Algorithm based on spectral analysis to detect numerical blocks in matrices

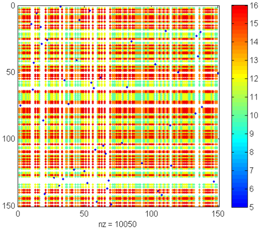
Luce le Gorrec[1]
with I. Duff, P.A. Knight, S. Mouysset, D. Ruiz

[1]IRIT

Sparse Days 2017
Toulouse, September 2017

What does it mean ?

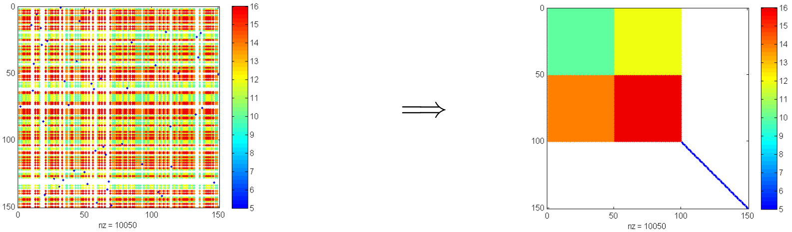
On a perfect matrix



No obvious block structure on this matrix.

What does it mean ?

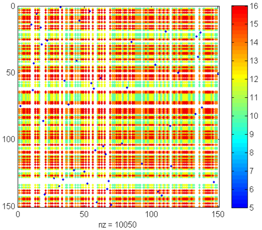
On a perfect matrix



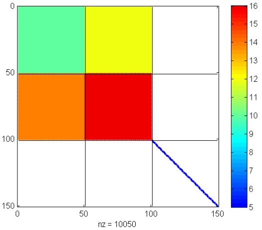
Finding permutations of rows and columns which highlight the block structure of the matrix.

What does it mean ?

On a perfect matrix



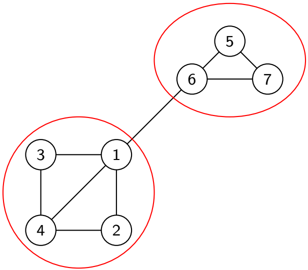
$\Longrightarrow$



Detecting the blocks. No *a priori* information about their number or their size.

What is for ?

- **Community Detection**
- Preconditioning of linear systems
- Clustering, Biclustering



$$\Longleftrightarrow \quad \left[\begin{array}{cccc|ccc} x & 1 & 1 & 1 & & 1 & \\ 1 & x & & 1 & & & \\ 1 & & x & 1 & & & \\ 1 & 1 & 1 & x & & & \\ \hline & & & & x & 1 & 1 \\ 1 & & & & 1 & x & 1 \\ & & & & 1 & 1 & x \end{array}\right]$$

What is for ?

- Community Detection
- **Preconditioning of linear systems**
- Clustering, Biclustering

What is for ?

- Community Detection
- Preconditioning of linear systems
- **Clustering, Biclustering**

Existing algorithms

Spectral

1. Find a singular vector.
2. Sort the rows/cols in the $\nearrow$ order of the vector.
3. Cut the vector into 2 blocks (its + and - entries).
4. Loop on previous process.

Drawbacks :

→ Only one cut per ite

→ Bipartition may not fit with the matrix structure

Papers :
[Fritzsche2007], [Newman2006]

Function optimisation

- Optimise a quality clustering function.

Drawbacks :

→ Depends on the chosen measure.

→ NP-Hard optimisation problem.

Papers :
[Aloise2010],[Campigotto2014]

Combination

- Like "Spectral" but the cut optimises a quality measure.

Drawbacks :

→ Only one cut per ite

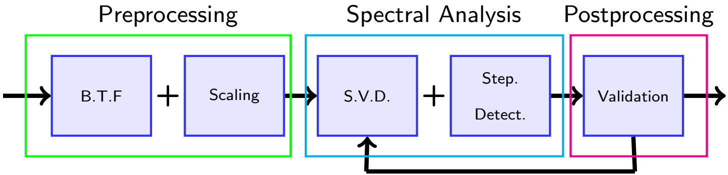
→ Depends on the chosen measure

Papers :
[Benson2016],[Vecharynski2014]

Our algorithm

Our algorithm :

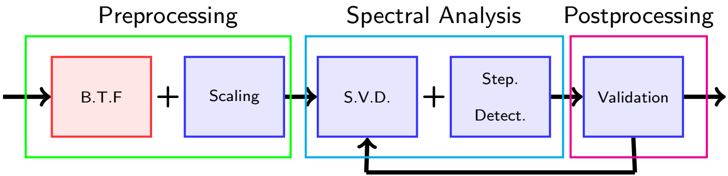
- Is a stage algorithm,
- Belongs to the first category, but
- Can find several blocks per iteration.



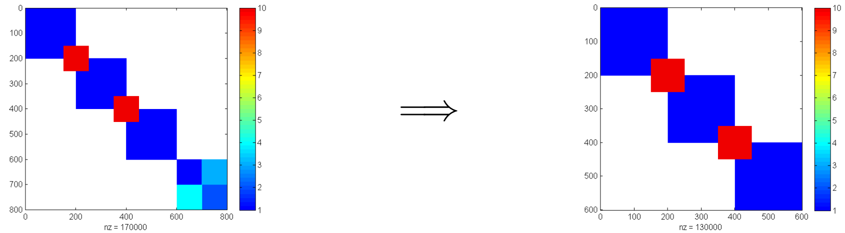
To find the blocks : analysis of the singular vector pattern, mainly based on tools from signal processing.

Our algorithm

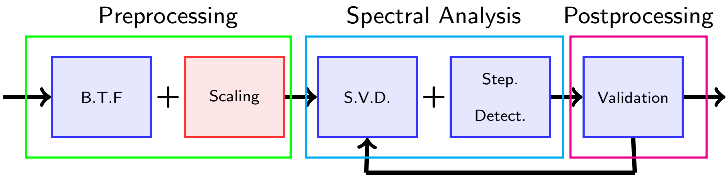
General pattern



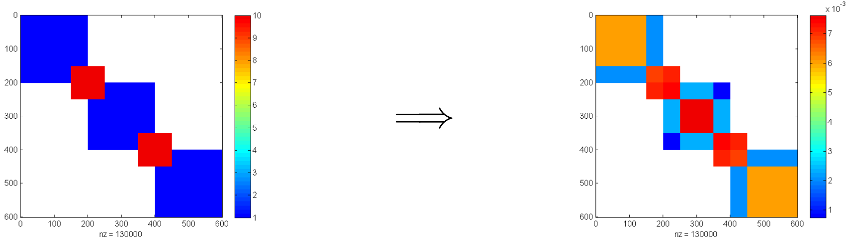
BTF permutation to find the dense blocks.



Our algorithm
General pattern

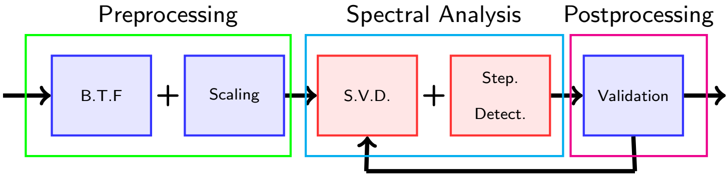


Doubly stochastic scaling to highlight the numerical blocks.

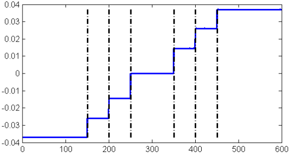


Our algorithm

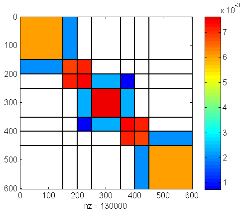
General pattern



Dominant singular vectors (potentially only 1) allow to detect these blocks.

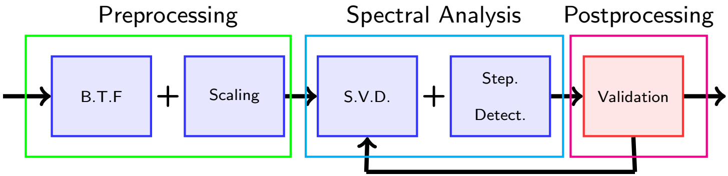


$\Longrightarrow$



Our algorithm

General pattern



If several iterations :

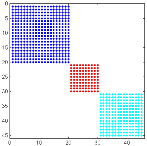
- Clustering overlapping.
- Removal of the no needed clusters.
- Convergence test.

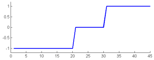
Doubly Stochastic Scaling

- **Concept :**
  For a fully indecomposable matrix A, finding two diagonal matrices R and S such that

$$RASe = e, SA^T Re = e, \text{ with } e = (1...1)^T$$

- **Interest :**

  Perfect case : $A =$ 

  , $v =$ 

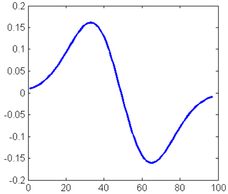
  and $Av = v$.

  Non perfect case : A has a structure near to a block structure $\Rightarrow$ v has a structure near to staircase.

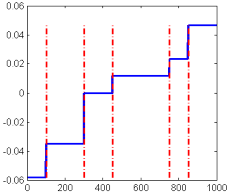
- **Remark :**
  $Ae = e$, so the dominant singular vectors are taken in $e^{\perp}$.

- Filters can detect steps in a signal.
- The singular vector : a signal we want to detect the steps.
- Convolution product between a signal and a filter.

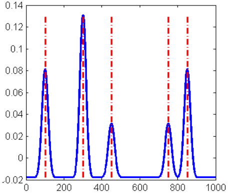
$\Rightarrow$ The maxima correspond to the steps in the signal.



filter



vector



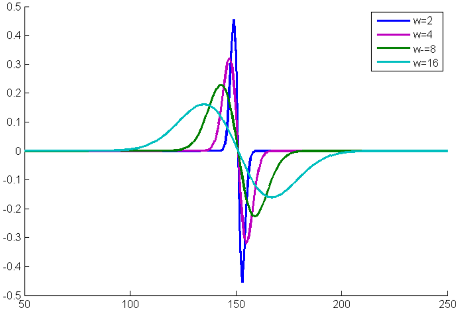
convolution

Block detection

Signal processing

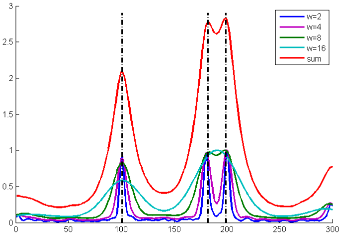
- **Important filter parameter : the width**
- Small width : detects small blocks, noise sensitive.
  Big width : noise resistant, detects only the largest blocks.

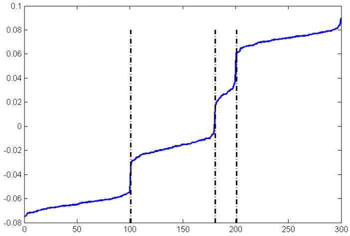
⇒ Convolution product for different sizes of width.
Sum of the results to detect small blocks and be noise resistant.



- Important filter parameter : the width
- Small width : detects small blocks, noise sensitive.
  Big width : noise resistant, detects only the largest blocks.

$\Rightarrow$ Convolution product for different sizes of width.
Sum of the results to detect small blocks and be noise resistant.

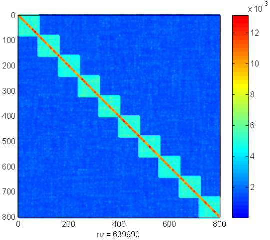


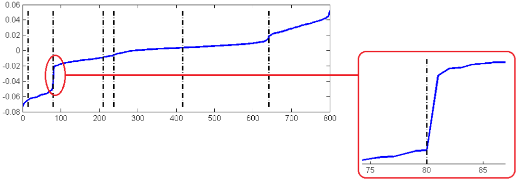


Block detection

The filter provides three kinds of separations :

1. **Those corresponding to the sharp edges.**
2. Those corresponding to the smooth edges.
3. The spurious.





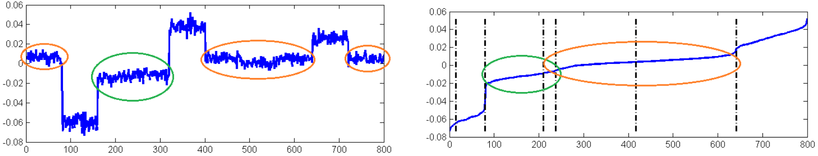
Block detection

The filter provides three kinds of separations :

1 Those corresponding to the sharp edges.

2 **Those corresponding to the smooth edges.**

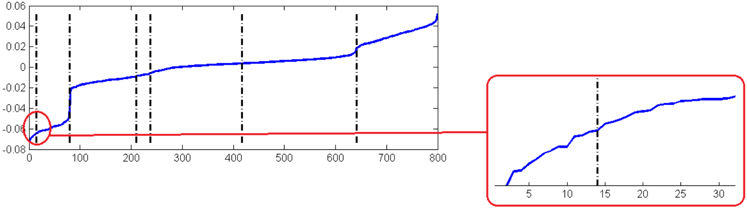
3 The spurious.



Block detection

The filter provides three kinds of separations :

1. Those corresponding to the sharp edges.
2. Those corresponding to the smooth edges.
3. **The spurious.**



Block detection

The filter provides three kinds of separations :

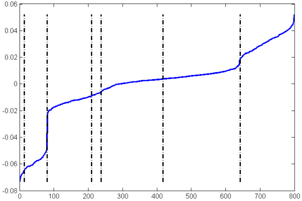
1 Those corresponding to the sharp edges.

2 Those corresponding to the smooth edges.

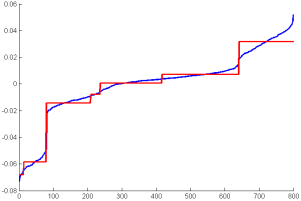
3 The spurious.

$\Rightarrow$ **Need of an edge refinement process**

- Creation of the clustering characteristic vector.



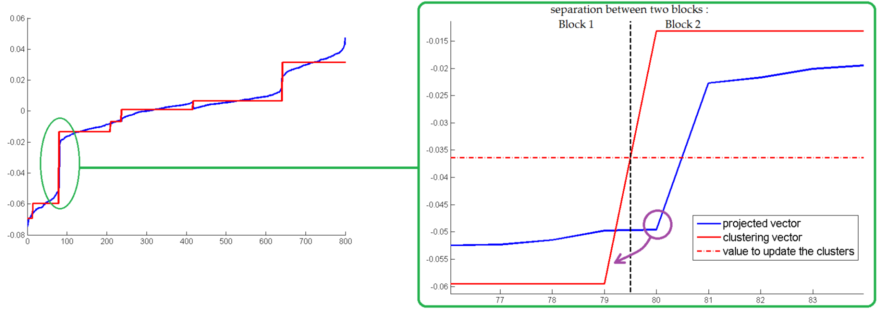
$\Longrightarrow$



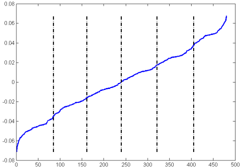
Block detection

Edge refinement : Headlines of the process

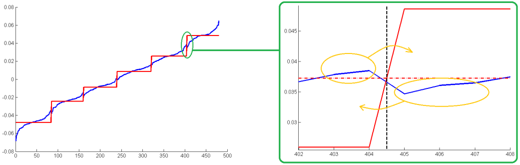
- Creation of the clustering characteristic vector.
- Projection of this vector in the space of the singular vectors :
  - highlights separations to shift (sharp case).



- Creation of the clustering characteristic vector.
- Projection of this vector in the space of the singular vectors :
  - highlights separations to shift (sharp case).
  - highlights indices to exchange (smooth case).



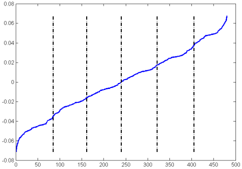
$\Rightarrow$



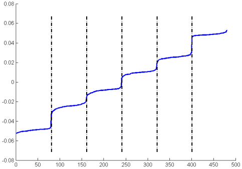
Block detection

Edge refinement : Headlines of the process

- Creation of the clustering characteristic vector.
- Projection of this vector in the space of the singular vectors :
  - highlights separations to shift (sharp case).
  - highlights indices to exchange (smooth case).
  - highlights spurious separation (spurious case).
- Update of the clustering.
- Loop on the previous process.
- After convergence, remaining separations correspond to sharp and some smoother steps.



$\Rightarrow$

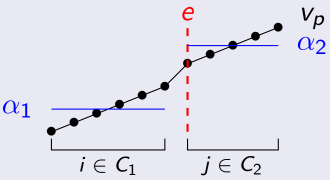


Block detection

Need of a sharpness measure to :

- Characterise the spurious edges,
- Keep only the sharpest edges.

For an edge e...



with v_p the projected vector, C_1 the set of indices in the block before e, C_2 the set of indices in the block after e.

...this measure is

$$r(e) = \frac{\alpha_2 - \alpha_1}{\epsilon_1 + \epsilon_2}$$

with :

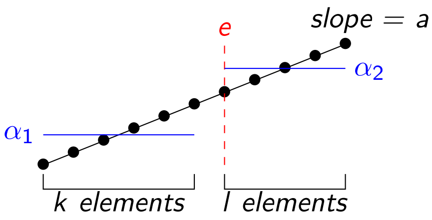
$\alpha_1 = \frac{1}{|C_1|} \sum_{i \in C_1} v_p(i)$,

$\alpha_2 = \frac{1}{|C_2|} \sum_{j \in C_2} v_p(j)$,

$\epsilon_1 = \frac{1}{|C_1|} \sum_{i \in C_1} |\alpha_1 - v_p(i)|$ noise in block C_1

$\epsilon_2 = \frac{1}{|C_2|} \sum_{j \in C_2} |\alpha_2 - v_p(j)|$ noise in block C_2

An obvious spurious edge :



its value depending on k and l :

$r(e)$	k odd	k even
l odd	$2(1+\frac{1}{kl-1})$	$2(1+\frac{1}{l^2+kl-1})$
l even	$2(1+\frac{1}{k^2+kl-1})$	2

- Spurious edges : an edge is spurious if its measure is under this threshold.
- Sharpest edges : keeping the edges with a value over 4.5 provides good results.

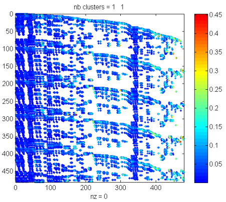
Block detection

Number of clusters

- Block detection process applied on each singular vector in the set.
- Possibility of finding a new set of vectors.

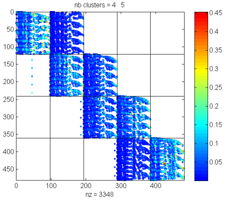
$\Rightarrow$ The number of clusters can quickly become huge.

RBSB matrix (University of Florida Sparse Matrix Collection)



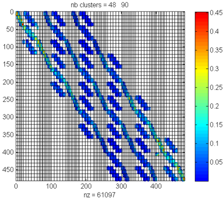
initial matrix

$\Rightarrow$



after the first step

$\Rightarrow$



after the third step

$\Rightarrow$ Process of cluster amalgamation after each iteration.

Modularity

Modularity measure :

$$\mathcal{Q}_r = \frac{1}{m} \sum_{k=1}^{r_c} \left(v_k^T A A^T v_k - \frac{1}{m} |C_k| \right) \geq 0$$

- Test for pairwise amalgamation that maximise the increase of the quality measure.
- Loop until local maximum is reached.
- No pairwise amalgamation improvement implies no improvement by any type of amalgamation on a current state.

Modularity

Non-linear upper bound of the modularity :

$$\mathcal{Q}_r \leq 1 - \frac{1}{nb_{Clust}} = \mathcal{K}_r$$

Calling ρ the ratio

$$\rho = \frac{\mathcal{Q}_r^{upd} - \mathcal{Q}_r^{ref}}{\mathcal{K}_r^{upd} - \mathcal{K}_r^{ref}} = \frac{\mathcal{Q}_r^{upd} - \mathcal{Q}_r^{ref}}{\frac{1}{nb_{clust}{}^{ref}} - \frac{1}{nb_{clust}{}^{upd}}},$$

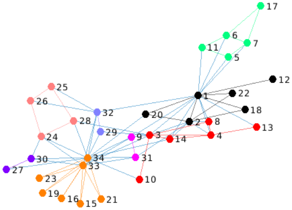
A new clustering is accepted iff $\rho > \epsilon$, with $\epsilon \in [0.01, 0.1]$ a threshold.

$\Rightarrow$ **A new clustering is accepted only if the gain of modularity worths it.**

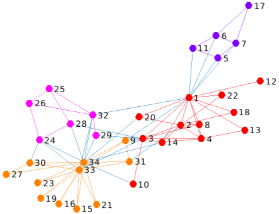
Community detection

Methodology

- The algorithm works on $A + 10^{-8} I_n$.
- Use of the Newman's Modularity to compare algorithms.
- Too many clusters : a last amalgamation on the adjacency matrix.



$\Longrightarrow$



Modularity $= 0.3303$, 8 clusters

Modularity $= 0.4188$, 4 clusters

Zachary's Karate club Network

Community detection

Results

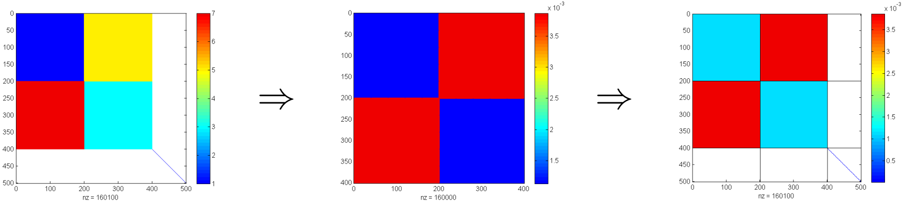
Network	Size	Modularity			
		Best	Ours	Louvain	Edge Betweeness
Karate	34	0.4198	**0.4188** (4ite) (2sv)	**0.4188** (8ite)	0.4013
Dolphins	62	0.5285	0.4723 (6ite) (4sv)	0.5188 (7ite)	0.5194
les Miserables	77	0.5600	0.5545 (6ite) (2sv)	0.5556 (8ite)	0.5381
A00	83	0.5309	**0.5268** (4ite) (3sv)	0.5259 (9ite)	0.5050
Politics Books	105	0.5272	0.4757 (7ite) (3sv)	0.4986 (6ite)	0.5168
Football Clubs	115	0.6046	0.5985 (7ite) (3sv)	0.6046 (6ite)	0.5996
USAir	332	0.3682	0.3201 (10ite) (3sv)	0.3518 (15ite)	0.1364
netsciences	379	0.8486	0.8227 (8ite) (3sv)	0.8440 (13ite)	0.8422
s838	512	0.8194	0.7954 (9ite) (3sv)	0.7185 (17ite)	0.8155

Louvain Algorithm : *[Blondel2008]*
Edge Betweeness : *[Newman2004]*

Algorithm : currently in testing phase

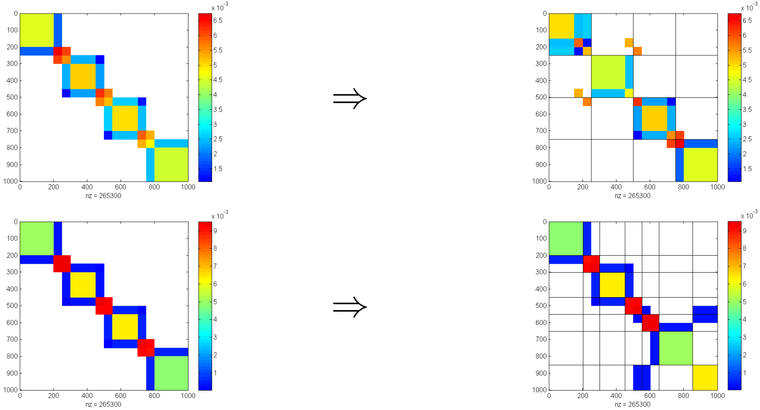
⇒ **Exact on perfect matrices.**

⇒ Good results on density matrices.



⇒ **Exact on perfect matrices.**

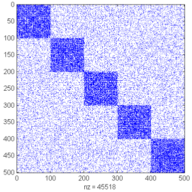
⇒ Good results on density matrices.



Algorithm : currently in testing phase

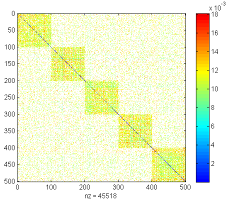
⇒ Exact on perfect matrices.

⇒ **Good results on density matrices.**



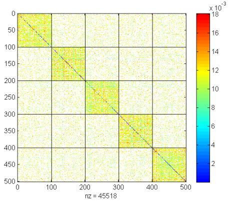
$p_{in} = 0.5; p_{out} = 0.1$

⇒

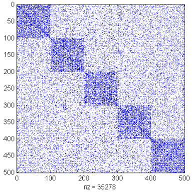


after scaling

⇒

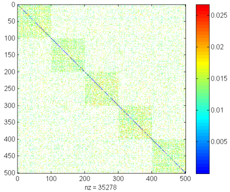


perfect clustering



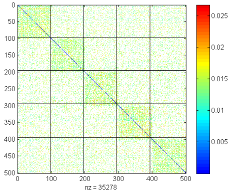
$p_{in} = 0.3; p_{out} = 0.1$

⇒



after scaling

⇒



a few mistakes

Outlooks

- Generalisation to the case of rectangular matrices.
- Scalability.
- Application in biology.
- Application as a preconditioner.
- Behaviour of other modularity measures.
  Remark : doubly stochastic scaling $\Rightarrow$ homogenisation of several modularity measures.

Thank you for your attention.

Any questions ?