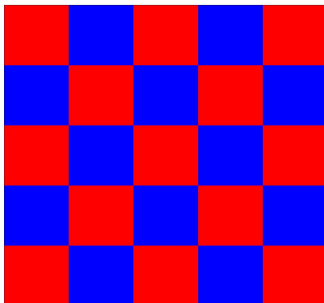


1/24

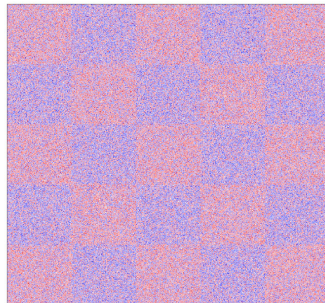
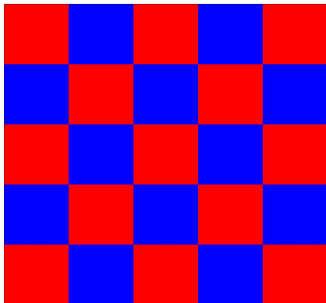
Problématique

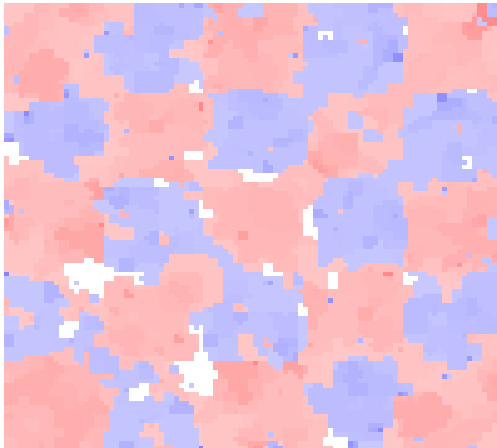
```
n <- 500
K <- 5
mu <- suppressWarnings(matrix(rep(c(1,0),ceiling(K**2/2)), K,K))
image(mu[K:1,1:K],col=mypalette,xaxt="n",yaxt="n")
require(blockseg)
output <- rblockdata(n,mu,sigma=2)
image(output$Y[n:1,1:n],col=mypalette,xaxt="n",yaxt="n")
```



Problématique

```
n <- 500
K <- 5
mu <- suppressWarnings(matrix(rep(c(1,0),ceiling(K**2/2)), K,K))
image(mu[K:1,1:K],col=mypalette,xaxt="n",yaxt="n")
require(blockseg)
output <- rblockdata(n,mu,sigma=2)
image(output$Y[n:1,1:n],col=mypalette,xaxt="n",yaxt="n")
```





Plan

1 Model and method

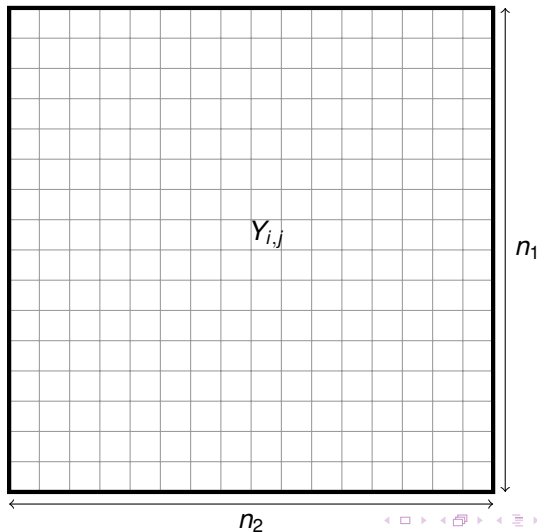
2 Practical use

Plan

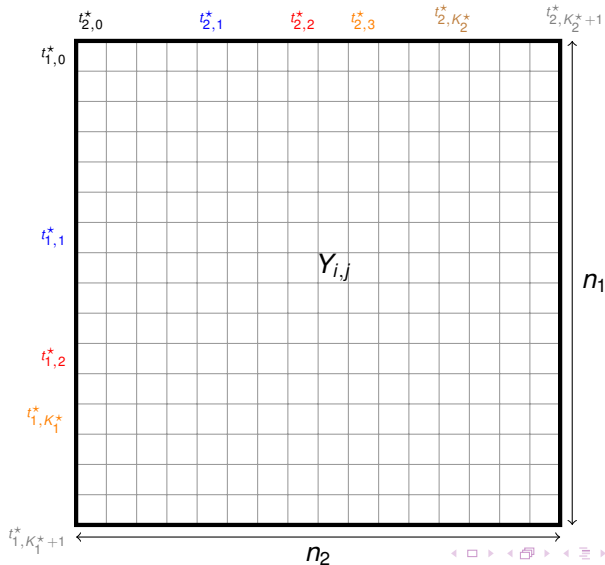
1 Model and method

2 Practical use

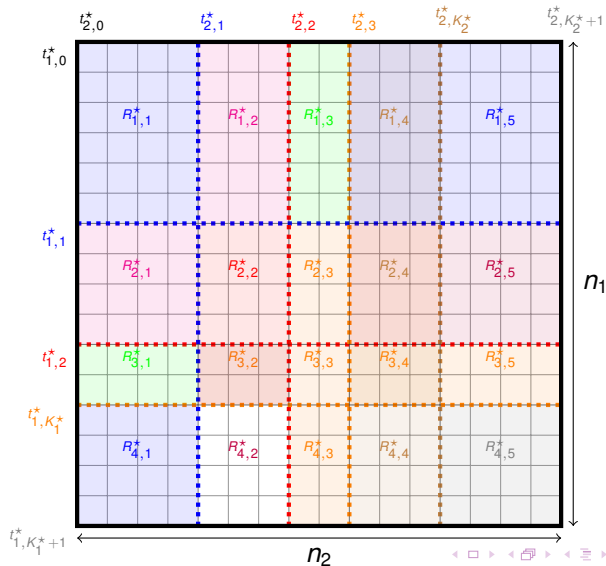
Notations



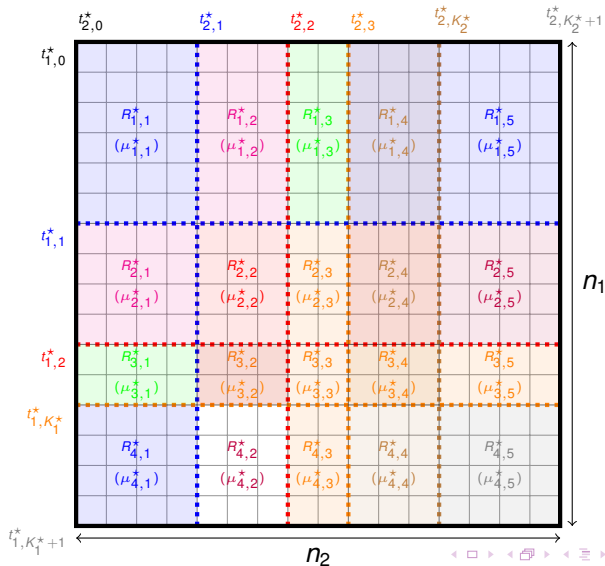
Notations



Notations



Notations



Model

Let $Y = (Y_{i,j})_{1 \leq i,j \leq n}$ be the random matrix defined by

$$\mathbf{Y} = \mathbf{U} + \mathbf{E},$$

where $\mathbf{U} = (U_{i,j})$ is a blockwise constant matrix such that

$$U_{i,j} = \mu_{k,\ell}^* \quad \text{if } t_{1,k-1}^* \leq i \leq t_{1,k}^* - 1 \text{ and } t_{2,\ell-1}^* \leq j \leq t_{2,\ell}^* - 1,$$

with the convention $t_{1,0}^* = t_{2,0}^* = 1$ and $t_{1,K_1^*+1}^* = t_{2,K_2^*+1}^* = n + 1$.

The entries $E_{i,j}$ of the matrix $E = (E_{i,j})_{1 \leq i,j \leq n}$ are iid zero-mean random variables.

Trick

$$\mathbf{U} = \mathbf{T}\mathbf{B}\mathbf{T}^\top$$

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix}$$

and

$$\mathbf{B} = \begin{matrix} & t_{2,0}^* & t_{2,1}^* & t_{2,2}^* & t_{2,3}^* & t_{2,K_2}^* & t_{2,K_2+1}^* \\ \begin{matrix} t_{1,0}^* \\ t_{1,1}^* \\ t_{1,2}^* \\ t_{1,K_1}^* \\ t_{1,K_1+1}^* \end{matrix} & \begin{matrix} B_{1,1} & 0 & 0 & 0 & B_{1,5} & 0 & B_{1,8} & 0 & B_{1,10} & 0 & B_{1,13} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{matrix}$$

$$\text{with } \mu_{k,\ell}^* = \sum_{i=1}^k \sum_{j=1}^{\ell} B_{t_{1,i}^*, t_{2,j}^*}$$

Vectorisation

$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{T}^\top + \mathbf{E}$$

is equivalent to

$$\text{Vec}(\mathbf{Y}) = \text{Vec}(\mathbf{T}\mathbf{B}\mathbf{T}^\top) + \text{Vec}(\mathbf{E})$$

with

$$\text{Vec}(\mathbf{T}\mathbf{B}\mathbf{T}^\top) = (\mathbf{T}^{\top\top} \otimes \mathbf{T}) \text{Vec}(\mathbf{B}) = (\mathbf{T} \otimes \mathbf{T}) \text{Vec}(\mathbf{B})$$

and we obtain

$$\underbrace{\mathcal{Y}}_{n^2 \times 1} = \underbrace{\mathcal{X}}_{n^2 \times n^2} \underbrace{\mathcal{B}}_{n^2 \times 1} + \underbrace{\mathcal{E}}_{n^2 \times 1}.$$

Kronecker product

Vectorisation

$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{T}^\top + \mathbf{E}$$

is equivalent to

$$\text{Vec}(\mathbf{Y}) = \text{Vec}(\mathbf{T}\mathbf{B}\mathbf{T}^\top) + \text{Vec}(\mathbf{E})$$

with

$$\text{Vec}(\mathbf{T}\mathbf{B}\mathbf{T}^\top) = \left(\mathbf{T}^{\top\top} \otimes \mathbf{T} \right) \text{Vec}(\mathbf{B}) = (\mathbf{T} \otimes \mathbf{T}) \text{Vec}(\mathbf{B})$$

and we obtain

$$\underbrace{\mathcal{Y}}_{n^2 \times 1} = \underbrace{\mathcal{X}}_{n^2 \times n^2} \underbrace{\mathcal{B}}_{n^2 \times 1} + \underbrace{\mathcal{E}}_{n^2 \times 1}.$$

Kronecker product

Vectorisation

$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{T}^\top + \mathbf{E}$$

is equivalent to

$$\text{Vec}(\mathbf{Y}) = \text{Vec}(\mathbf{T}\mathbf{B}\mathbf{T}^\top) + \text{Vec}(\mathbf{E})$$

with

$$\text{Vec}(\mathbf{T}\mathbf{B}\mathbf{T}^\top) = \left(\mathbf{T}^{\top\top} \otimes \mathbf{T} \right) \text{Vec}(\mathbf{B}) = (\mathbf{T} \otimes \mathbf{T}) \text{Vec}(\mathbf{B})$$

and we obtain

$$\underbrace{\mathcal{Y}}_{n^2 \times 1} = \underbrace{\mathcal{X}}_{n^2 \times n^2} \underbrace{\mathcal{B}}_{n^2 \times 1} + \underbrace{\mathcal{E}}_{n^2 \times 1}.$$

Kronecker product

Least Absolute Shrinkage and Selection Operator (LASSO)

For all $\lambda_n \geq 0$, we define

$$\hat{\mathcal{B}}(\lambda_n) = \underset{\mathcal{B} \in \mathbb{R}^{n^2}}{\operatorname{Argmin}} \left\{ \|\mathcal{Y} - \mathcal{X}\mathcal{B}\|_2^2 + \lambda_n \|\mathcal{B}\|_1 \right\}$$

and the active set

$$\hat{\mathcal{A}}(\lambda_n) = \left\{ j \in \{1, \dots, n^2\} : \hat{\mathcal{B}}_j(\lambda_n) \neq 0 \right\}$$

A reminder on the norm

Least Absolute Shrinkage and Selection Operator (LASSO)

For all $\lambda_n \geq 0$, we define

$$\hat{\mathcal{B}}(\mathbf{0}) = \underset{\mathcal{B} \in \mathbb{R}^{n^2}}{\operatorname{Argmin}} \left\{ \|\mathcal{Y} - \mathcal{X}\mathcal{B}\|_2^2 + \lambda_n \|\mathcal{B}\|_1 \right\}$$

and the active set

$$\hat{\mathcal{A}}(\mathbf{0}) = \left\{ j \in \{1, \dots, n^2\} : \hat{\mathcal{B}}_j(\lambda_n) \neq 0 \right\} \approx \{1, \dots, n^2\}$$

A reminder on the norm

Least Absolute Shrinkage and Selection Operator (LASSO)

For all $\lambda_n \geq 0$, we define

$$\hat{\mathcal{B}}(+\infty) = \underset{\mathcal{B} \in \mathbb{R}^{n^2}}{\operatorname{Argmin}} \left\{ \|\mathcal{Y} - \mathcal{X}\mathcal{B}\|_2^2 + \lambda_n \|\mathcal{B}\|_1 \right\}$$

and the active set

$$\hat{\mathcal{A}}(+\infty) = \left\{ j \in \{1, \dots, n^2\} : \hat{\mathcal{B}}_j(\lambda_n) \neq 0 \right\} = \emptyset$$

A reminder on the norm

Least Absolute Shrinkage and Selection Operator (LASSO)

For all $\lambda_n \geq 0$, we define

$$\hat{\mathcal{B}}(\lambda_n) = \underset{\mathcal{B} \in \mathbb{R}^{n^2}}{\operatorname{Argmin}} \left\{ \|\mathcal{Y} - \mathcal{X}\mathcal{B}\|_2^2 + \lambda_n \|\mathcal{B}\|_1 \right\}$$

and the active set

$$\hat{\mathcal{A}}(\lambda_n) = \left\{ j \in \{1, \dots, n^2\} : \hat{\mathcal{B}}_j(\lambda_n) \neq 0 \right\}$$

A reminder on the norm

Estimation of break change-point

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} \begin{matrix} q_a + 1 \\ r_a + 1 \end{matrix}$$

$\forall a \in \hat{\mathcal{A}}(\lambda_n)$, we define (q_a, r_a) as the Euclidean division of $(a - 1)$ by n , namely $(a - 1) = nq_a + r_a$ then

$$\hat{\mathbf{t}}_1 = (\hat{t}_{1,k})_{1 \leq k \leq |\hat{\mathcal{A}}_1(\lambda_n)|} \in \hat{\mathcal{A}}_1(\lambda_n) = \{r_a + 1 : a \in \hat{\mathcal{A}}(\lambda_n)\},$$

$$\hat{\mathbf{t}}_2 = (\hat{t}_{2,\ell})_{1 \leq \ell \leq |\hat{\mathcal{A}}_2(\lambda_n)|} \in \hat{\mathcal{A}}_2(\lambda_n) = \{q_a + 1 : a \in \hat{\mathcal{A}}(\lambda_n)\}$$

where $\hat{t}_{1,1} < \hat{t}_{1,2} < \dots < \hat{t}_{1,|\hat{\mathcal{A}}_1(\lambda_n)|}$,

and $\hat{t}_{2,1} < \hat{t}_{2,2} < \dots < \hat{t}_{2,|\hat{\mathcal{A}}_2(\lambda_n)|}$.

04

$2+1=3$ and $1+1=2$

01

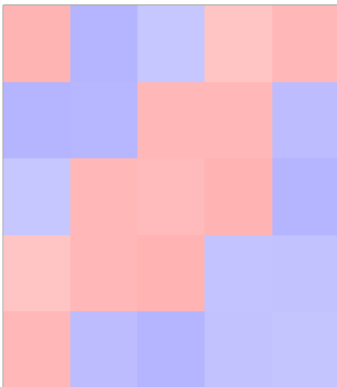
For example, $a=10$
 $10 - 1 = 9 = 4 \times 2 + 1$
 $2+1=3$ and $1+1=2$

$\forall \mathbf{a} \in \hat{\mathcal{A}}(\lambda_n)$, we define (q_a, r_a) as the Euclidean division of $(\mathbf{a} - 1)$ by n , namely $(\mathbf{a} - 1) = nq_a + r_a$ then

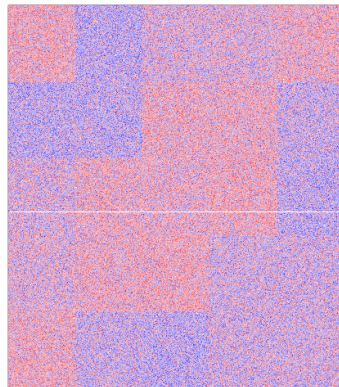
01

$2+1=3$ and $1+1=2$

Mu matrix



Original matrix



Gray version

Plan

1 Model and method

2 Practical use

```
res <- blockSeg(Y, max.break = floor(min(ncol(Y),  
nrow(Y))/10 + 1), max.var = floor(ncol(Y)^2/2),  
verbose = TRUE, Beta = FALSE)  
plot(res, Y, col="color")
```

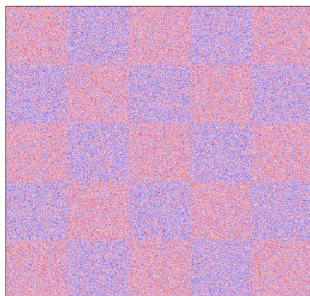
```
res <- blockSeg(Y, 50
                , max.var = floor(ncol(Y)^2/2),
verbose = TRUE, Beta = FALSE)
plot(res, Y, col="color")
```

```
res <- blockSeg(Y, 50
                , max.var = floor(ncol(Y)^2/2),
verbose = TRUE, Beta = FALSE)
plot(res,Y,col="color")
```

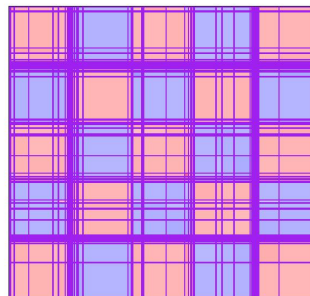
```
res <- blockSeg(Y, 50  
                , max.var = floor(ncol(Y)^2/2),  
verbose = TRUE, Beta = FALSE)  
plot(res,Y,col="color")
```

Lambda = 717.518 with (51,49) breaks along the (rows,columns).

Original data



Estimated matrix



Plot

```
lambda=res@Lambda[c(1:10)]  
plot(x , y, lambda = NULL ,ask = TRUE, col =  
"GrayLevel", ...)
```

Plot

```
lambda=res@Lambda[c(1:10)]  
plot(x , y, lambda = NULL ,ask = TRUE, col =  
"GrayLevel", ...)
```


Plot

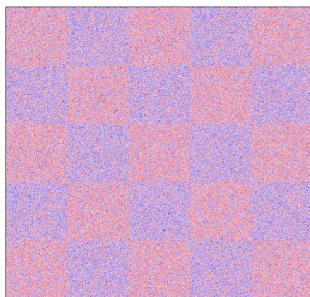
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color"      , ...)
```

Plot

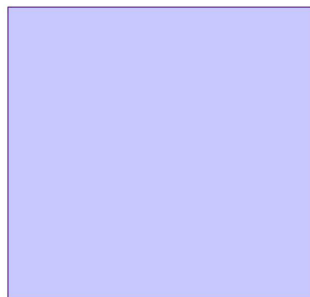
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 129072.352 with (0,0) breaks along the (rows,columns).

Original data



Estimated matrix

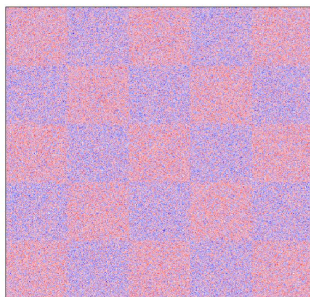


Plot

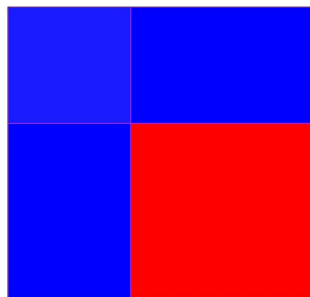
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 5921.327 with (1,1) breaks along the (rows,columns).

Original data



Estimated matrix

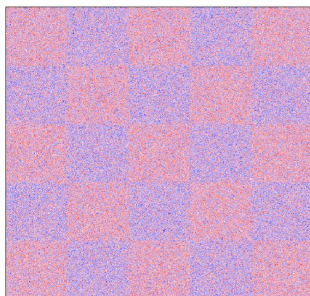


Plot

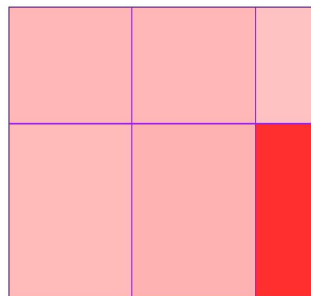
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 5052.901 with (2,2) breaks along the (rows,columns).

Original data



Estimated matrix

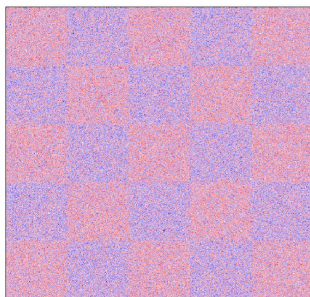


Plot

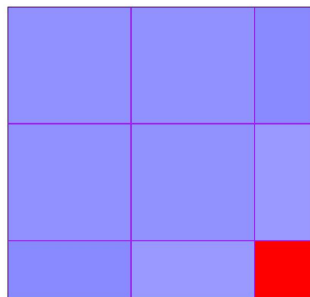
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 5052.901 with (3,3) breaks along the (rows,columns).

Original data



Estimated matrix

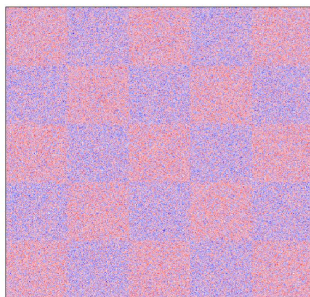


Plot

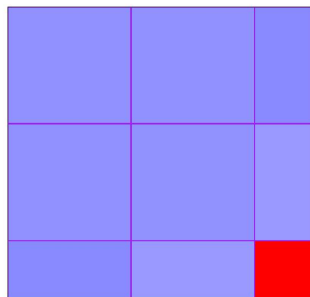
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 4669.223 with (3,3) breaks along the (rows,columns).

Original data



Estimated matrix

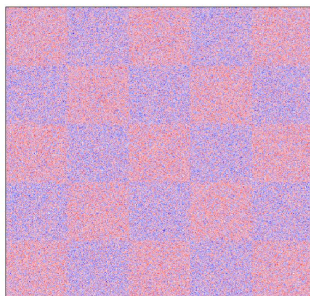


Plot

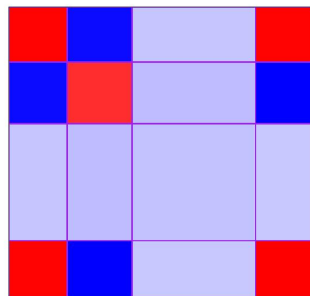
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 2772.482 with (4,4) breaks along the (rows,columns).

Original data



Estimated matrix

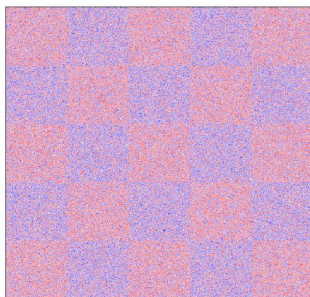


Plot

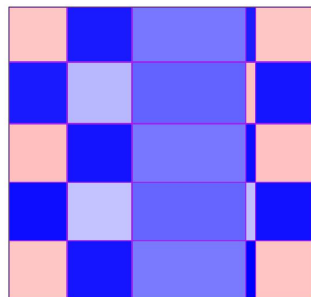
```
lambda=res@Lambda[c(1:10)]
plot(res, Y, lambda = lambda, ask = TRUE, col =
"Color" , ...)
```

Lambda = 2623.582 with (5,5) breaks along the (rows,columns).

Original data



Estimated matrix

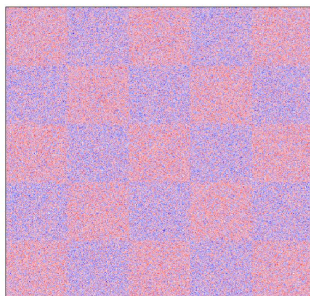


Plot

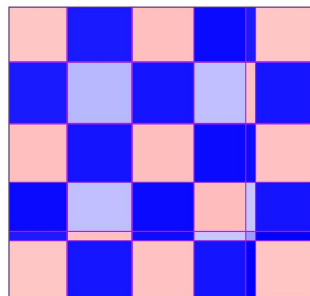
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 2623.582 with (6,6) breaks along the (rows,columns).

Original data



Estimated matrix

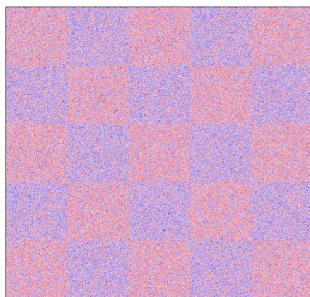


Plot

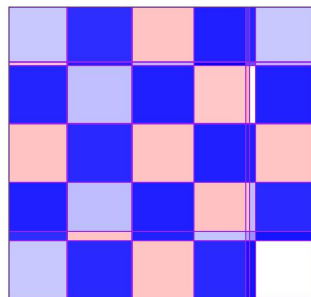
```
lambda=res@Lambda[c(1:10)]  
plot(res, Y, lambda = lambda, ask = TRUE, col =  
"Color", ...)
```

Lambda = 2611.41 with (7,7) breaks along the (rows,columns).

Original data



Estimated matrix

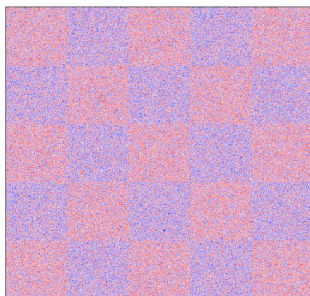


Plot

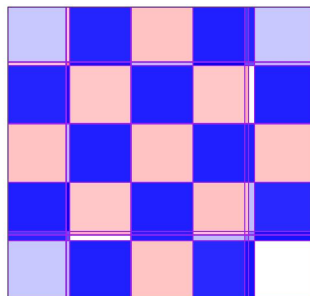
```
lambda=res@Lambda[c(1:10)]
plot(res, Y, lambda = lambda, ask = TRUE, col =
"Color" , ...)
```

Lambda = 2611.41 with (8,8) breaks along the (rows,columns).

Original data



Estimated matrix



Stability selection

Stability selection :

Input : data vector $\mathcal{Y} \in \mathcal{M}_{n^2 \times 1}$, an integer $M \in \mathbb{N}^*$, a pair of numbers $(K_1^*, K_2^*) \in \{1, \dots, n\}^2$.

For $iter \in \{1, \dots, M\}$

Chose randomly $ind^{(iter)} = \{i_1, \dots, i_{n^2/2}\} \subset \{1, \dots, n^2\}$.

Use the procedure with (K_1^*, K_2^*) change-points on the data

$\mathcal{Y}_{ind^{(iter)}}$ to obtain $(\hat{\mathbf{t}}_1^{(iter)}, \hat{\mathbf{t}}_2^{(iter)})$.

Output : Sequence of couples $(\hat{\mathbf{t}}_1^{(iter)}, \hat{\mathbf{t}}_2^{(iter)})$ recorded at each iteration or only the couple of change-points appearing a number of times larger than a given threshold.

Adaptation

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix}$$

Adaptation

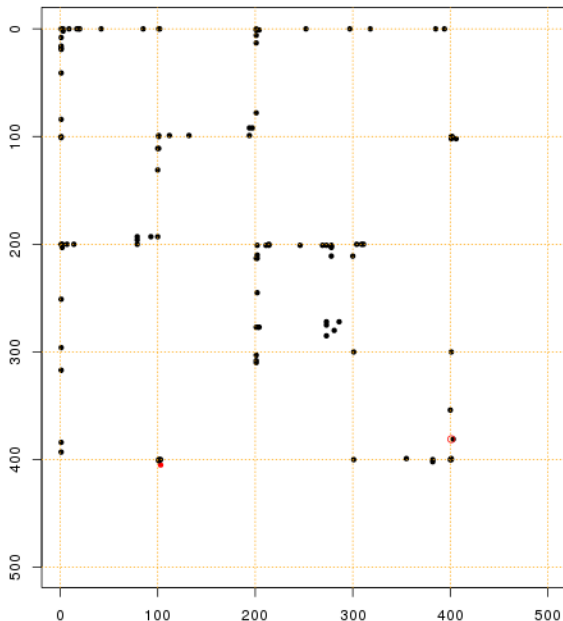
$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ \cdot \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & \cdot & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix}$$

Adaptation

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 13 \\ 14 \\ 15 \\ 16 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 5 & \cdot & 13 \\ 2 & 6 & \cdot & 14 \\ 3 & 7 & \cdot & 15 \\ 4 & 8 & \cdot & 16 \end{pmatrix}$$

Adaptation

$$\begin{pmatrix} 1 \\ 2 \\ \cdot \\ 4 \\ 5 \\ 6 \\ \cdot \\ 8 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 13 \\ 14 \\ \cdot \\ 16 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & 5 & \cdot & 13 \\ 2 & 6 & \cdot & 14 \\ \cdot & \cdot & \cdot & \cdot \\ 4 & 8 & \cdot & 16 \end{pmatrix}$$



Adaptation stability selection

Stability selection :

Input : data matrix $Y \in \mathcal{M}_{n \times n}$, an integer $M \in \mathbb{N}^*$, a pair of numbers $(K_1^*, K_2^*) \in \{1, \dots, n\}^2$.

For $iter \in \{1, \dots, M\}$

Choose randomly $ind_1^{(iter)} = \{i_1^{(1)}, \dots, i_{n/2}^{(1)}\} \subset \{1, \dots, n\}$ and $ind_2^{(iter)} = \{i_1^{(2)}, \dots, i_{n/2}^{(2)}\} \subset \{1, \dots, n\}$.

Use the procedure with (K_1^*, K_2^*) change-points on the data $Y_{ind_1^{(iter)}, ind_2^{(iter)}}$ to obtain $(N_1^{(iter)}, N_2^{(iter)})$ the number of times that each change-point of $\{1, \dots, n\}^2$ was selected.

Output : Sequence of couple of numbers $(N_1^{(iter)}, N_2^{(iter)})$ recorded at each iteration.

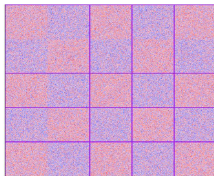
```
select=stab.blockSeg(Y,nsimu=1000,max.break = 10)
plot(select,Y,col="color")
```

```
select=stab.blockSeg(Y,nsimu=1000,max.break = 10)  
plot(select,Y,col="color")
```

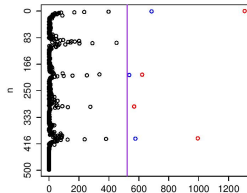
```
select=stab.blockSeg(Y,nsimu=1000,max.break = 10)
plot(select,Y,col="color")
```

40%

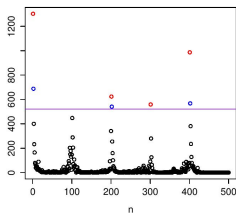
Original data



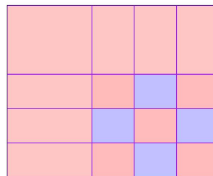
4 row breaks



4 column breaks



Summarized data

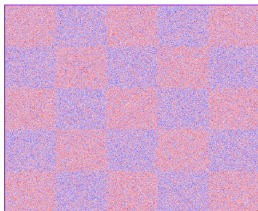


```
evolution(x      , y, thresholds = 10 * (8:1),  
postprocessing = list(post = TRUE, adjacent = 2),  
col = "GrayLevel", ask = TRUE)
```

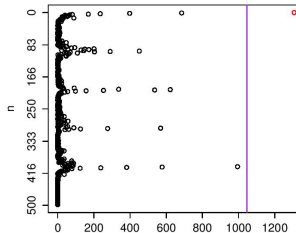
```
evolution(select, Y, thresholds = 10 * (8:1),  
postprocessing = list(post = TRUE, adjacent = 2),  
col = "Color"      , ask = TRUE)
```

80%

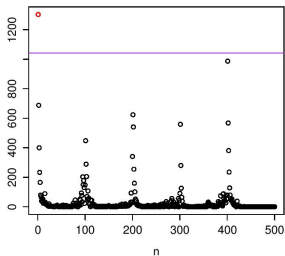
Original data



1 row break



1 column break

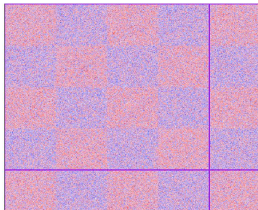


Summarized data

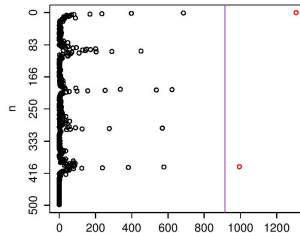


70%

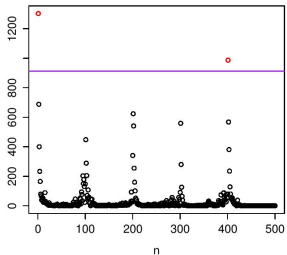
Original data



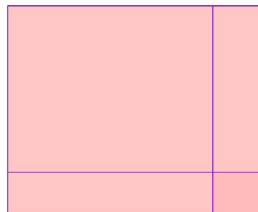
2 row breaks



2 column breaks

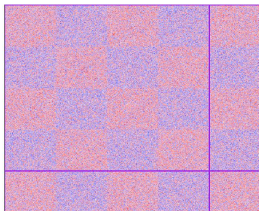


Summarized data

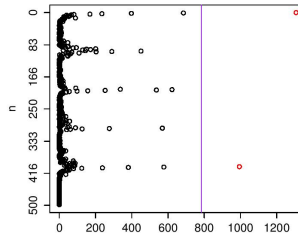


60%

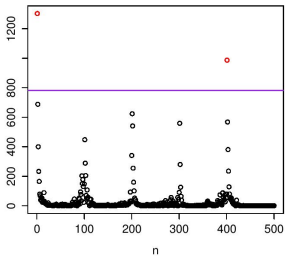
Original data



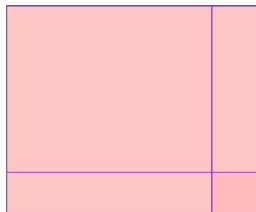
2 row breaks



2 column breaks

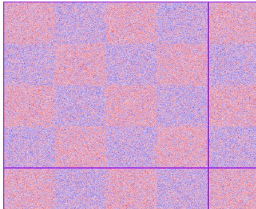


Summarized data

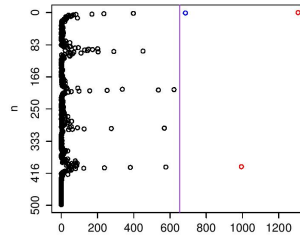


50%

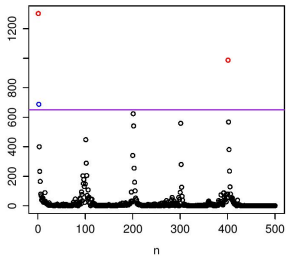
Original data



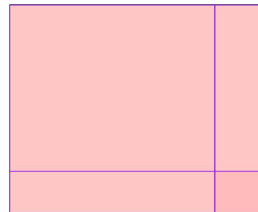
2 row breaks



2 column breaks

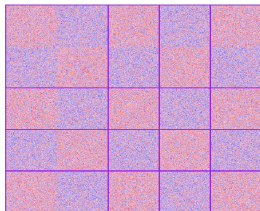


Summarized data

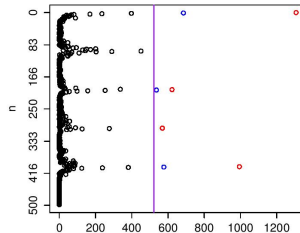


40%

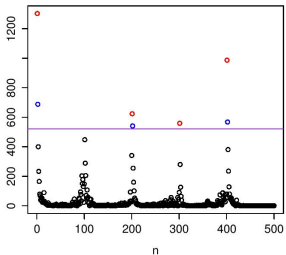
Original data



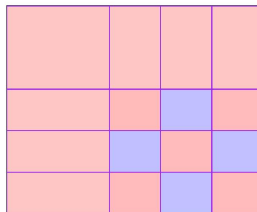
4 row breaks



4 column breaks

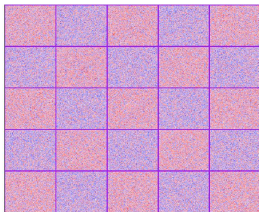


Summarized data

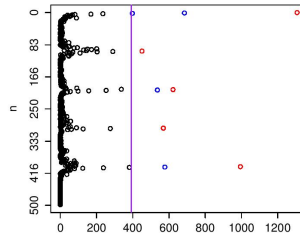


30%

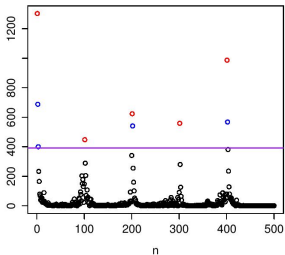
Original data



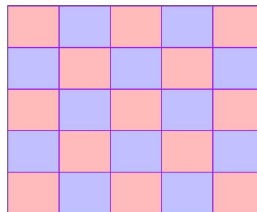
5 row breaks



5 column breaks

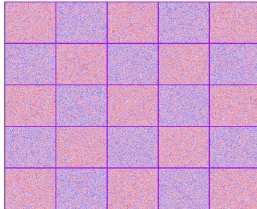


Summarized data

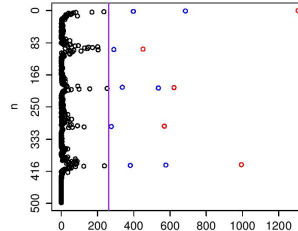


20%

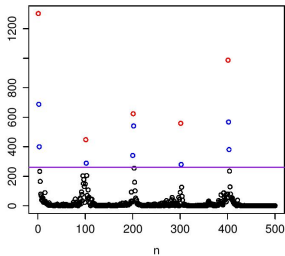
Original data



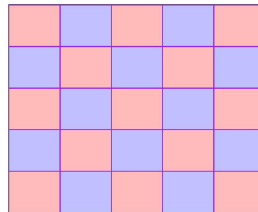
5 row breaks



5 column breaks

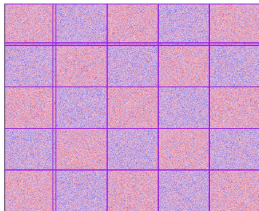


Summarized data

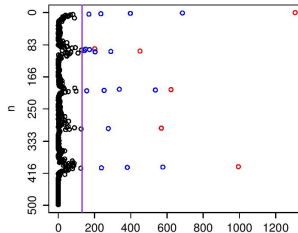


10%

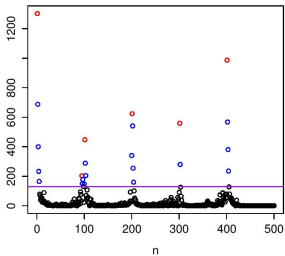
Original data



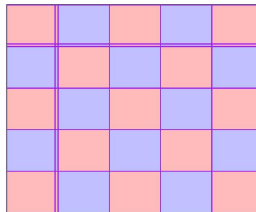
6 row breaks



6 column breaks



Summarized data



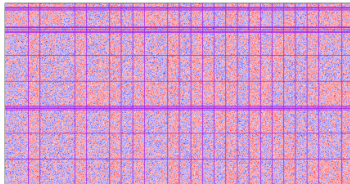
Perspectives

- Improvement the function plot.
- Inclusion of the case where $n_1 \neq n_2$.

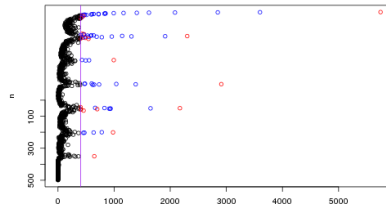
Thank you for your attention

7%

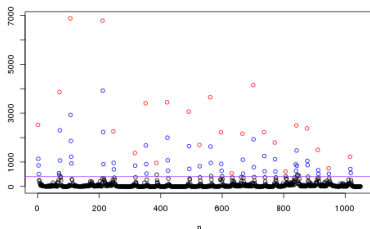
Original data



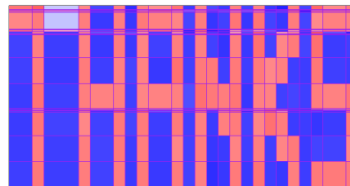
20 line breaks



23 column breaks



Summarized data



Bibliographie

- F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.
- V. Brault, J. Chiquet, and C. Lévy-Leduc. A fast approach for multiple change-point detection in two-dimensional data. Submitted. URL <http://arxiv.org/abs/1603.03593>.
- J. R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398) :376–380, 2012.
- Z. Harchaoui and C. Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492), 2010.

Plan

3 Usual notations

4 Gray film

Let $\mathbf{A} \in \mathcal{M}_{n \times m}(\mathbb{R})$ and $\mathbf{B} \in \mathcal{M}_{p \times q}(\mathbb{R})$ two matrices, the kronecker product of \mathbf{A} and \mathbf{B} is a matrix $(np) \times (mq)$ satisfying :

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & a_{n2}\mathbf{B} & \cdots & a_{nm}\mathbf{B} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & a_{12}b_{11} & \cdots & \cdots & a_{1m}b_{11} & \cdots & a_{1m}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & a_{12}b_{21} & \cdots & \cdots & a_{1m}b_{21} & \cdots & a_{1m}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & a_{12}b_{p1} & \cdots & \cdots & a_{1m}b_{p1} & \cdots & a_{1m}b_{pq} \\ \vdots & \vdots & & \vdots & \vdots & \ddots & & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \ddots & \vdots & & \vdots \\ a_{n1}b_{11} & a_{n1}b_{12} & \cdots & a_{n1}b_{1q} & a_{n2}b_{11} & \cdots & \cdots & a_{nm}b_{11} & \cdots & a_{nm}b_{1q} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & & \vdots & \ddots & \vdots \\ a_{n1}b_{p1} & a_{n1}b_{p2} & \cdots & a_{n1}b_{pq} & a_{n2}b_{p1} & \cdots & \cdots & a_{nm}b_{p1} & \cdots & a_{nm}b_{pq} \end{pmatrix}.$$

$$\begin{aligned}\mathcal{X} &= \mathbf{T} \otimes \mathbf{T} \\ &= \begin{pmatrix} \mathbf{T} & 0 & 0 & \dots & 0 \\ \mathbf{T} & \mathbf{T} & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ \mathbf{T} & \dots & \dots & & \mathbf{T} \end{pmatrix}\end{aligned}$$

[Return vectorisation](#)

$\|u\|_2^2$ is defined for a vector u in \mathbb{R}^N by

$$\|u\|_2^2 = \sum_{i=1}^N u_i^2$$

and $\|u\|_1$ is defined for a vector u in \mathbb{R}^N by

$$\|u\|_1 = \sum_{i=1}^N |u_i|.$$

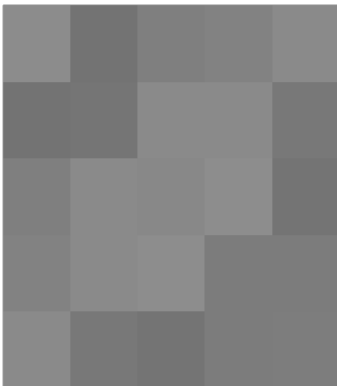
[Return LASSO](#)

Plan

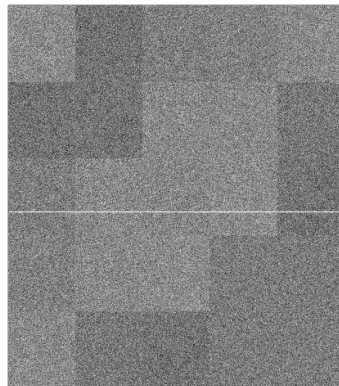
3 Usual notations

4 Gray film

Mu matrix



Original matrix



Colour version