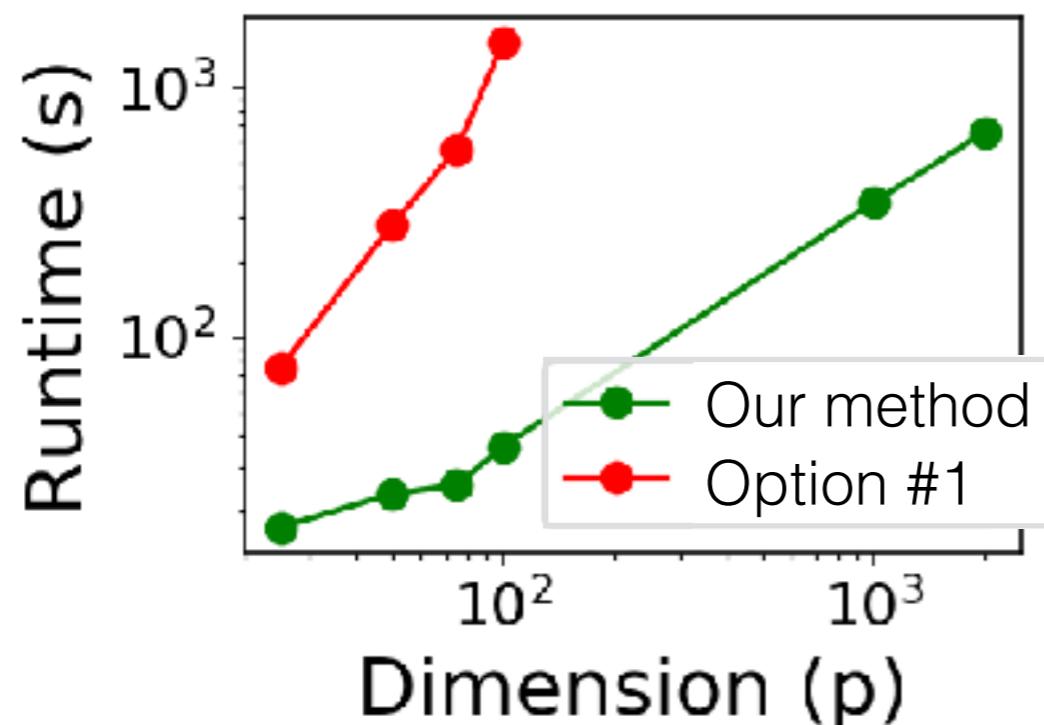
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample θ (p^2 parameters)
 - Time cost: $O(p^2N)$



Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample θ (p^2 parameters)
 - Time cost: $O(p^2N)$



- Mixing (1000 iters Stan):
 - Option #1: all $\hat{R} > 1.05$
 - Our method: all $\hat{R} < 1.05$

Kernel Interaction Sampler vs. Naive MCMC

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$X^\top X$$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$X^\top X + \text{ prior precision matrix}$$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$X^\top X$$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert
$$\Phi_2(X)^\top \Phi_2(X)$$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$$X: N \times p$$

$$\Phi_2: N \times p^2$$

$$N \quad p^2$$

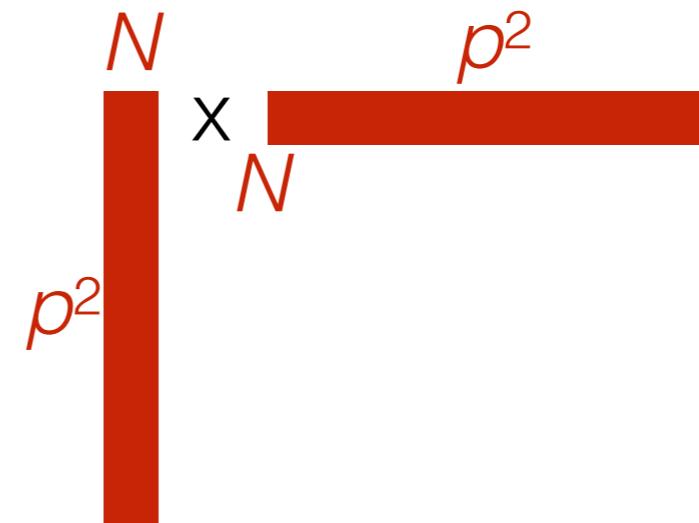
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$



Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$

$$\begin{array}{c} N \\ | \\ \times \\ | \\ p^2 \end{array} = \begin{array}{c} p^2 \\ | \\ N \end{array}$$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$

$$\begin{array}{c} N \\ \times \\ N \end{array} = \begin{array}{c} p^2 \\ p^2 \\ p^2 \end{array}$$

Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$

$$\begin{array}{c|c} N & p^2 \\ \times & N \\ \hline p^2 & \end{array} = \begin{array}{c|c} & p^2 \\ & \vdots \\ & p^2 \end{array}$$

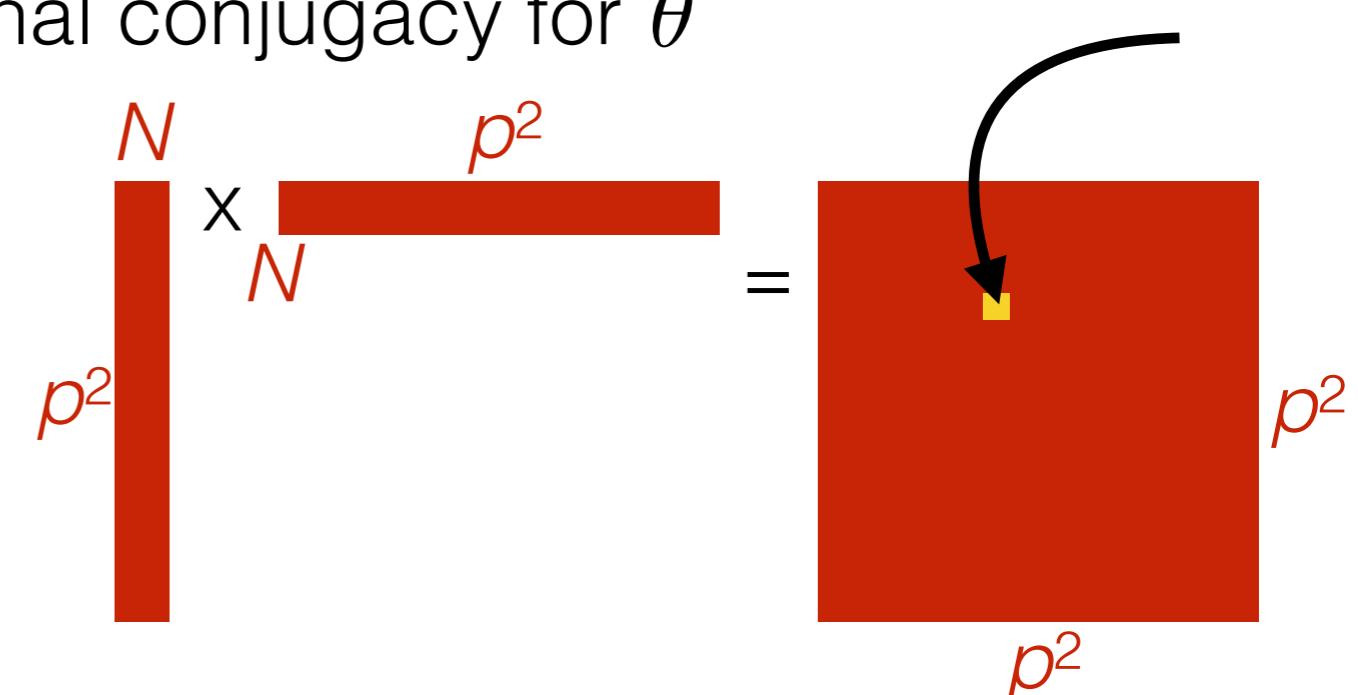
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$



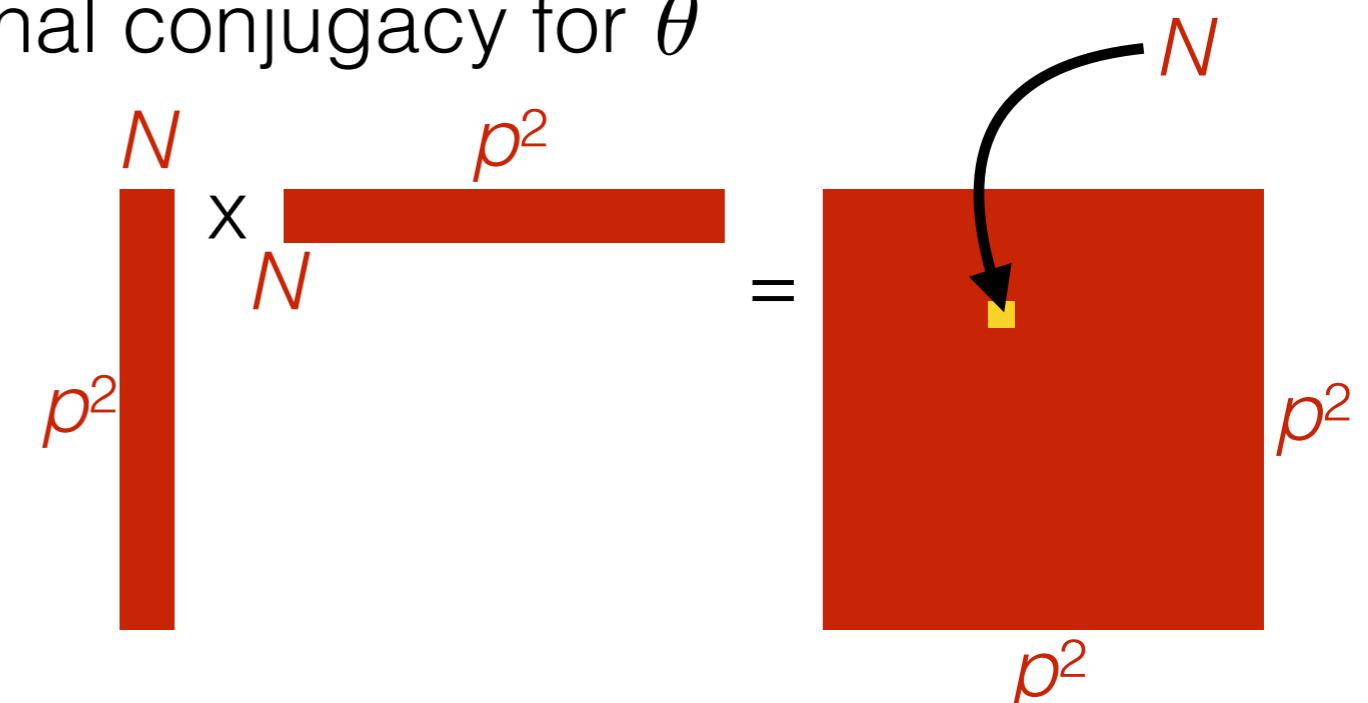
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ
 - Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$



Kernel Interaction Sampler vs. Naive MCMC

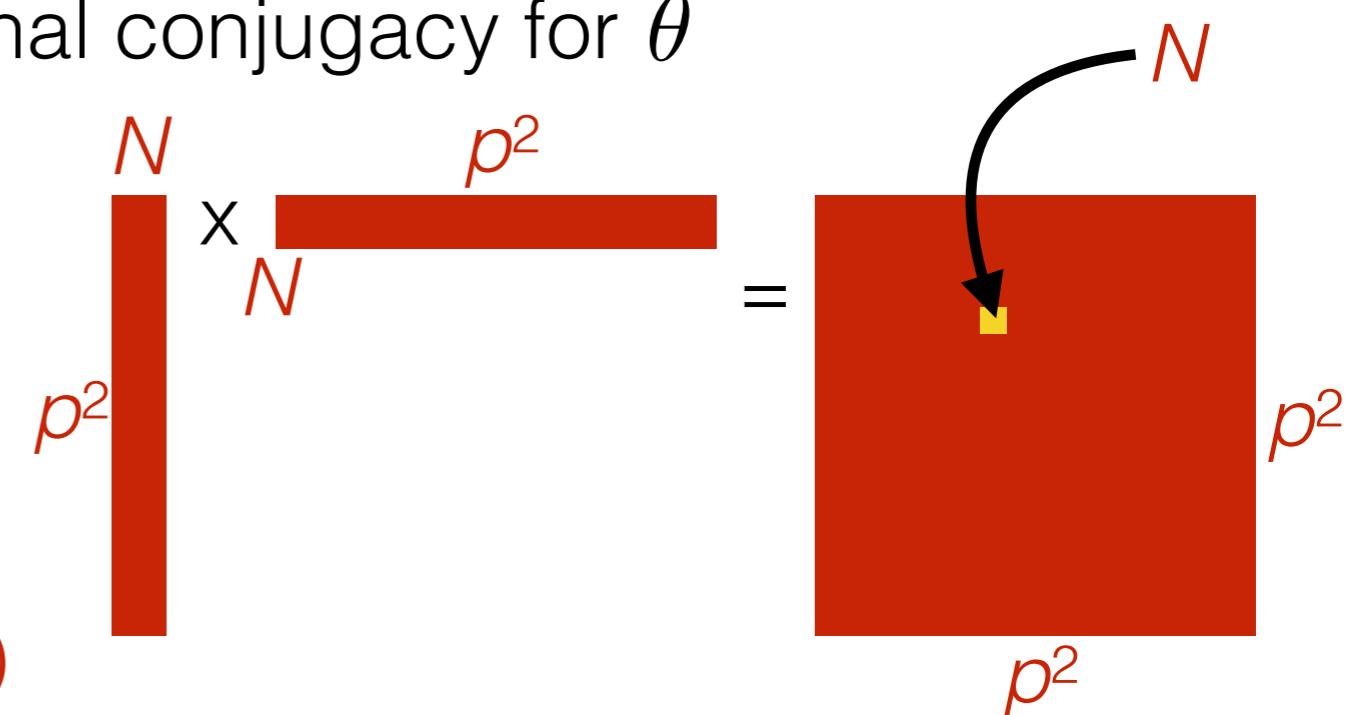
- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$



- Naive time cost: $O(p^4N + p^6)$

Kernel Interaction Sampler vs. Naive MCMC

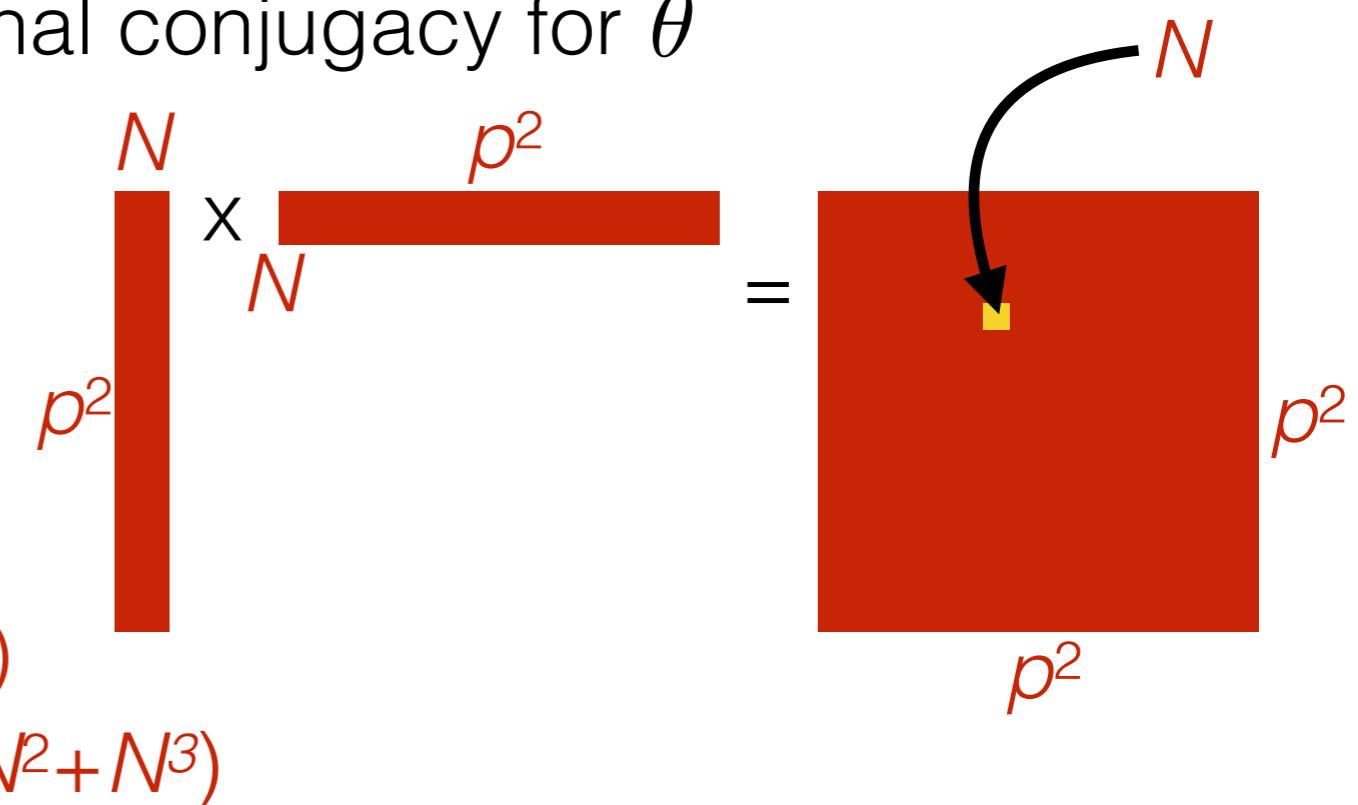
- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$



- Naive time cost: $O(p^4N + p^6)$
 - Woodbury time cost: $O(p^2N^2 + N^3)$

Kernel Interaction Sampler vs. Naive MCMC

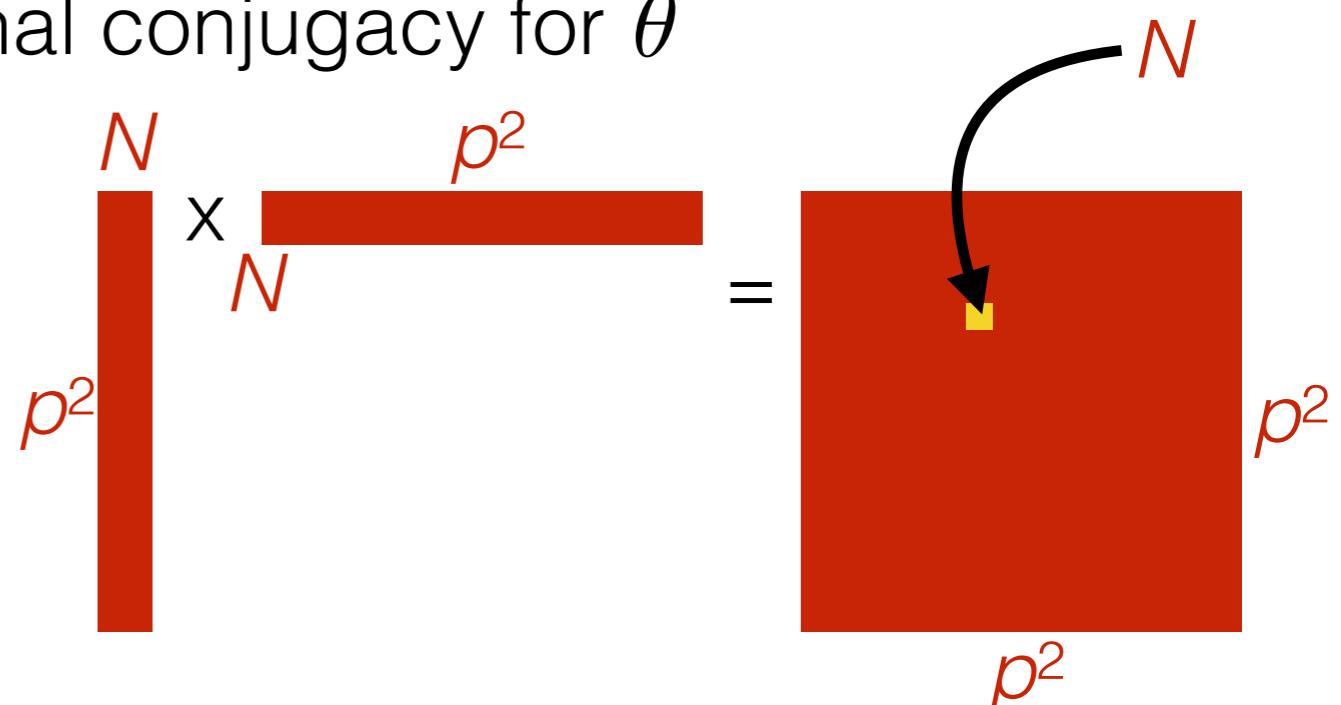
- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

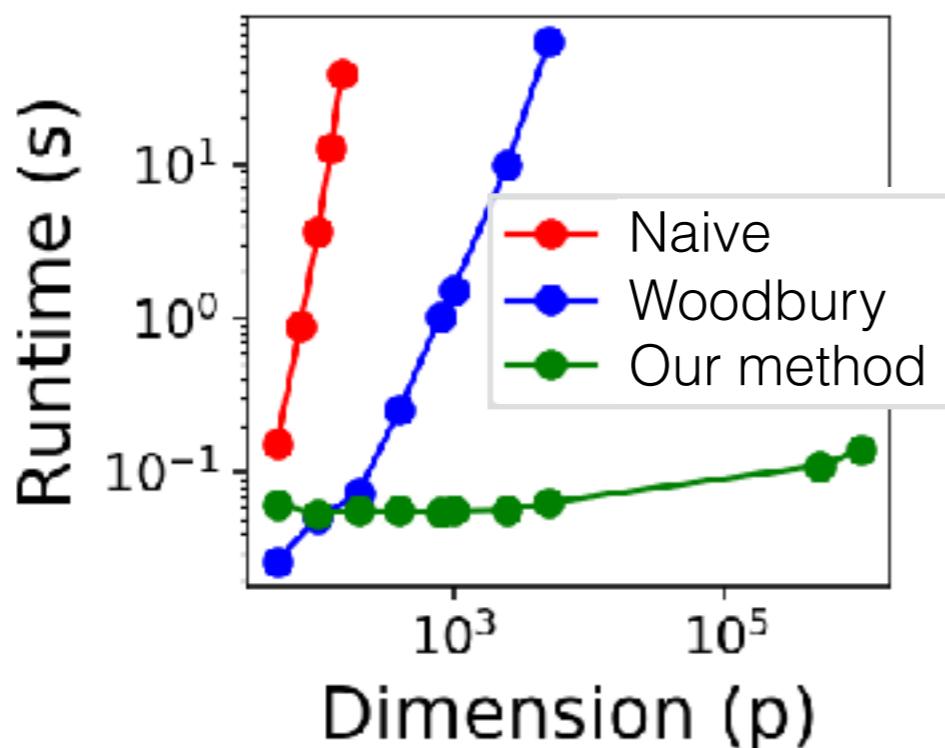
$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$



- Naive time cost: $O(p^4N + p^6)$
 - Woodbury time cost: $O(p^2N^2 + N^3)$



Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for θ

- Compute and invert

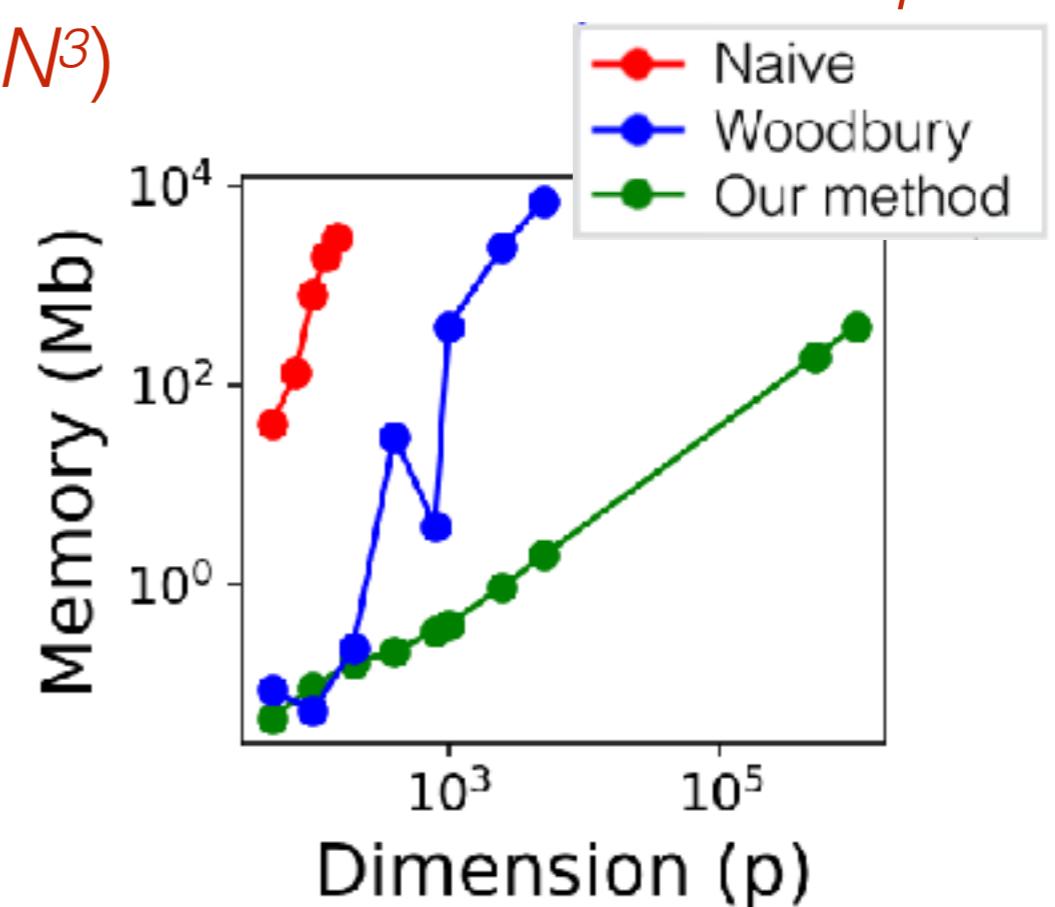
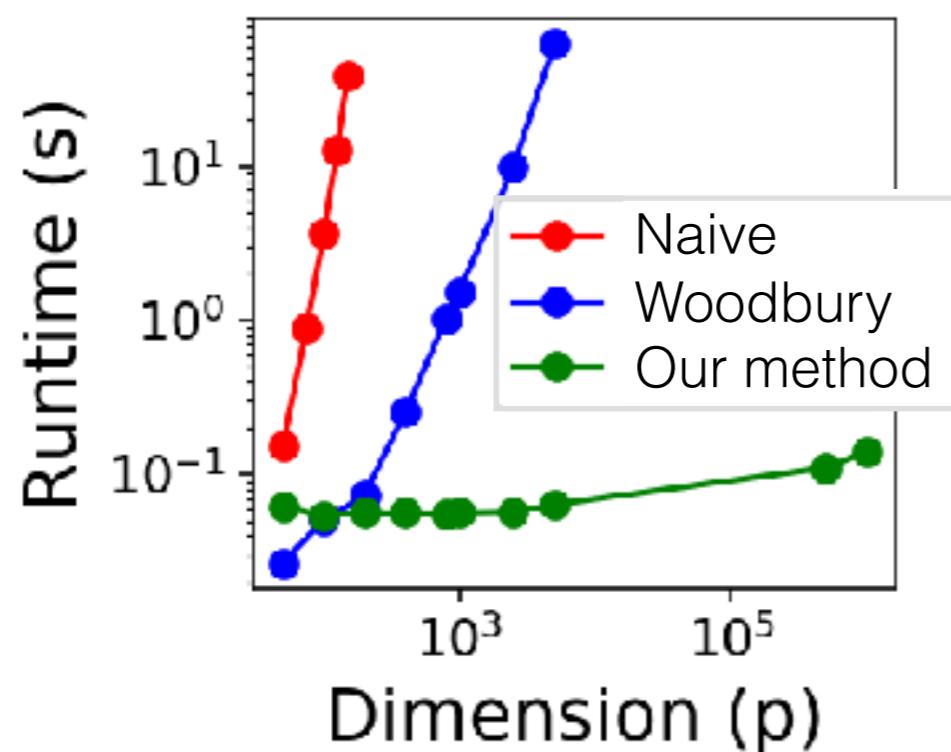
$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$

$$N \times \frac{p^2}{N} = p^2$$

- Naive time cost: $O(p^4N + p^6)$
 - Woodbury time cost: $O(p^2N^2 + N^3)$



Kernel Interaction Sampler vs. Naive MCMC

- Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Compute and invert
 ~~$\Phi_2(X)^\top \Phi_2(X)$~~

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

use conditional conjugacy for $\theta^T \Phi_2(X)$

- Compute and invert

~~$\Phi_2(X)^\top \Phi_2(X)$~~

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

~~$\Phi_2(X)^\top \Phi_2(X)$~~

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X)^\top \Phi_2(X)$$

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X)\Phi_2(X)^\top\Phi_2(X)\Phi_2(X)^\top$$

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X) \Phi_2(X)^\top \Phi_2(X) \Phi_2(X)^\top$$

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X)\Phi_2(X)^\top \Phi_2(X)\Phi_2(X)^\top$$

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X)\Phi_2(X)^\top$$

$X: N \times p$

$\Phi_2: N \times p^2$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
$$\Phi_2(X)\Phi_2(X)^\top$$
$$X: N \times p$$
$$\Phi_2: N \times p^2$$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X)\Phi_2(X)^\top$$

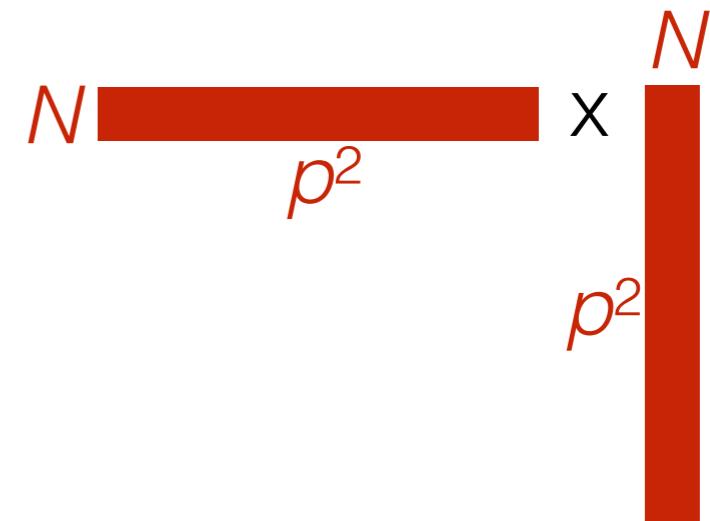
$$X: N \times p$$

$$\Phi_2: N \times p^2$$

$$N \overline{p^2}$$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
 $\Phi_2(X)\Phi_2(X)^\top$
 $X: N \times p$
 $\Phi_2: N \times p^2$



Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X)\Phi_2(X)^\top$$

$X: N \times p$

$\Phi_2: N \times p^2$

$$N \begin{matrix} \textcolor{red}{p^2} \\ \times \end{matrix} = \begin{matrix} \textcolor{red}{N} \\ \textcolor{red}{p^2} \end{matrix}$$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X) \Phi_2(X)^T$$

$X: N \times p$

$\Phi_2: N \times p^2$

$$N \begin{array}{c} \\[-1ex] p^2 \end{array} \times \begin{array}{c} N \\[-1ex] p^2 \end{array} = \begin{array}{c} N \\[-1ex] N \end{array}$$

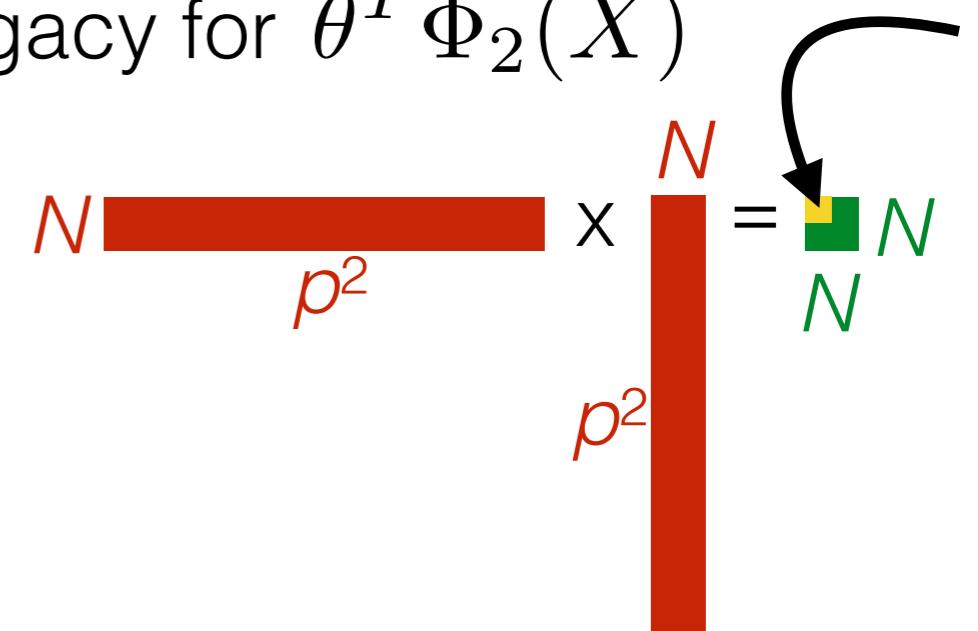
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert

$$\Phi_2(X) \Phi_2(X)^T$$

$X: N \times p$

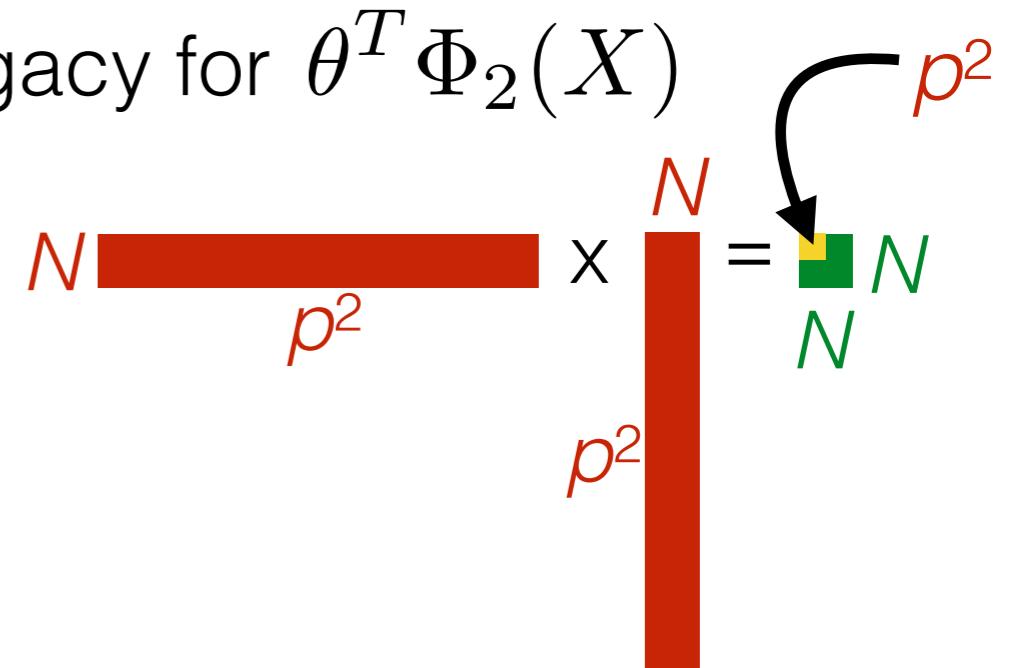
$\Phi_2: N \times p^2$

$$N \xrightarrow{p^2} \times \begin{matrix} N \\ p^2 \end{matrix} = \begin{matrix} N \\ N \end{matrix}$$


A diagram illustrating matrix multiplication. On the left, a horizontal red bar is labeled $N \xrightarrow{p^2}$. To its right is a times sign (\times). To the right of the times sign is a vertical red bar, also labeled $N \xrightarrow{p^2}$. An arrow points from the right side of the horizontal bar to the left side of the vertical bar. To the right of the vertical bar is an equals sign ($=$). To the right of the equals sign is a green square containing the text $\begin{matrix} N \\ N \end{matrix}$.

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
 $\Phi_2(X)\Phi_2(X)^\top$
- $X: N \times p$
 $\Phi_2: N \times p^2$

$$N \begin{matrix} p^2 \\ \hline \end{matrix} \times \begin{matrix} N \\ p^2 \\ \hline \end{matrix} = \begin{matrix} N \\ \text{---} \\ \text{---} \end{matrix}$$


Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
$$\Phi_2(X) \Phi_2(X)^T$$
$$X: N \times p$$
$$\Phi_2: N \times p^2$$
 - Kernel trick: $O(p)$ cost

$$N \xrightarrow{p^2} \times \begin{matrix} N \\ p^2 \end{matrix} = \begin{matrix} N \\ N \end{matrix}$$

A diagram illustrating matrix multiplication. On the left, a red horizontal bar is labeled $N \xrightarrow{p^2}$. To its right is a times sign (\times). To the right of the times sign is a red vertical bar labeled $N \xrightarrow{p^2}$. To the right of the vertical bar is an equals sign ($=$). To the right of the equals sign is a green square divided into four quadrants, with the top-right quadrant labeled N/N . A black arrow points from the top-right quadrant of the green square to the rightmost N in the label N/N .

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
 $\Phi_2(X)\Phi_2(X)^\top$
 $X: N \times p$
 $\Phi_2: N \times p^2$
 - Kernel trick: $O(p)$ cost

The diagram shows a matrix multiplication operation. On the left, a red horizontal bar labeled N above and p^2 below is multiplied by a red vertical bar labeled N above and p^2 below. The result is a green square labeled N above and N below. A yellow box labeled p^2 is shown above the horizontal bar, and a black arrow points from it to the green square result.

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
$$\Phi_2(X)\Phi_2(X)^\top$$
$$X: N \times p$$
$$\Phi_2: N \times p^2$$
 - Kernel trick: $O(p)$ cost

The diagram shows a matrix multiplication operation. On the left, a horizontal red bar labeled N above and p^2 below is multiplied by a vertical red bar labeled N above and p^2 below. The result is a green square labeled N above and N below. A yellow arrow points from the top of the horizontal bar to a yellow box containing the letter p . Another yellow arrow points from the bottom of the vertical bar to a green box containing the letter N .

$$N \begin{matrix} p^2 \\ \times \end{matrix} = \begin{matrix} N \\ N \end{matrix}$$

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
 $\Phi_2(X)\Phi_2(X)^\top$
 $X: N \times p$
 $\Phi_2: N \times p^2$
 - Kernel trick: $O(p)$ cost

$$N \underset{p^2}{\textcolor{red}{\boxed{\quad}}} \times \underset{p^2}{\textcolor{red}{\boxed{\quad}}} = \underset{N}{\textcolor{green}{\boxed{N}}}$$

A diagram illustrating matrix multiplication. On the left, a red horizontal bar labeled N above and p^2 below is multiplied by a red vertical bar labeled N to its right, also with p^2 written vertically below it. The result is a green square labeled N . A black arrow points from the p^2 label on the horizontal bar to the p^2 label on the vertical bar. A green bracket labeled p is positioned above the horizontal bar.

Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
 - Compute and invert
 $\Phi_2(X)\Phi_2(X)^\top$
 $X: N \times p$
 $\Phi_2: N \times p^2$
 - Kernel trick: $O(p)$ cost
 - Our time cost: $O(\textcolor{red}{p}N^2 + N^3)$

$$\begin{array}{c} N \\ \textcolor{red}{p^2} \end{array} \times \begin{array}{c} N \\ \textcolor{red}{p^2} \end{array} = \begin{array}{c} \textcolor{red}{N} \\ \textcolor{green}{N} \end{array}$$

A diagram illustrating matrix multiplication. On the left, a horizontal red bar is labeled N above p^2 . To its right is a vertical red bar labeled N above p^2 . An equals sign follows. To the right of the equals sign is a vertical red bar labeled N above N , with a green square at the top.

Reporting: Kernel Interaction Trick

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of θ_{x_i} or $\theta_{x_i x_j}$ in $O(1)$ time

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of θ_{x_i} or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \dots, 0, \underset{\substack{\text{arrow} \\ i^{\text{th}} \text{ position}}}{1}, 0, \dots, 0]$$

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of θ_{x_i} or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \dots, 0, \underset{\substack{\text{→} \\ i^{\text{th}} \text{ position}}}{1}, 0, \dots, 0] \quad g(e_i) = \theta_{x_i} + \theta_{x_i^2}$$

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of θ_{x_i} or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \dots, 0, \underset{\substack{\text{arrow} \\ i^{\text{th}} \text{ position}}}{1}, 0, \dots, 0]$$

$$g(e_i) = \theta_{x_i} + \theta_{x_i^2}$$

$$g(-e_i) = -\theta_{x_i} + \theta_{x_i^2}$$

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of θ_{x_i} or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \dots, 0, \underset{i^{\text{th}} \text{ position}}{\overset{\blacktriangleleft}{1}}, 0, \dots, 0]$$

$$\begin{aligned} g(e_i) &= \theta_{x_i} + \theta_{x_i^2} \\ g(-e_i) &= -\theta_{x_i} + \theta_{x_i^2} \\ \frac{g(e_i) - g(-e_i)}{2} &= \theta_{x_i} \end{aligned}$$

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of θ_{x_i} or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \dots, 0, \underset{i^{\text{th}} \text{ position}}{\overset{\blacktriangleleft}{1}}, 0, \dots, 0]$$

$$\begin{aligned} g(e_i) &= \theta_{x_i} + \theta_{x_i^2} \\ g(-e_i) &= -\theta_{x_i} + \theta_{x_i^2} \\ \frac{g(e_i) - g(-e_i)}{2} &= \theta_{x_i} \end{aligned}$$

- Step B: Find $k \ll p$ sparse main effects: takes $O(p)$ time

Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of θ_{x_i} or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \dots, 0, \underset{i^{\text{th}} \text{ position}}{\overset{\blacktriangleleft}{1}}, 0, \dots, 0]$$

$$\begin{aligned} g(e_i) &= \theta_{x_i} + \theta_{x_i^2} \\ g(-e_i) &= -\theta_{x_i} + \theta_{x_i^2} \\ \frac{g(e_i) - g(-e_i)}{2} &= \theta_{x_i} \end{aligned}$$

- Step B: Find $k \ll p$ sparse main effects: takes $O(p)$ time
- Step C: Report just the k^2 strong-hierarchy interaction effects: takes $O(k^2)$ time

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference
 - Fast reporting of results
- Experiments on simulated and real data

Roadmap

- Setup: Discovering main and interaction effects
- Our method
 - A Bayesian generative model
 - Fast inference
 - Fast reporting of results
- Experiments on simulated and real data

Timing vs. LASSO-based methods

Timing vs. LASSO-based methods

- LASSO (pairs, hierarchical): $O(p^2)$ per iteration

Timing vs. LASSO-based methods

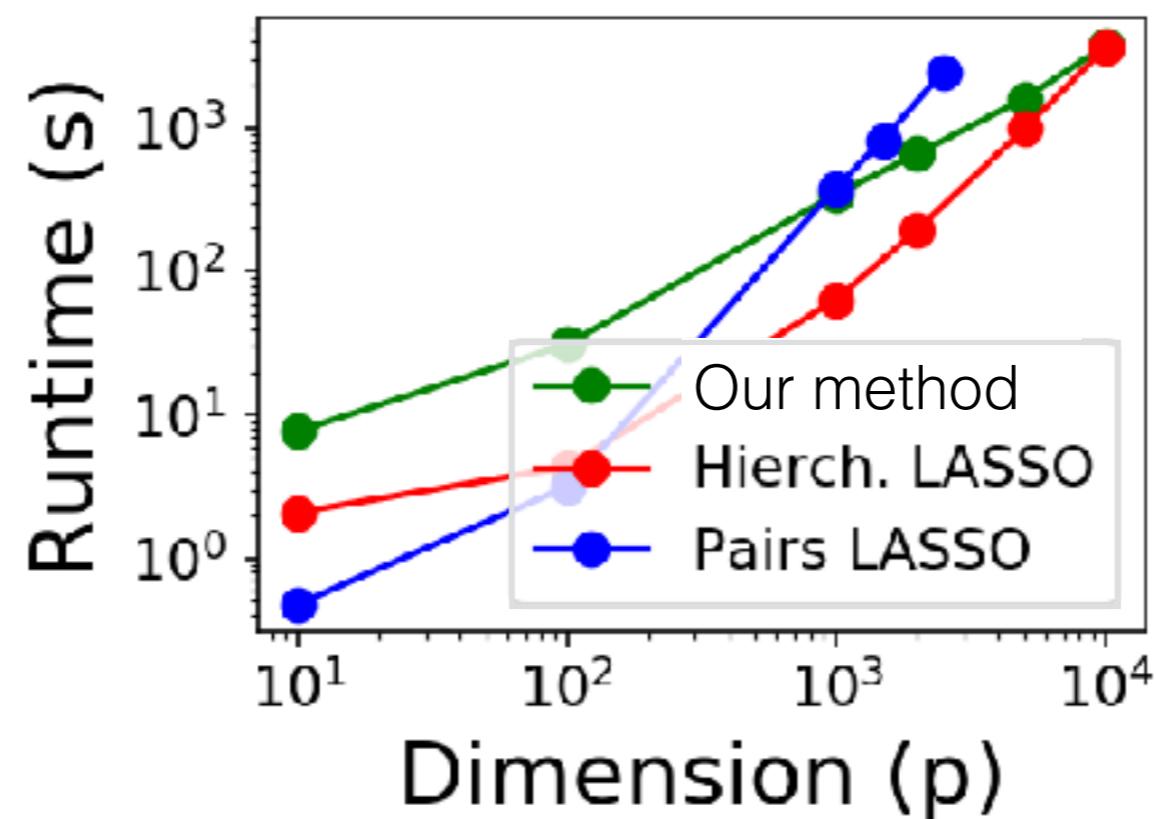
- LASSO (pairs, hierarchical): $O(p^2)$ per iteration [Lim, Hastie 2015]

Timing vs. LASSO-based methods

- LASSO (pairs, hierarchical): $O(p^2)$ per iteration [Lim, Hastie 2015]
- Our method: $O(p)$ per iteration

Timing vs. LASSO-based methods

- LASSO (pairs, hierarchical): $O(p^2)$ per iteration [Lim, Hastie 2015]
- Our method: $O(p)$ per iteration
- Competitive empirically for moderate p :



Experiments: Simulated

Experiments: Simulated

- 36 different simulated data sets (so know true effects)

Experiments: Simulated

- 36 different simulated data sets (so know true effects)
 - Up to $p = 500 \rightarrow \approx 125,000$ total parameters

Experiments: Simulated, Selection

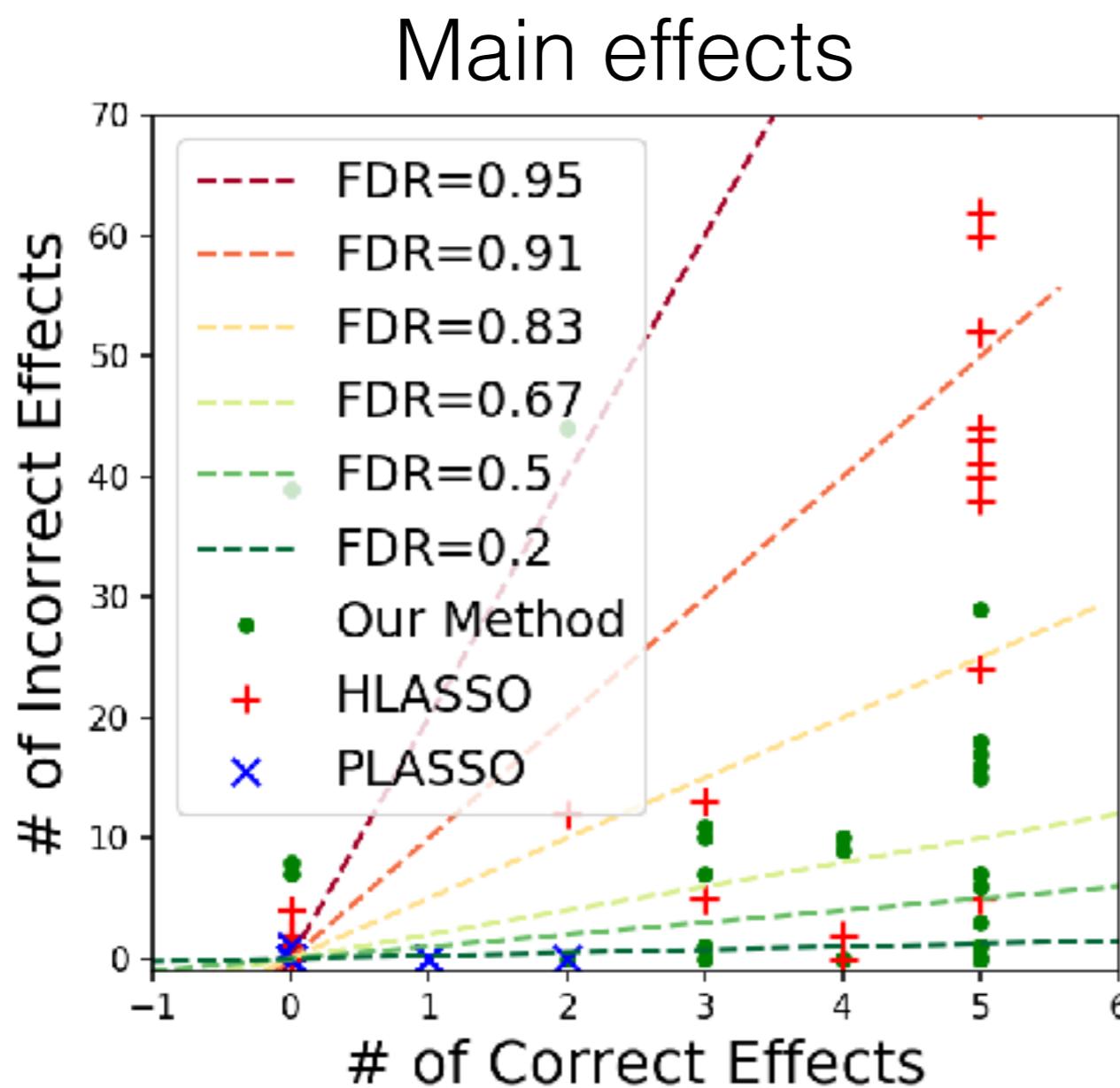
- 36 different simulated data sets (so know true effects)
 - Up to $p = 500 \rightarrow \approx 125,000$ total parameters

Experiments: Simulated, Selection

- 36 different simulated data sets (so know true effects)
 - Up to $p = 500 \rightarrow \approx 125,000$ total parameters
 - False discovery rate (FDR): proportion incorrect

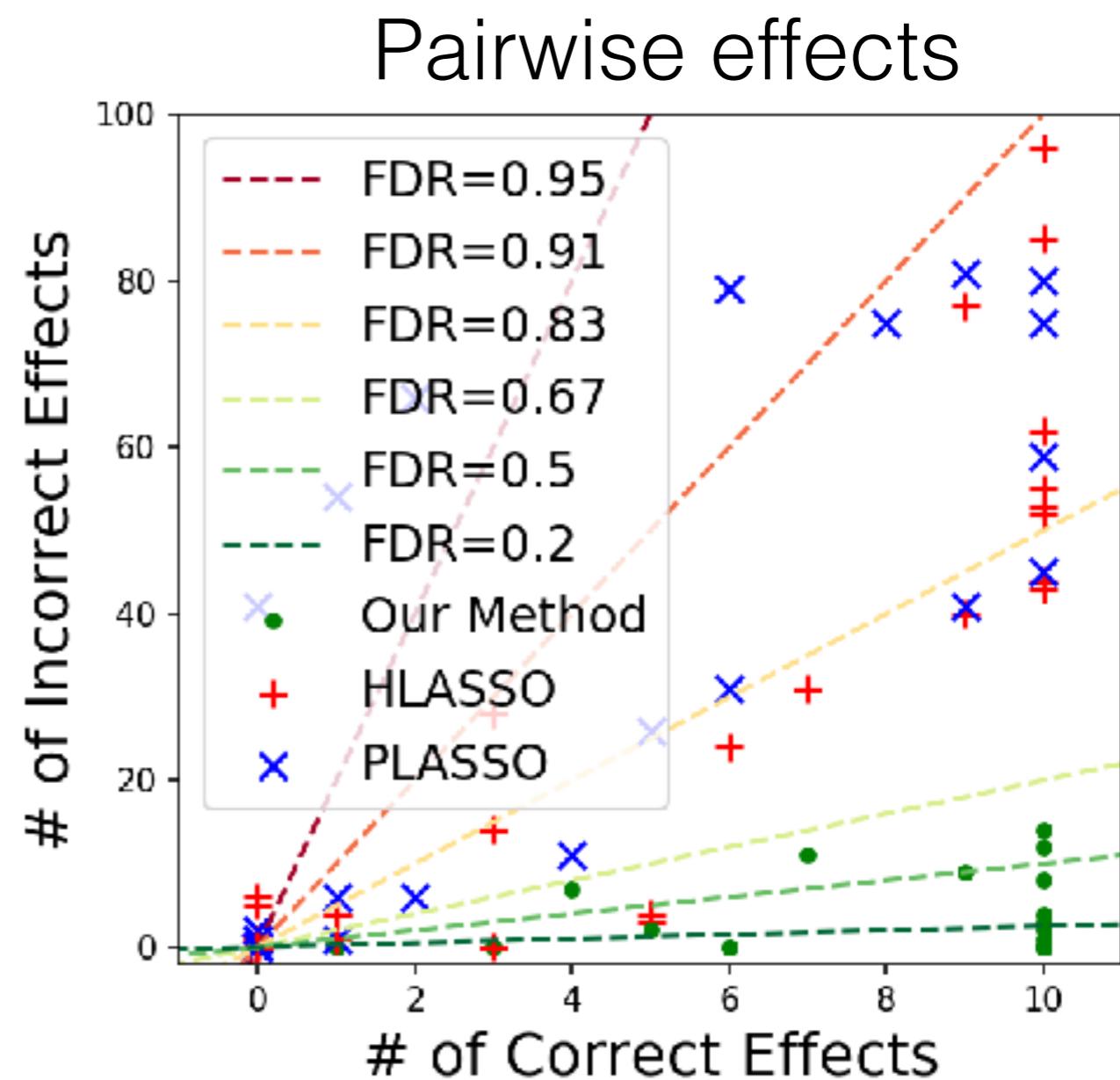
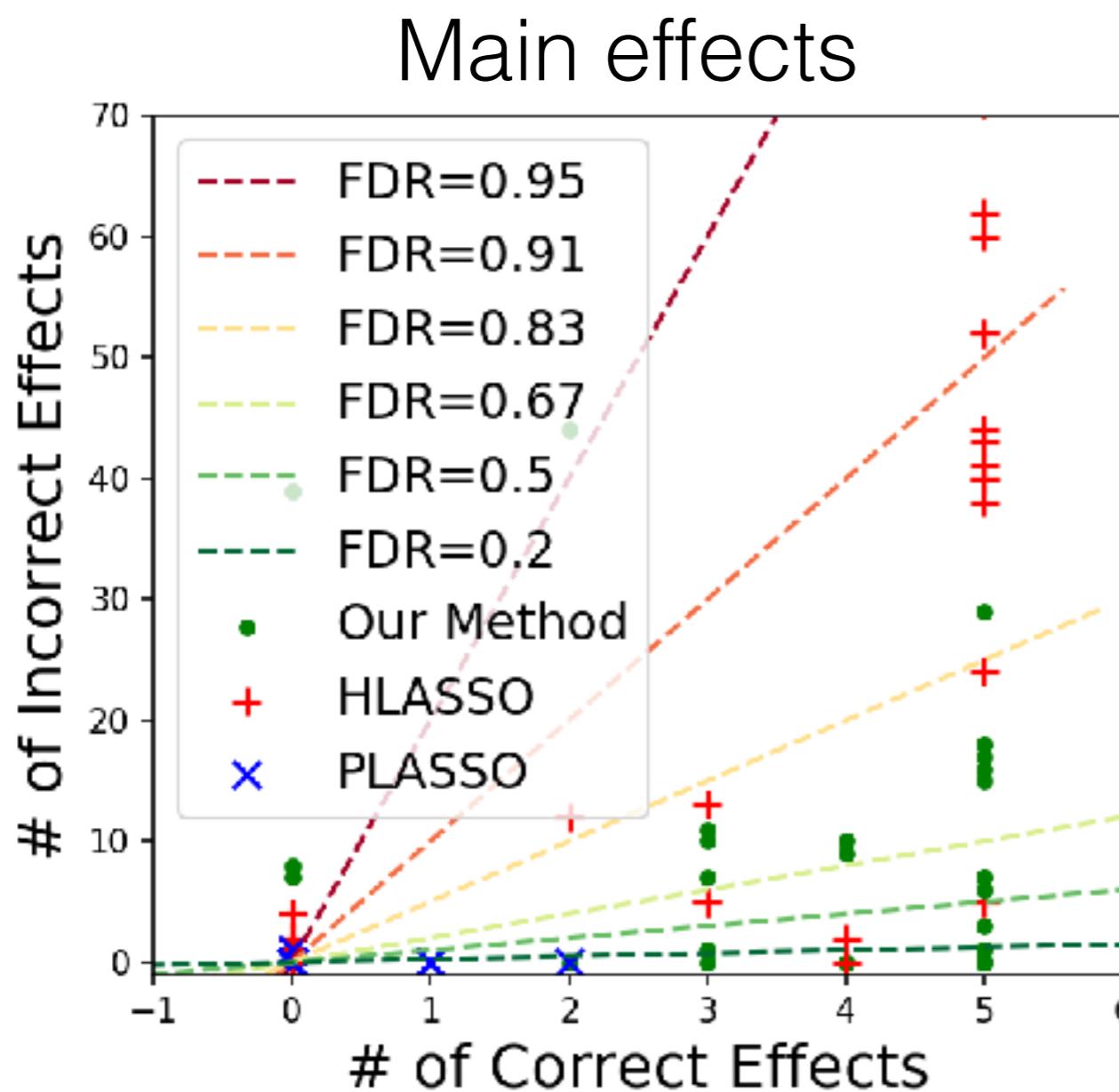
Experiments: Simulated, Selection

- 36 different simulated data sets (so know true effects)
 - Up to $p = 500 \rightarrow \approx 125,000$ total parameters
- False discovery rate (FDR): proportion incorrect



Experiments: Simulated, Selection

- 36 different simulated data sets (so know true effects)
 - Up to $p = 500 \rightarrow \approx 125,000$ total parameters
- False discovery rate (FDR): proportion incorrect



Experiments: Real covariates

Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction

Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
 - Highly correlated: 20 of 105 capture 99% of variance

Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
 - Highly correlated: 20 of 105 capture 99% of variance
- Key: (# correct effects): (# of incorrect effects)
 - **Higher** green is better: **lower** red is better

Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
 - Highly correlated: 20 of 105 capture 99% of variance
- Key: (# correct effects): (# of incorrect effects)
 - **Higher** green is better: **lower** red is better

METHOD	#MAIN	#PAIR
PLASSO	2 : 5	3 : 21

Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
 - Highly correlated: 20 of 105 capture 99% of variance
- Key: (# correct effects): (# of incorrect effects)
 - **Higher** green is better: **lower** red is better

METHOD	#MAIN	#PAIR
PLASSO	2 : 5	3 : 21
HLASSO	3 : 19	3 : 18

Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
 - Highly correlated: 20 of 105 capture 99% of variance
- Key: (# correct effects): (# of incorrect effects)
 - **Higher** green is better: **lower** red is better

METHOD	#MAIN	#PAIR
Our method	3 : 0	3 : 0
PLASSO	2 : 5	3 : 21
HLASSO	3 : 19	3 : 18

Experiments: Real data

Experiments: Real data

- Covariates and response: Auto MPG

Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398$, $p = 6$ (real-valued), but...

Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398$, $p = 6$ (real-valued), but...
- Augment p with 200 fake (noise) covariates
 - 21,321 total parameters

Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398$, $p = 6$ (real-valued), but...
- Augment p with 200 fake (noise) covariates
 - 21,321 total parameters
- Key: (# original effects): (# of fake effects)
 - **No order** to blue: **lower** red is better

Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398$, $p = 6$ (real-valued), but...
- Augment p with 200 fake (noise) covariates
 - 21,321 total parameters
- Key: (# original effects): (# of fake effects)
 - **No order** to blue: **lower** red is better

METHOD	#MAIN	#PAIR
PLASSO	4 : 0	2 : 78

Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398$, $p = 6$ (real-valued), but...
- Augment p with 200 fake (noise) covariates
 - 21,321 total parameters
- Key: (# original effects): (# of fake effects)
 - **No order** to blue: **lower** red is better

METHOD	#MAIN	#PAIR
PLASSO	4 : 0	2 : 78
HLASSO	6 : 46	4 : 38

Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398$, $p = 6$ (real-valued), but...
- Augment p with 200 fake (noise) covariates
 - 21,321 total parameters
- Key: (# original effects): (# of fake effects)
 - **No order** to blue: **lower** red is better

METHOD	#MAIN	#PAIR
Our method	3 : 0	1 : 0
PLASSO	4 : 0	2 : 78
HLASSO	6 : 46	4 : 38

Conclusions

We provide: fast, accurate detection of pairwise interactions

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. *ICML* 2019. ArXiv:1905.06501

Conclusions

We provide: fast, accurate detection of pairwise interactions

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. ICML 2019. ArXiv:1905.06501

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

- In Pyro: http://pyro.ai/numppyro/sparse_regression.html

Conclusions

We provide: fast, accurate detection of pairwise
(and higher-order) interactions

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. ICML 2019. ArXiv:1905.06501

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

- In Pyro: http://pyro.ai/numppyro/sparse_regression.html

Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. ICML 2019. ArXiv:1905.06501

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

- In Pyro: http://pyro.ai/numppyro/sparse_regression.html

Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:

- Response types (binary, count, etc) & nonlinearity

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. ICML 2019. ArXiv:1905.06501

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

- In Pyro: http://pyro.ai/numppyro/sparse_regression.html

Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:

- Response types (binary, count, etc) & nonlinearity
- Improve scaling in N

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. ICML 2019. ArXiv:1905.06501

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

- In Pyro: http://pyro.ai/numppyro/sparse_regression.html

Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:

- Response types (binary, count, etc) & nonlinearity
- Improve scaling in N

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. ICML 2019. ArXiv:1905.06501

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

- In Pyro: http://pyro.ai/numppyro/sparse_regression.html

JH Huggins, T Campbell, M Kasprzak, and T Broderick. Scalable Gaussian process inference with finite-data mean and variance guarantees. AISTATS 2019.

R Agrawal, T Campbell, JH Huggins, and T Broderick. Data-dependent compression of random features for large-scale kernel approximation. AISTATS 2019.

Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:

- Response types (binary, count, etc) & nonlinearity
- Improve scaling in N
- Applications!

R Agrawal, BL Trippe, JH Huggins, and T Broderick. The Kernel interaction trick: Fast Bayesian discovery of pairwise interactions in high dimensions. ICML 2019. ArXiv:1905.06501

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

- In Pyro: http://pyro.ai/numppyro/sparse_regression.html

JH Huggins, T Campbell, M Kasprzak, and T Broderick. Scalable Gaussian process inference with finite-data mean and variance guarantees. AISTATS 2019.

R Agrawal, T Campbell, JH Huggins, and T Broderick. Data-dependent compression of random features for large-scale kernel approximation. AISTATS 2019.

More in the Broderick Group

L Masoero, F Camerlenghi, S Favaro, T Broderick. More for less: Predicting and maximizing variant discovery under a fixed budget via Bayesian nonparametrics. <https://arxiv.org/abs/1912.05516>

- For fixed budget, there is trade-off in sequencing more genomes and sequencing at greater depth
- We provide new method for prediction of # new variants and optimal allocation of more genomes vs. depth
 - Lowest error when using pilot TCGA dataset to predict the number of new variants to be observed in the follow-up MSK-impact dataset ($N=9593$) across 197 highly variable, cancerous genes
 - (Only) our prediction can handle when sequencing depth changes between pilot and follow-up study
 - (Only) our method optimizes under fixed budget

T Broderick, R Giordano, R Meager. An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Change Conclusions? In preparation.