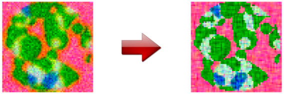
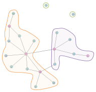
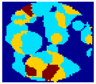
Reconstruction and Clustering with Graph optimization and Priors on Gene Networks and Images

Aurélie Pirayre

PhD supervisors:	Frédérique BIDARD-MICHELOT	*IFP Energies nouvelles*
	Camille COUPRIE	*Facebook A.I. Research*
	Laurent DUVAL	*IFP Energies nouvelles*
	Jean-Christophe PESQUET	*CentraleSupélec*

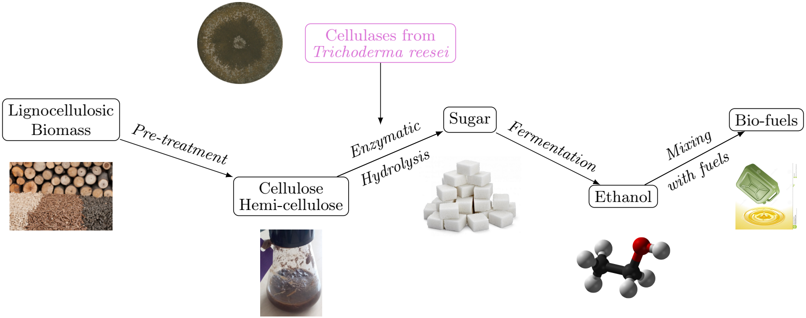
July 3th, 2017

An overview

	Gene regulatory networks	Signals and images
Reconstruction		
Clustering		
Our framework	Variational	Bayes variational
Method	*BRANE*	*HOGMep*

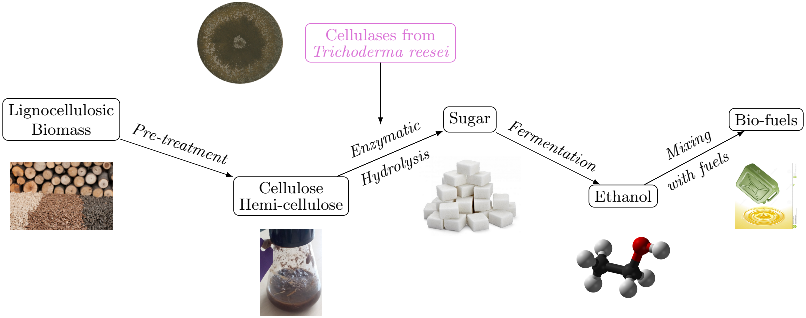
Biological motivation

- Second generation bio-fuel production



Biological motivation

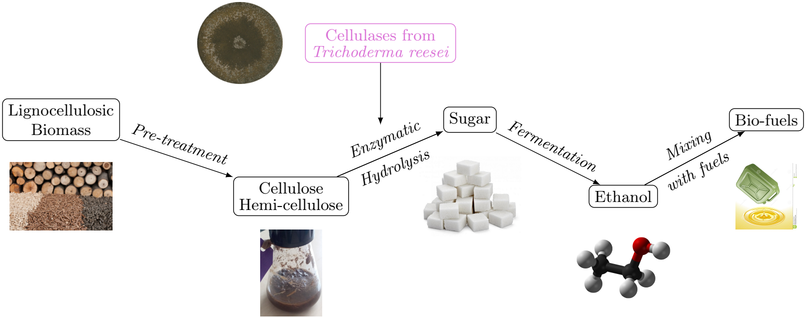
- Second generation bio-fuel production



- Improve *Trichoderma reseei* cellulase production
- Understand cellulase production mechanisms

Biological motivation

- Second generation bio-fuel production

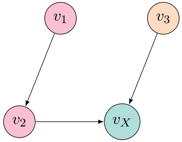


- Improve *Trichoderma reseei* cellulase production
- Understand cellulase production mechanisms

$\Rightarrow$ Use of Gene Regulatory Network (GRN)

What is a Gene Regulatory Network (GRN)?

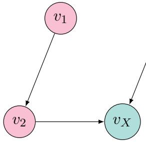
GRN: a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$
$\mathcal{V} = \{v_1, \ldots, v_G\}$: a set of G nodes (corresponding to genes)
$\mathcal{E}$: a set of edges (corresponding to interactions between genes)



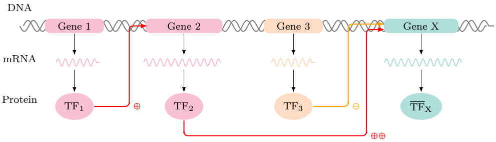
What is a Gene Regulatory Network (GRN)?

GRN: a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$
$\mathcal{V} = \{v_1, \ldots, v_G\}$: a set of G nodes (corresponding to genes)
$\mathcal{E}$: a set of edges (corresponding to interactions between genes)

A gene regulatory network...



... models biological gene regulatory mechanisms



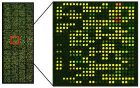
What biological data can be used?

For a given experimental condition, transcriptomic data answer to:
which genes are expressed? in which amount?

What biological data can be used?

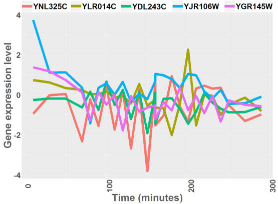
For a given experimental condition, transcriptomic data answer to:
which genes are expressed? in which amount?

How to obtain transcriptomic data?
Microarray and RNAseq experiments



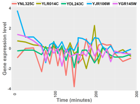
What do transcriptomic data look like?
Gene expression data (GED): G genes $\times$ S conditions

$$\mathbf{M} = \overbrace{\begin{pmatrix} -0.948 & -0.013 & \ldots & -1.308 & -0.977 \\ 0.737 & 0.619 & \ldots & -0.141 & -0.803 \\ -0.253 & -0.175 & \ldots & -0.859 & -0.595 \\ 3.747 & 1.115 & \ldots & -0.418 & -0.084 \\ 1.383 & 1.184 & \ldots & -0.493 & -0.562 \end{pmatrix}}^{S \text{ conditions}} \left.\vphantom{\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}}\right\} G \text{ genes}$$



How to use GED to produce a GRN ?

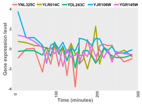
From gene expression data...



$$\mathbf{M} = \begin{array}{cc} & s_j \\ g_i & \begin{pmatrix} & \vdots \\ \cdots & m_{i,j} \end{pmatrix} \end{array},$$

How to use GED to produce a GRN ?

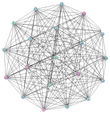
From gene expression data...



$$\mathbf{M} = \begin{array}{cc} & s_j \\ g_i & \begin{pmatrix} & \vdots \\ \cdots & m_{i,j} \end{pmatrix} \end{array},$$

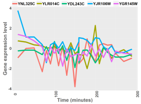
- $\mathcal{V} = \{v_1, \cdots, v_G\}$ a set of **vertices** (**genes**) and $\mathcal{E}$ a set of **edges**

leading to a complete graph...



How to use GED to produce a GRN ?

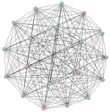
From gene expression data...



$$\mathbf{M} = \begin{array}{cc} & s_j \\ g_i & \begin{pmatrix} & \vdots & \\ \cdots & m_{i,j} & \\ & & \end{pmatrix} \end{array},$$

- $\mathcal{V} = \{v_1, \cdots, v_G\}$ a set of **vertices** (**genes**) and $\mathcal{E}$ a set of **edges**
- Each edge $e_{i,j}$ is **weighted** by $\omega_{i,j}$

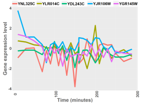
leading to a complete weighted graph...



$$\mathbf{W} = \begin{array}{cc} & g_j \\ g_i & \begin{pmatrix} & \vdots & \\ \cdots & \omega_{i,j} & \\ & & \end{pmatrix} \end{array},$$

How to use GED to produce a GRN ?

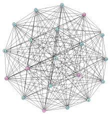
From gene expression data...



$$\mathbf{M} = \begin{array}{cc} & s_j \\ g_i & \begin{pmatrix} & \vdots \\ \cdots & m_{i,j} \end{pmatrix} \end{array},$$

- $\mathcal{V} = \{v_1, \cdots, v_G\}$ a set of **vertices** (**genes**) and $\mathcal{E}$ a set of **edges**
- Each edge $e_{i,j}$ is **weighted** by $\omega_{i,j}$

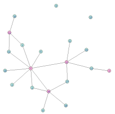
leading to a complete weighted graph...



$$\mathbf{W} = \begin{array}{cc} & g_j \\ g_i & \begin{pmatrix} & \vdots \\ \cdots & \omega_{i,j} \end{pmatrix} \end{array},$$

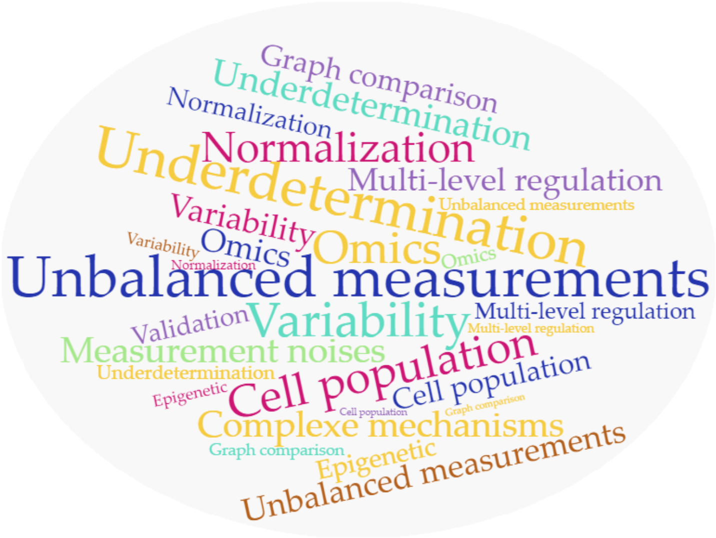
- We look for a subset of edges $\mathcal{E}^*$ reflecting **regulatory links between genes**

to infer a meaningful gene network.

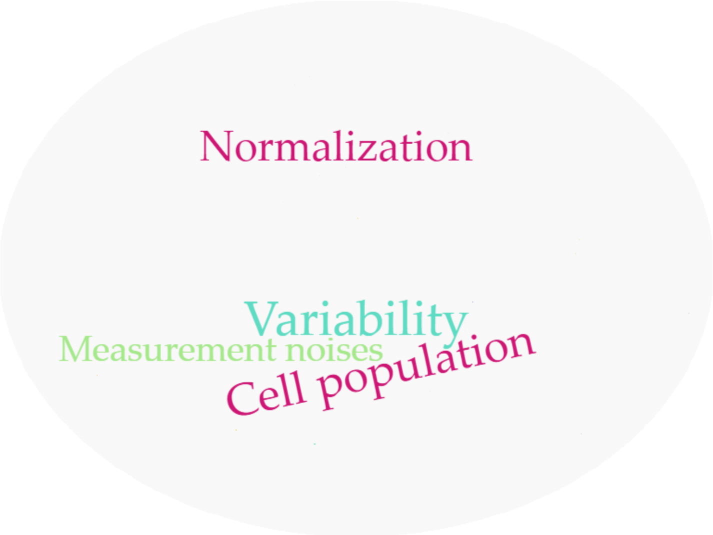


$$\mathbf{W}_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \in \mathcal{E}^* \\ 0 & \text{otherwise.} \end{cases}$$

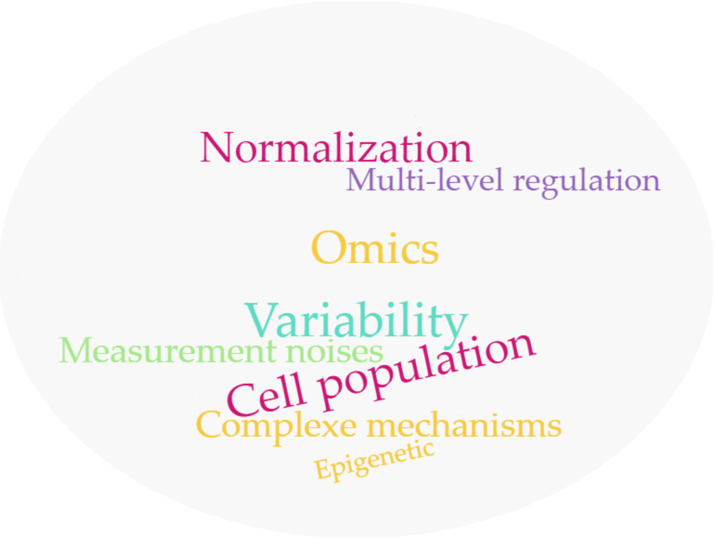
Difficulties in GRN inference



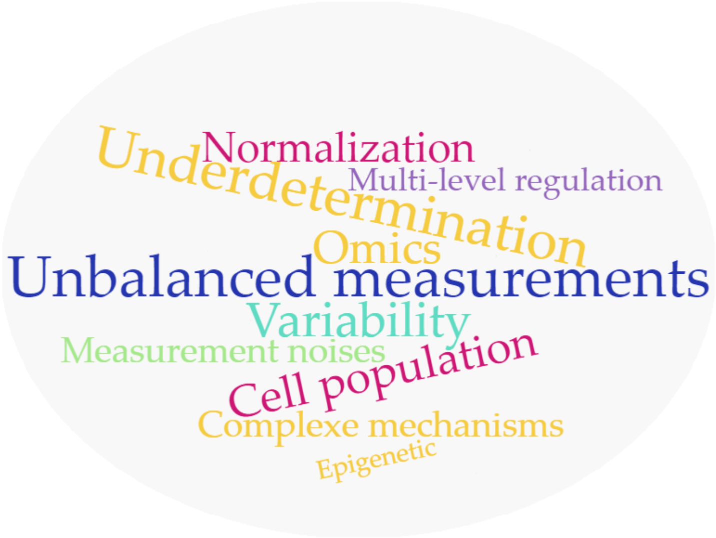
Difficulties in GRN inference



Difficulties in GRN inference



Difficulties in GRN inference



Our *BRANE* strategy

What is the subset of edges $\mathcal{E}^*$ reflecting real regulatory links between genes?
$\Rightarrow$ what is the binary adjacency matrix $\mathbf{W} \in \{0, 1\}^{G \times G}$?

Our *BRANE* strategy

What is the subset of edges $\mathcal{E}^*$ reflecting real regulatory links between genes?
$\Rightarrow$ what is the binary adjacency matrix $\mathbf{W} \in \{0,1\}^{G\times G}$?

- We note $x_{i,j}$ the binary label of edge presence: $x_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \in \mathcal{E}^*, \\ 0 & \text{otherwise.} \end{cases}$

Our *BRANE* strategy

What is the subset of edges $\mathcal{E}^*$ reflecting real regulatory links between genes?
$\Rightarrow$ what is the binary adjacency matrix $\mathbf{W} \in \{0,1\}^{G\times G}$?

- We note $x_{i,j}$ the binary label of edge presence: $x_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \in \mathcal{E}^*, \\ 0 & \text{otherwise.} \end{cases}$
- Classical thresholding: $x^*_{i,j} = \begin{cases} 1 & \text{if } \omega_{i,j} > \lambda, \\ 0 & \text{otherwise.} \end{cases}$

Our *BRANE* strategy

What is the subset of edges $\mathcal{E}^*$ reflecting real regulatory links between genes?
$\Rightarrow$ what is the binary adjacency matrix $\mathbf{W} \in \{0,1\}^{G \times G}$?

- We note $x_{i,j}$ the binary label of edge presence: $x_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \in \mathcal{E}^*, \\ 0 & \text{otherwise.} \end{cases}$
- Classical thresholding: $x^*_{i,j} = \begin{cases} 1 & \text{if } \omega_{i,j} > \lambda, \\ 0 & \text{otherwise.} \end{cases}$
- Given by a cost function for given weights ω:

$$\underset{\boldsymbol{x} \in \{0,1\}^E}{\text{maximize}} \sum_{(i,j) \in \mathbb{V}^2} \omega_{i,j}\, x_{i,j} + \lambda(1 - x_{i,j})$$

Our *BRANE* strategy

What is the subset of edges $\mathcal{E}^*$ reflecting real regulatory links between genes?
$\Rightarrow$ what is the binary adjacency matrix $\mathbf{W} \in \{0,1\}^{G \times G}$?

- We note $x_{i,j}$ the binary label of edge presence: $x_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \in \mathcal{E}^*, \\ 0 & \text{otherwise.} \end{cases}$
- Classical thresholding: $x^*_{i,j} = \begin{cases} 1 & \text{if } \omega_{i,j} > \lambda, \\ 0 & \text{otherwise.} \end{cases}$
- Given by a cost function for given weights ω:

$$\underset{\boldsymbol{x} \in \{0,1\}^E}{\text{maximize}} \sum_{(i,j) \in \mathbb{V}^2} \omega_{i,j}\, x_{i,j} + \lambda(1 - x_{i,j}) \Leftrightarrow \underset{\boldsymbol{x} \in \{0,1\}^E}{\text{minimize}} \sum_{(i,j) \in \mathbb{V}^2} \omega_{i,j}\,(1 - x_{i,j}) + \lambda\, x_{i,j}$$

Our *BRANE* strategy

BRANE: Biologically Related A priori Network Enhancement

- Extend classical thresholding
- Integrate biological priors into the functional to be optimized
- Enforce modular networks
- Additional knowledge:
  - Transcription factors (TFs): regulators
  - Non transcription factors ($\overline{\text{TFs}}$): targets

Our *BRANE* strategy

BRANE: Biologically Related A priori Network Enhancement

- Extend classical thresholding
- Integrate biological priors into the functional to be optimized
- Enforce modular networks
- Additional knowledge:
  - Transcription factors (TFs): regulators
  - Non transcription factors ($\overline{\text{TFs}}$): targets

	Method	*a priori*	Formulation	Algorithm
Inference	*BRANE Cut* *BRANE Relax*	Gene co-regulatiton $\overline{\text{TF}}$-connectivity	Discrete Continuous	Maximal flow Proximal method
Joint inference and clustering	*BRANE Clust*	Gene grouping	Mixed	Alternating scheme

A discrete method: *BRANE Cut*

We look for a discrete solution for $\boldsymbol{x} \Leftrightarrow \boldsymbol{x} \in \{0, 1\}^E$

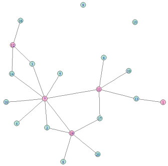
A priori: modular structure and gene co-regulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \;+\; \lambda_{i,j}\varphi(x_{i,j}) \;+\; \mu\Psi(x_{i,j})$$

A priori: modular structure and gene co-regulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \;+\; \lambda_{i,j}\varphi(x_{i,j}) \;+\; \mu\Psi(x_{i,j})$$

- Modular network: favors links between TFs and $\overline{\text{TFs}}$

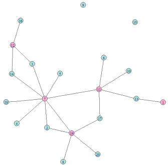


$$\lambda_{i,j} = \begin{cases} 2\eta & \text{if } (i,j) \notin \mathbb{T}^2 \\ 2\lambda_{\text{TF}} & \text{if } (i,j) \in \mathbb{T}^2 \\ \lambda_{\text{TF}} + \lambda_{\overline{\text{TF}}} & \text{otherwise.} \end{cases}$$

A priori: modular structure and gene co-regulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \; + \; \lambda_{i,j}\varphi(x_{i,j}) \; + \; \mu\Psi(x_{i,j})$$

- Modular network: favors links between TFs and $\overline{\text{TFs}}$



$$\lambda_{i,j} = \begin{cases} 2\eta & \text{if } (i,j) \notin \mathbb{T}^2 \\ 2\lambda_{\text{TF}} & \text{if } (i,j) \in \mathbb{T}^2 \\ \lambda_{\text{TF}} + \lambda_{\overline{\text{TF}}} & \text{otherwise.} \end{cases}$$

with:

- $\mathbb{T}$: the set of TF indices
- $\eta > \max\{\omega_{i,j} \,|\, (i,j) \in \mathbb{V}^2\}$
- $\lambda_{\text{TF}} > \lambda_{\overline{\text{TF}}}$

A linear relation is sufficient: $\lambda_{\text{TF}} = \beta\lambda_{\overline{\text{TF}}}$ with $\beta = \frac{|\mathcal{V}|}{|\mathcal{T}|}$

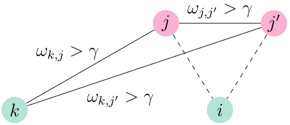
A priori: modular structure and gene co-regulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \;+\; \lambda_{i,j}\varphi(x_{i,j}) \;+\; \mu\Psi(x_{i,j})$$

A priori: modular structure and gene co-regulation

$$\boxed{\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \;+\; \lambda_{i,j}\varphi(x_{i,j}) \;+\; \mu\Psi(x_{i,j})}$$

- Gene co-regulation: favors edge coupling



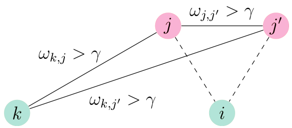
$$\Psi(x_{i,j}) = \sum_{\substack{(j,j')\in\mathbb{T}^2\\ i\in\mathbb{V}\setminus\mathbb{T}}} \rho_{i,j,j'}\left|x_{i,j}-x_{i,j'}\right|$$

$\rho_{i,j,j'}$: co-regulation probability

A priori: modular structure and gene co-regulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \; + \; \lambda_{i,j}\varphi(x_{i,j}) \; + \; \mu\Psi(x_{i,j})$$

- Gene co-regulation: favors edge coupling



$$\Psi(x_{i,j}) = \sum_{\substack{(j,j')\in\mathbb{T}^2\\ i\in\mathbb{V}\setminus\mathbb{T}}} \rho_{i,j,j'}|x_{i,j}-x_{i,j'}|$$

$\rho_{i,j,j'}$: co-regulation probability

with

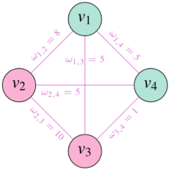
- $\rho_{i,j,j'} = \dfrac{\sum_{k\in\mathcal{V}\setminus(\mathcal{T}\cup\{i\})} \mathbb{1}(\min\{\omega_{j,j'},\omega_{j,k},\omega_{j',k}\} > \gamma)}{|\mathcal{V}\setminus\mathcal{T}|-1}$
- γ: the $(|\mathcal{V}|-1)^{\text{th}}$ of the normalized weights ω

A maximal flow for a minimum cut formulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}|x_{i,j}-1| \;+\; \lambda_{i,j}\,x_{i,j} \;+\; \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T} \\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\,|x_{i,j}-x_{i,j'}|$$

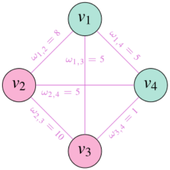
A maximal flow for a minimum cut formulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}|x_{i,j}-1| \;+\; \lambda_{i,j}\, x_{i,j} \;+\; \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T}\\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\,|x_{i,j}-x_{i,j'}|$$



A maximal flow for a minimum cut formulation

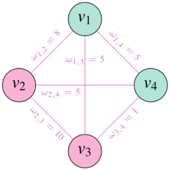
$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}|x_{i,j}-1| \;+\; \lambda_{i,j}\,x_{i,j} \;+ \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T}\\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\left|x_{i,j}-x_{i,j'}\right|$$



s t

A maximal flow for a minimum cut formulation

$$\underset{x \in \{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}|x_{i,j}-1| \;+\; \lambda_{i,j}\, x_{i,j} \;+\; \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T} \\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\,|x_{i,j}-x_{i,j'}|$$



$x_{1,2}$

$x_{1,3}$

$x_{1,4}$

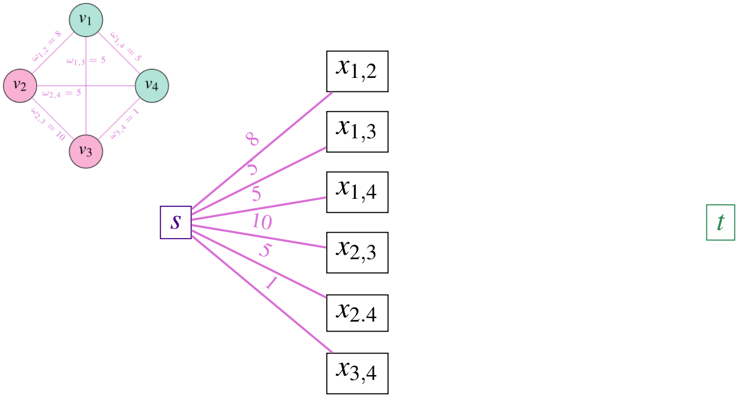
s $x_{2,3}$ t

$x_{2.4}$

$x_{3,4}$

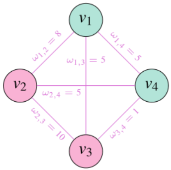
A maximal flow for a minimum cut formulation

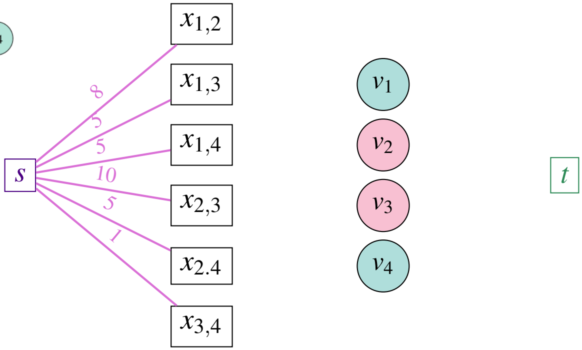
$$\underset{x \in \{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}|x_{i,j} - 1| \;+\; \lambda_{i,j}\, x_{i,j} \;+\; \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T} \\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\,|x_{i,j} - x_{i,j'}|$$



A maximal flow for a minimum cut formulation

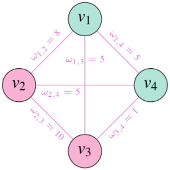
$$\underset{x\in\{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}|x_{i,j}-1| + \lambda_{i,j}\, x_{i,j} + \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T}\\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\left|x_{i,j}-x_{i,j'}\right|$$

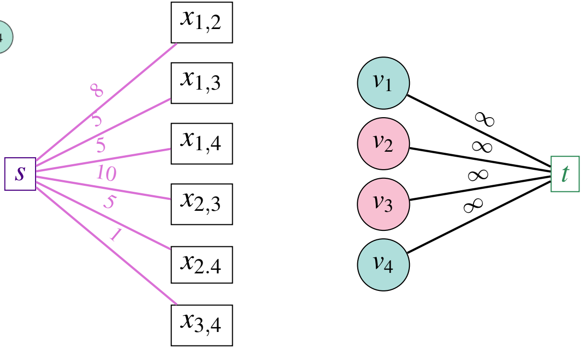




A maximal flow for a minimum cut formulation

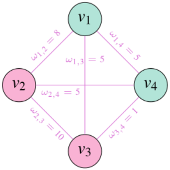
$$\underset{x \in \{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}|x_{i,j}-1| \;+\; \lambda_{i,j}\, x_{i,j} \;+\; \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T}\\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\,|x_{i,j}-x_{i,j'}|$$

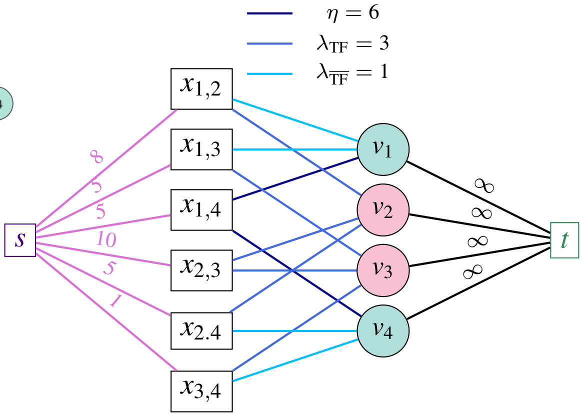




A maximal flow for a minimum cut formulation

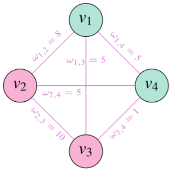
$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}|x_{i,j}-1| \; + \; \lambda_{i,j}\, x_{i,j} \; + \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T}\\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\left|x_{i,j}-x_{i,j'}\right|$$

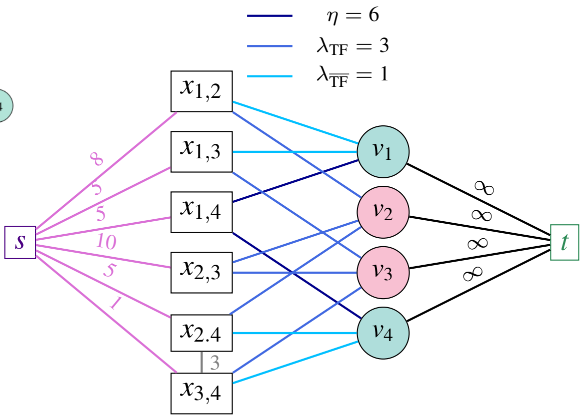




A maximal flow for a minimum cut formulation

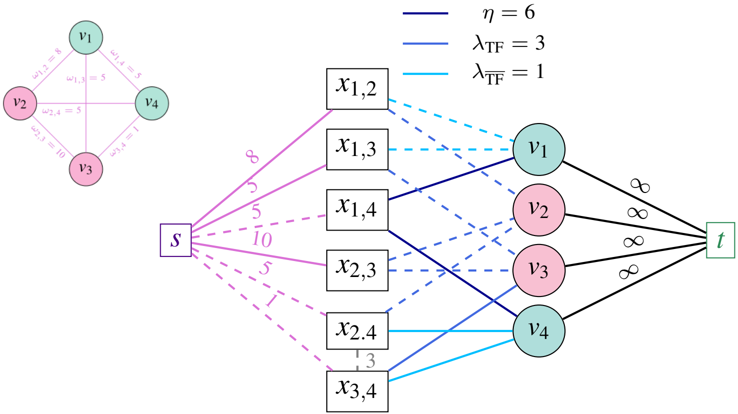
$$\underset{x \in \{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}|x_{i,j}-1| + \lambda_{i,j}\, x_{i,j} + \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T} \\ (j,j')\in\mathbb{T}^2, j'>j}} \rho_{i,j,j'}|x_{i,j}-x_{i,j'}|$$





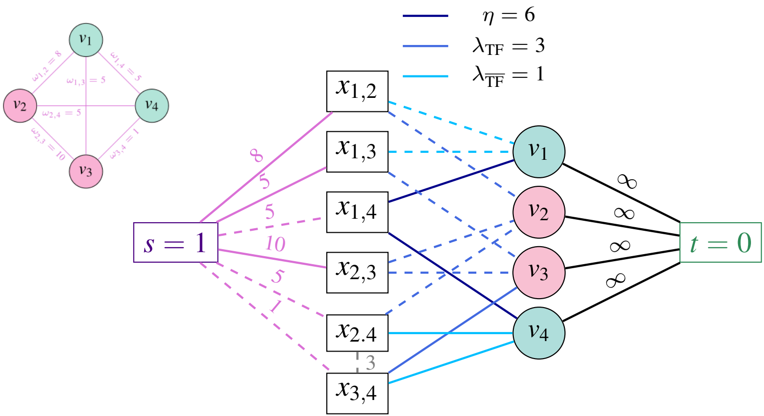
A maximal flow for a minimum cut formulation

$$\underset{x \in \{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j) \in \mathbb{V}^2 \\ j>i}} \omega_{i,j}|x_{i,j} - 1| + \lambda_{i,j}\, x_{i,j} + \sum_{\substack{i \in \mathbb{V} \setminus \mathbb{T} \\ (j,j') \in \mathbb{T}^2, j'>j}} \rho_{i,j,j'} |x_{i,j} - x_{i,j'}|$$



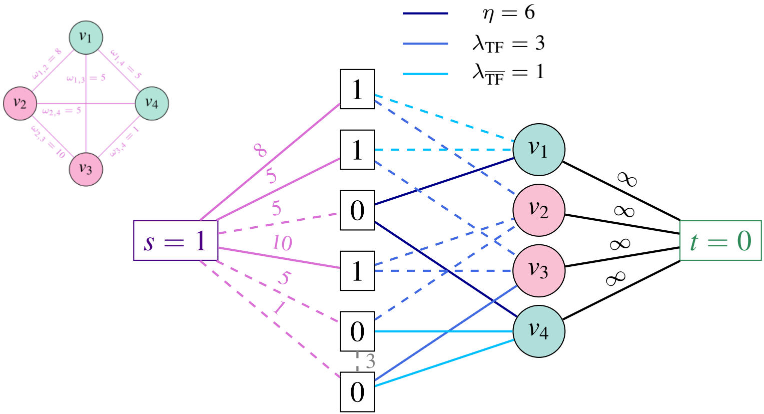
A maximal flow for a minimum cut formulation

$$\underset{x\in\{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}|x_{i,j}-1| + \lambda_{i,j}\, x_{i,j} + \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T}\\ (j,j')\in\mathbb{T}^2, j'>j}} \rho_{i,j,j'}|x_{i,j}-x_{i,j'}|$$



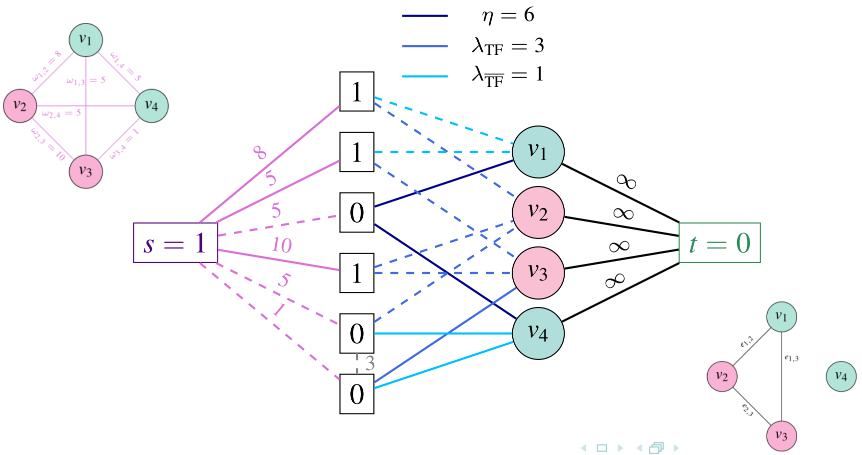
A maximal flow for a minimum cut formulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}|x_{i,j}-1| \;+\; \lambda_{i,j}\, x_{i,j} \;+ \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T}\\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\,|x_{i,j}-x_{i,j'}|$$



A maximal flow for a minimum cut formulation

$$\underset{x \in \{0,1\}^E}{\text{minimize}} \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}|x_{i,j}-1| \;+\; \lambda_{i,j}\, x_{i,j} \;+ \sum_{\substack{i\in\mathbb{V}\setminus\mathbb{T} \\ (j,j')\in\mathbb{T}^2,\, j'>j}} \rho_{i,j,j'}\,|x_{i,j}-x_{i,j'}|$$



A continuous method: *BRANE Relax*

We look for a continuous solution for $\boldsymbol{x} \Leftrightarrow \boldsymbol{x} \in [0, 1]^E$

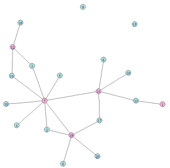
A priori: modular structure and $\overline{\text{TF}}$ connectivity

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \;+\; \lambda_{i,j}\varphi(x_{i,j}) \;+\; \mu\Psi(x_{i,j})$$

A priori: modular structure and $\overline{\text{TF}}$ connectivity

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \; + \; \lambda_{i,j}\varphi(x_{i,j}) \; + \; \mu\Psi(x_{i,j})$$

- Modular network: favors links between TFs and $\overline{\text{TFs}}$

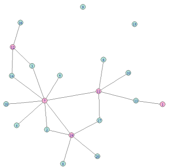


$$\lambda_{i,j} = \begin{cases} 2\eta & \text{if } (i,j) \notin \mathbb{T}^2 \\ 2\lambda_{\text{TF}} & \text{if } (i,j) \in \mathbb{T}^2 \\ \lambda_{\text{TF}} + \lambda_{\overline{\text{TF}}} & \text{otherwise.} \end{cases}$$

A priori: modular structure and $\overline{\text{TF}}$ connectivity

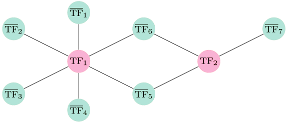
$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\varphi(x_{i,j}-1) \; + \; \lambda_{i,j}\varphi(x_{i,j}) \; + \; \mu\Psi(x_{i,j})$$

- Modular network: favors links between TFs and $\overline{\text{TFs}}$



$$\lambda_{i,j} = \begin{cases} 2\eta & \text{if } (i,j) \notin \mathbb{T}^2 \\ 2\lambda_{\text{TF}} & \text{if } (i,j) \in \mathbb{T}^2 \\ \lambda_{\text{TF}} + \lambda_{\overline{\text{TF}}} & \text{otherwise.} \end{cases}$$

- $\overline{\text{TF}}$ connectivity: constraint $\overline{\text{TF}}$ node degree



$$\Psi(x_{i,j}) = \sum_{i\in\mathbb{V}\setminus\mathbb{T}} \phi\left(\sum_{j\in\mathbb{V}} x_{i,j} - d\right)$$

$\phi(\cdot)$: a convex distance function with β-Lipschitz continuous gradient

A convex relaxation for a continuous formulation

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \omega_{i,j}(1-x_{i,j}) \ + \ \lambda_{i,j}\, x_{i,j} \ + \ \mu \sum_{i\in\mathbb{V}\setminus\mathbb{T}} \phi\left(\sum_{j\in\mathbb{V}} x_{i,j} - d\right)$$

A convex relaxation for a continuous formulation

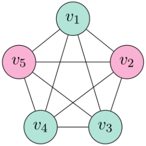
$$\underset{x \in \{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}(1 - x_{i,j}) \; + \; \lambda_{i,j}\, x_{i,j} \; + \; \mu \sum_{i\in\mathbb{V}\setminus\mathbb{T}} \phi\left(\sum_{j\in\mathbb{V}} x_{i,j} - d\right)$$

- Relaxation and vectorization:

$$\underset{\boldsymbol{x}\in[0,1]^E}{\text{minimize}} \quad \sum_{l=1}^{E} \omega_l(1 - x_l) \; + \; \lambda_l\, x_l \; + \; \mu \sum_{i=1}^{P} \phi\left(\sum_{k=1}^{E} \Omega_{i,k} x_k - d\right),$$

where $\boldsymbol{\Omega} \in \{0,1\}^{P\times E}$ encodes the degree of the P $\overline{\text{TFs}}$ nodes in the complete graph.

$$\Omega_{i,j} = \begin{cases} 1 & \text{if } j \text{ is the index of an edge linking the } \overline{\text{TF}} \text{ node } v_i \text{ in the complete graph,} \\ 0 & \text{otherwise.} \end{cases}$$



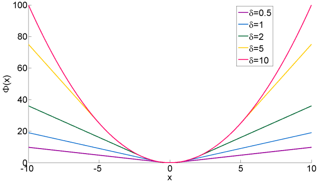
$$\boldsymbol{\Omega} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \quad \boldsymbol{\Omega x} = \begin{pmatrix} x_1 + x_2 + x_3 + x_4 \\ x_2 + x_5 + x_8 + x_9 \\ x_3 + x_6 + x_8 + x_{10} \end{pmatrix}$$

Distance function in *BRANE Relax*

$$\underset{\boldsymbol{x}\in[0,1]^E}{\text{minimize}} \quad \sum_{l=1}^{E} \omega_l(1-x_l) \;+\; \lambda_l x_l \;+\; \mu \sum_{i=1}^{P} \phi\left(\sum_{k=1}^{E} \Omega_{i,k} x_k - d\right)$$

Choice of ϕ: node degree distance function, with respect to d

- $z_i = \sum_{k=1}^{E} \Omega_{i,k} x_k - d$
- squared ℓ_2 norm: $\phi(\boldsymbol{z}) = ||\boldsymbol{z}||^2$
- Huber function: $\phi(z_i) = \begin{cases} z_i^2 & \text{if } |z_i| \le \delta \\ 2\delta(|z_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$



Optimization strategy *via* proximal methods

- Splitting

$$\boxed{\underset{\boldsymbol{x}\in\mathbb{R}^E}{\text{minimize}} \;\; \underbrace{\boldsymbol{\omega}^\top(\mathbf{1}_E - \boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{x} + \mu\Phi(\boldsymbol{\Omega}\boldsymbol{x} - \mathbf{d})}_{f_2} + \underbrace{\iota_{[0,1]^E}(\boldsymbol{x})}_{f_1}}$$

- $f_1 \in \Gamma_0(\mathbb{R}^E)$: proper, convex, and lower semi-continuous
- f_2: convex, differentiable with an $L-$Lipschitz continuous gradient

Optimization strategy *via* proximal methods

- Splitting

$$\boxed{\underset{\boldsymbol{x}\in\mathbb{R}^E}{\text{minimize}} \quad \underbrace{\boldsymbol{\omega}^\top(\mathbf{1}_E - \boldsymbol{x}) + \boldsymbol{\lambda}^\top\boldsymbol{x} + \mu\Phi(\boldsymbol{\Omega}\boldsymbol{x} - \mathbf{d})}_{f_2} + \underbrace{\iota_{[0,1]^E}(\boldsymbol{x})}_{f_1}}$$

- $f_1 \in \Gamma_0(\mathbb{R}^E)$: proper, convex, and lower semi-continuous
- f_2: convex, differentiable with an $L-$Lipschitz continuous gradient

Algorithm 1: Forward-Backward

Fix $\boldsymbol{x}_0 \in \mathbb{R}^E$
for $k = 0, 1, \ldots$ **do**

$$\boldsymbol{z}_k = \boldsymbol{x}_k - \gamma_k \nabla f_2(\boldsymbol{x}_k)$$
$$\boldsymbol{x}_{k+1} = \text{prox}_{\gamma_k f_1}(\boldsymbol{z}_k)$$

Optimization strategy *via* proximal methods

- Splitting

$$\underset{\boldsymbol{x}\in\mathbb{R}^E}{\text{minimize}} \quad \underbrace{\boldsymbol{\omega}^\top(\mathbf{1}_E - \boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{x} + \mu\Phi(\boldsymbol{\Omega}\boldsymbol{x} - \mathbf{d})}_{f_2} + \underbrace{\iota_{[0,1]^E}(\boldsymbol{x})}_{f_1}$$

- $f_1 \in \Gamma_0(\mathbb{R}^E)$: proper, convex, and lower semi-continuous
- f_2: convex, differentiable with an $L-$Lipschitz continuous gradient

Algorithm 2: Preconditioned Forward-Backward

Fix $\boldsymbol{x}_0 \in \mathbb{R}^E$

for $k = 0, 1, \ldots$ **do**

$$\boldsymbol{z}_k = \boldsymbol{x}_k - \gamma_k A^{-1}\nabla f_2(\boldsymbol{x}_k)$$

$$\boldsymbol{x}_{k+1} = \text{prox}_{\gamma_k{}^{-1}, A, f_1}(\boldsymbol{z}_k)$$

Optimization strategy *via* proximal methods

- Splitting

$$\boxed{\underset{\boldsymbol{x}\in\mathbb{R}^E}{\text{minimize}} \quad \underbrace{\boldsymbol{\omega}^\top(\mathbf{1}_E - \boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{x} + \mu\Phi(\boldsymbol{\Omega}\boldsymbol{x} - \mathbf{d})}_{f_2} + \underbrace{\iota_{[0,1]^E}(\boldsymbol{x})}_{f_1}}$$

- $f_1 \in \Gamma_0(\mathbb{R}^E)$: proper, convex, and lower semi-continuous
- f_2: convex, differentiable with an $L-$Lipschitz continuous gradient

Algorithm 3: Block Coordinate + Preconditioned Forward-Backward

Fix $\boldsymbol{x}_0 \in \mathbb{R}^E$

for $k = 0, 1, \ldots$ **do**

Select the index $j_k \in \{1, \ldots, J\}$ of a block of variables

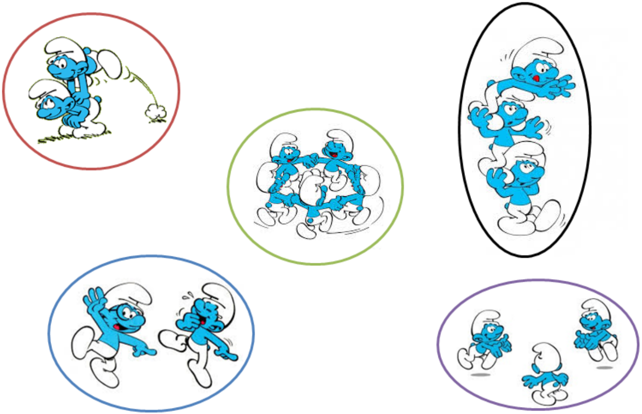
$\boldsymbol{z}_k^{(j_k)} = \boldsymbol{x}_k^{(j_k)} - \gamma_k A_{j_k}^{-1} \nabla_{j_k} f_2(\boldsymbol{x}_k)$

$\boldsymbol{x}_{k+1}^{(j_k)} = \operatorname{prox}_{\gamma_k^{-1}, A_{j_k}, f_1^{(j_k)}}(\boldsymbol{z}_k^{(j_k)})$

$\boldsymbol{x}_{k+1}^{(\bar{j}_k)} = \boldsymbol{x}_k^{(\bar{j}_k)}, \quad \bar{j}_k = \{1, \ldots, J\} \backslash \{j_k\}$

A mixed method: *BRANE Clust*

We look for a discrete solution for $\boldsymbol{x}$ and a continuous one for $\boldsymbol{y}$



A priori: gene grouping and modular structure

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^E \\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} f(y_i, y_j)\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \Psi(y_i)$$

A priori: gene grouping and modular structure

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^E \\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} f(y_i,y_j)\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \Psi(y_i)$$

- Clustering-assisted inference
  - Node labeling $\boldsymbol{y} \in \mathbb{N}^G$
  - Weight $\omega_{i,j}$ reduction if nodes v_i and v_j belong to distinct clusters
  - Cost function:

$$f(y_i,y_j) = \frac{\beta - \mathbb{1}(y_i \neq y_j)}{\beta}$$

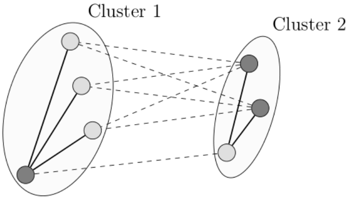
A priori: gene grouping and modular structure

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^E \\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} f(y_i, y_j)\omega_{i,j}x_{i,j} \; + \; \lambda(1 - x_{i,j}) \; + \; \Psi(y_i)$$

- Clustering-assisted inference
  - Node labeling $\boldsymbol{y} \in \mathbb{N}^G$
  - Weight $\omega_{i,j}$ reduction if nodes v_i and v_j belong to distinct clusters
  - Cost function:

$$f(y_i, y_j) = \frac{\beta - \mathbb{1}(y_i \neq y_j)}{\beta}$$

- TF-driven clustering promoting modular structure



$$\Psi(y_i) = \sum_{\substack{i\in\mathbb{V} \\ j\in\mathbb{T}}} \mu_{i,j}\mathbb{1}(y_i = j)$$

$\mu_{i,j}$: modular structure controlling parameter

Alternating optimization strategy

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^n \\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \sum_{\substack{i\in\mathbb{V}\\ j\in\mathbb{T}}}\mu_{i,j}\mathbb{1}(y_i=j)$$

Alternating optimization strategy

Alternating optimization

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^n\\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \sum_{\substack{i\in\mathbb{V}\\ j\in\mathbb{T}}} \mu_{i,j}\mathbb{1}(y_i=j)$$

Alternating optimization strategy

Alternating optimization

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^n\\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \sum_{\substack{i\in\mathbb{V}\\ j\in\mathbb{T}}} \mu_{i,j}\mathbb{1}(y_i=j)$$

- At $\boldsymbol{y}$ fixed and $\boldsymbol{x}$ variable:

$$\underset{\boldsymbol{x}\in\{0,1\}^n}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\,\omega_{i,j}\,x_{i,j} \quad + \quad \lambda(1-x_{i,j})$$

Alternating optimization strategy

Alternating optimization

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^n\\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \sum_{\substack{i\in\mathbb{V}\\ j\in\mathbb{T}}} \mu_{i,j}\mathbb{1}(y_i=j)$$

- At $\boldsymbol{y}$ fixed and $\boldsymbol{x}$ variable:

$$\underset{\boldsymbol{x}\in\{0,1\}^n}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\,\omega_{i,j}\,x_{i,j} \quad + \quad \lambda(1-x_{i,j})$$

- At $\boldsymbol{x}$ fixed and $\boldsymbol{y}$ variable:

$$\underset{\boldsymbol{y}\in\mathbb{N}^G}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\,x_{i,j}}{\beta}\mathbb{1}(y_i\neq y_j) \;+\; \sum_{i\in\mathbb{V},j\in\mathbb{T}} \mu_{i,j}\,\mathbb{1}(y_i\neq j)$$

Alternating optimization strategy

Alternating optimization

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^n\\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \sum_{\substack{i\in\mathbb{V}\\ j\in\mathbb{T}}}\mu_{i,j}\mathbb{1}(y_i=j)$$

- At $\boldsymbol{y}$ fixed and $\boldsymbol{x}$ variable:

$$\underset{\boldsymbol{x}\in\{0,1\}^n}{\text{maximize}} \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta-\mathbb{1}(y_i\neq y_j)}{\beta}\,\omega_{i,j}\,x_{i,j} \;+\; \lambda(1-x_{i,j})$$

Explicit form: $x^*_{i,j} = \begin{cases} 1 & \text{if } \omega_{i,j} > \frac{\lambda\beta}{\beta-\mathbb{1}(y_i\neq y_j)} \\ 0 & \text{otherwise.} \end{cases}$

- At $\boldsymbol{x}$ fixed and $\boldsymbol{y}$ variable:

$$\underset{\boldsymbol{y}\in\mathbb{N}^G}{\text{minimize}} \sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\,x_{i,j}}{\beta}\mathbb{1}(y_i\neq y_j) \;+\; \sum_{i\in\mathbb{V},\,j\in\mathbb{T}} \mu_{i,j}\,\mathbb{1}(y_i\neq j)$$

Clustering optimization strategy

- At $\boldsymbol{x}$ fixed and $\boldsymbol{y}$ variable:

$$\underset{\mathbf{y}\in\mathbb{N}^G}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \mathbb{1}(y_i \neq y_j) + \sum_{i\in\mathbb{V},\ j\in\mathbb{T}} \mu_{i,j} \mathbb{1}(y_i \neq j)$$

Clustering optimization strategy

- At $\boldsymbol{x}$ fixed and $\boldsymbol{y}$ variable:

$$\underset{\mathbf{y}\in\mathbb{N}^G}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \mathbb{1}(y_i \neq y_j) + \sum_{i\in\mathbb{V},\ j\in\mathbb{T}} \mu_{i,j} \mathbb{1}(y_i \neq j) \tag{NP}$$

Clustering optimization strategy

- At $\boldsymbol{x}$ fixed and $\boldsymbol{y}$ variable:

$$\underset{\mathbf{y}\in\mathbb{N}^G}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \mathbb{1}(y_i \neq y_j) + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j} \mathbb{1}(y_i \neq j) \qquad \text{(NP)}$$

- discrete problem $\Rightarrow$ quadratic relaxation
- T-class problem $\Rightarrow$ T binary sub-problems
  - label restriction to $\mathbb{T}$: $\{s^{(1)}, \ldots, s^{(T)}\}$ such that $s_j^{(t)} = 1$ if $j = t$ and 0 otherwise.
  - $\mathcal{Y} = \{y^{(1)}, \ldots, y^{(T)}\}$ such that $y^{(t)} \in [0, 1]^G$

Clustering optimization strategy

- At $\boldsymbol{x}$ fixed and $\boldsymbol{y}$ variable:

$$\underset{\mathbf{y}\in\mathbb{N}^G}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \mathbb{1}(y_i \neq y_j) + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j} \mathbb{1}(y_i \neq j) \qquad \text{(NP)}$$

- discrete problem $\Rightarrow$ quadratic relaxation
- T-class problem $\Rightarrow$ T binary sub-problems
  - label restriction to $\mathbb{T}$: $\{s^{(1)}, \dots, s^{(T)}\}$ such that $s_j^{(t)} = 1$ if $j = t$ and 0 otherwise.
  - $\mathcal{Y} = \{y^{(1)}, \dots, y^{(T)}\}$ such that $y^{(t)} \in [0,1]^G$

Problem re-expressed as:

$$\underset{\mathcal{Y}}{\text{minimize}} \quad \sum_{t=1}^{T} \left(\sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \left(y_i^{(t)} - y_j^{(t)}\right)^2 + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j} \left(y_i^{(t)} - s_j^{(t)}\right)^2 \right)$$

Clustering optimization strategy

$$\underset{\mathcal{Y}}{\text{minimize}} \sum_{t=1}^{T} \left(\sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \left(y_i^{(t)} - y_j^{(t)} \right)^2 + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j} \left(y_i^{(t)} - s_j^{(t)} \right)^2 \right)$$

- This is the Combinatorial Dirichlet problem
- Minimization *via* solving a linear system of equations *[Grady, 2006]*

Clustering optimization strategy

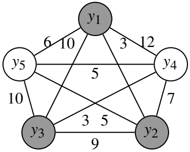
$$\underset{\mathcal{Y}}{\text{minimize}} \quad \sum_{t=1}^{T} \left(\sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \left(y_i^{(t)} - y_j^{(t)} \right)^2 + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j} \left(y_i^{(t)} - s_j^{(t)} \right)^2 \right)$$

- This is the Combinatorial Dirichlet problem
- Minimization *via* solving a linear system of equations *[Grady, 2006]*
- Final labeling: node i is assigned to label t for which $y_i^{(t)}$ is maximal

$$y_i^* = \underset{t\in\mathbb{T}}{\text{argmax}} \quad y_i^{(t)}$$

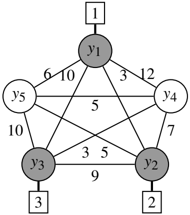
Random walker in graphs

We want to obtain the optimal labeling $\mathbf{y}^*$ based on a weighted graph ⇒ Random Walker algorithm



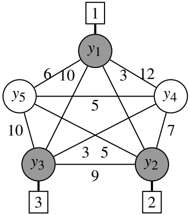
Random walker in graphs

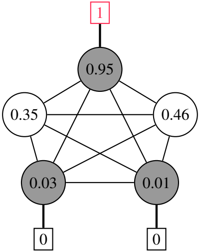
We want to obtain the optimal labeling $\mathbf{y}^*$ based on a weighted graph ⇒ Random Walker algorithm



Random walker in graphs

We want to obtain the optimal labeling $\mathbf{y}^*$ based on a weighted graph ⇒ Random Walker algorithm

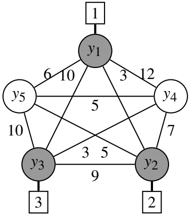


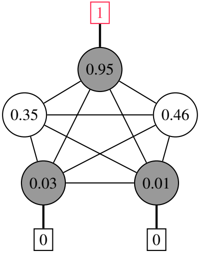


$y^{(1)}$

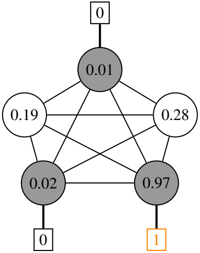
Random walker in graphs

We want to obtain the optimal labeling $\mathbf{y}^*$ based on a weighted graph $\Rightarrow$ Random Walker algorithm





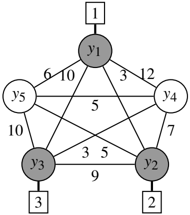
$y^{(1)}$

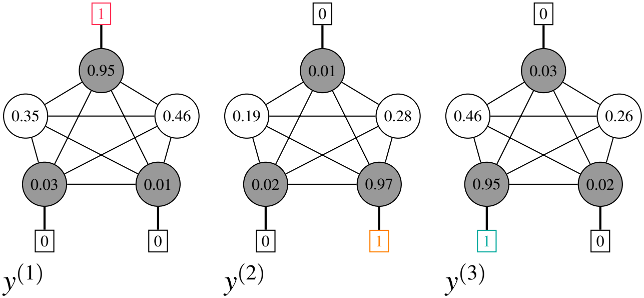


$y^{(2)}$

Random walker in graphs

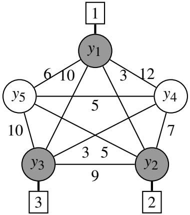
We want to obtain the optimal labeling $\mathbf{y}^*$ based on a weighted graph $\Rightarrow$ Random Walker algorithm

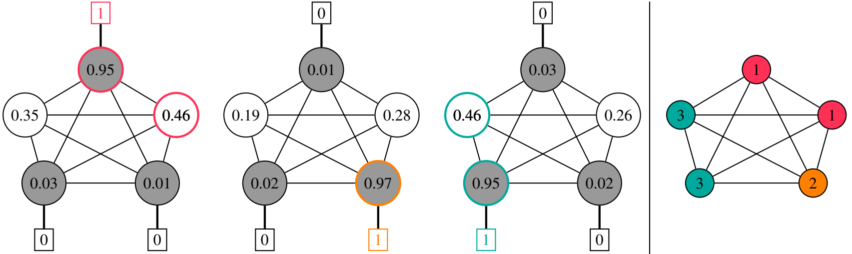




Random walker in graphs

We want to obtain the optimal labeling $\mathbf{y}^*$ based on a weighted graph $\Rightarrow$ Random Walker algorithm



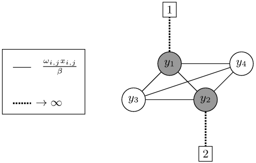
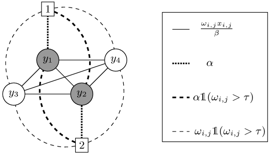


$y^{(1)}$ $y^{(2)}$ $y^{(3)}$

$$y^* = \{1, 1, 2, 3, 3\}$$

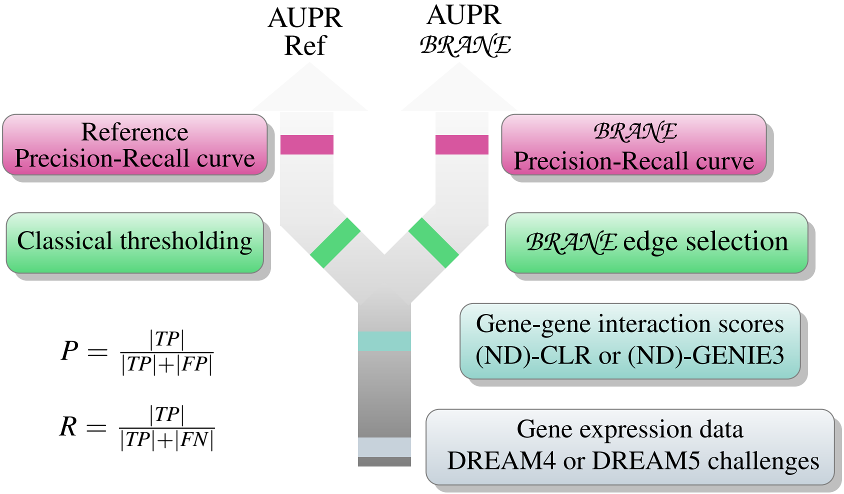
hard- vs *soft-* clustering in *BRANE Clust*

$$\underset{\mathcal{Y}}{\text{minimize}} \quad \sum_{t=1}^{T} \left(\sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}\, x_{i,j}}{\beta} \left(y_i^{(t)} - y_j^{(t)}\right)^2 + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j} \left(y_i^{(t)} - s_j^{(t)}\right)^2 \right)$$

hard-clustering	*soft*-clustering
# clusters $=$ # TF	# clusters $\leq$ # TF
$\mu_{i,j} = \begin{cases} \to \infty & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$	$\mu_{i,j} = \begin{cases} \alpha & \text{if } i = j \\ \alpha \mathbb{1}(\omega_{i,j} > \tau) & \text{if } i \neq j \text{ and } i \in \mathbb{T} \\ \omega_{i,j} \mathbb{1}(\omega_{i,j} > \tau) & \text{if } i \neq j \text{ and } i \notin \mathbb{T} \end{cases}$
	

It's time to test the *BRANE* philosophy...

Numerical evaluation strategy



$$P = \frac{|TP|}{|TP|+|FP|}$$

$$R = \frac{|TP|}{|TP|+|FN|}$$

BRANE performance on *in-silico* data

- DREAM4 *[Marbach et al., 2010]*

BRANE performance on *in-silico* data

- DREAM4 *[Marbach et al., 2010]*

Network	1	2	3	4	5	Average	Gain
CLR	0.256	0.275	0.314	0.313	0.318	0.295	
BRANE Cut	**0.282**	0.308	0.343	**0.344**	**0.356**	0.327	**10.9 %**
BRANE Relax	0.278	0.293	0.336	0.333	0.345	0.317	**7.8 %**
BRANE Clust	0.275	**0.337**	**0.360**	0.335	0.342	**0.330**	**12.2 %**
GENIE3	0.269	0.288	0.331	0.323	0.329	0.308	
BRANE Cut	**0.298**	0.316	0.357	0.344	0.352	0.333	**8.4 %**
BRANE Relax	0.293	0.320	0.356	0.345	0.354	0.334	**8.5 %**
BRANE Clust	0.287	**0.348**	**0.364**	**0.371**	**0.367**	**0.347**	**12.8 %**

Network	1	2	3	4	5	Average	Gain
ND-CLR	0.254	0.250	0.324	0.318	0.331	0.295	
BRANE Cut	**0.271**	**0.277**	**0.334**	0.335	**0.343**	**0.312**	**5.9 %**
BRANE Relax	0.270	0.264	0.327	0.325	0.332	0.304	**3.1 %**
BRANE Clust	0.258	0.251	0.327	**0.337**	0.342	0.303	**2.5 %**
ND-GENIE3	0.263	0.275	0.336	0.328	0.354	0.309	
BRANE Cut	0.275	**0.312**	0.367	0.346	0.368	0.334	**7.2 %**
BRANE Relax	**0.276**	0.307	**0.369**	0.347	**0.371**	0.334	**7.3 %**
BRANE Clust	0.273	0.311	0.354	**0.373**	0.370	**0.336**	**8.1 %**

BRANE performance on *in-silico* data

- DREAM4 *[Marbach et al., 2010]*

	CLR	GENIE3	ND-CLR	ND-GENIE3
BRANE Cut	10.9 %	8.4 %	5.9 %	7.2 %
BRANE Relax	7.8 %	8.5 %	3.1 %	7.3 %
BRANE Clust	12.2 %	12.8 %	2.5 %	8.1 %

- *BRANE* approaches validated on small synthetic data
- *BRANE* methodologies outperform classical thresholding
- First and second best performers: *BRANE Clust* and *BRANE Cut*

$\Rightarrow$ Validation on more realistic synthetic data

BRANE performance on *in-silico* data

- DREAM5 *[Marbach et al., 2012]*

BRANE performance on *in-silico* data

- DREAM5 *[Marbach et al., 2012]*

	AUPR	Gain		AUPR	Gain
CLR	0.252		GENIE3	0.283	
BRANE Cut	0.268	**6.3 %**	*BRANE Cut*	0.295	**4.2 %**
BRANE Relax	0.272	**7.9 %**	*BRANE Relax*	0.294	**3.8 %**
BRANE Clust	**0.301**	**19.4 %**	*BRANE Clust*	**0.336**	**18.6 %**

	AUPR	Gain		AUPR	Gain
ND-CLR	0.272		ND-GENIE3	0.313	
BRANE Cut	0.277	**1.9 %**	*BRANE Cut*	0.317	**1.1 %**
BRANE Relax	0.274	**0.6 %**	*BRANE Relax*	0.314	**0.3 %**
BRANE Clust	**0.289**	**6.2 %**	*BRANE Clust*	**0.345**	**10.2 %**

BRANE performance on *in-silico* data

- DREAM5 *[Marbach et al., 2012]*

	AUPR	Gain		AUPR	Gain
CLR	0.252		GENIE3	0.283	
BRANE Cut	0.268	6.3 %	*BRANE Cut*	0.295	4.2 %
BRANE Relax	0.272	7.9 %	*BRANE Relax*	0.294	3.8 %
BRANE Clust	**0.301**	19.4 %	*BRANE Clust*	**0.336**	18.6 %

	AUPR	Gain		AUPR	Gain
ND-CLR	0.272		ND-GENIE3	0.313	
BRANE Cut	0.277	1.9 %	*BRANE Cut*	0.317	1.1 %
BRANE Relax	0.274	0.6 %	*BRANE Relax*	0.314	0.3 %
BRANE Clust	**0.289**	6.2 %	*BRANE Clust*	**0.345**	10.2 %

- *BRANE* approaches validated on realistic synthetic data and outperform classical thresholding
- First and second best performer: *BRANE Clust* and *BRANE Cut*

⇒ Validation of *BRANE Cut* and *BRANE Clust* on real data

BRANE Clust performance on *real* data

- *Escherichia coli dataset*

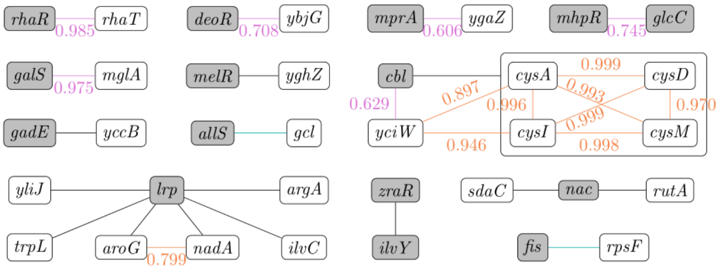
	AUPR	Gain		AUPR	Gain
CLR	0.0378	**5.5 %**	GENIE3	0.0488	**9.8 %**
BRANE Clust	0.0399		*BRANE Clust*	0.0536	

BRANE Clust performance on *real* data

- *Escherichia coli dataset*

	AUPR	Gain		AUPR	Gain
CLR	0.0378	**5.5 %**	GENIE3	0.0488	**9.8 %**
BRANE Clust	0.0399		*BRANE Clust*	0.0536	

- *BRANE Clust* predictions using GENIE3 weights

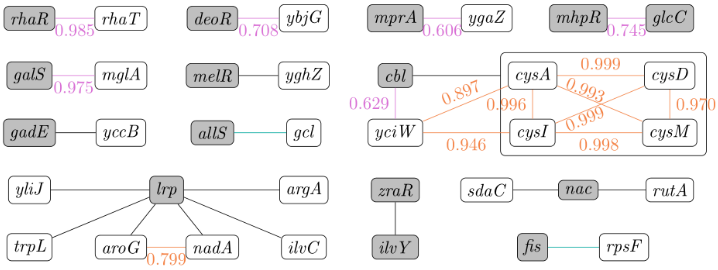


BRANE Clust performance on *real* data

- *Escherichia coli dataset*

	AUPR	Gain		AUPR	Gain
CLR	0.0378	**5.5 %**	GENIE3	0.0488	**9.8 %**
BRANE Clust	0.0399		*BRANE Clust*	0.0536	

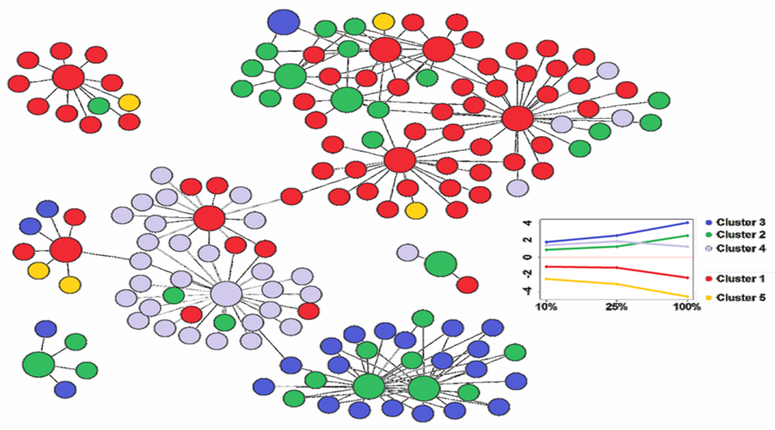
- *BRANE Clust* predictions using GENIE3 weights



- *BRANE Clust* validated on real dataset

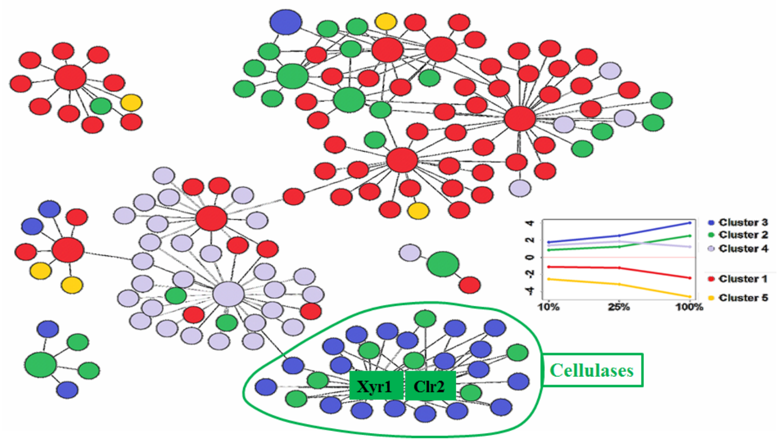
BRANE Cut in the real life

- GRN of *T. reesei* obtained with *BRANE Cut* using CLR weights



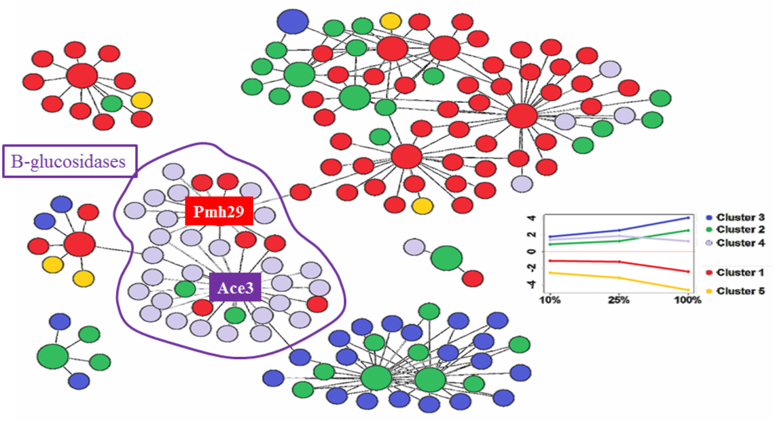
BRANE Cut in the real life

- GRN of *T. reesei* obtained with *BRANE Cut* using CLR weights



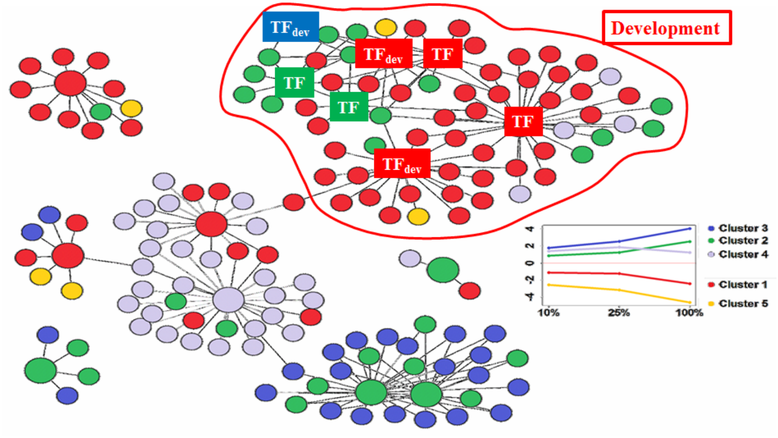
BRANE Cut in the real life

- GRN of *T. reesei* obtained with *BRANE Cut* using CLR weights



BRANE Cut in the real life

- GRN of *T. reesei* obtained with *BRANE Cut* using CLR weights



It's time to conclude...

Conclusions

Inference: *BRANE Cut* and *BRANE Relax*
Joint inference and clustering: *BRANE Clust*

Conclusions

Inference: *BRANE Cut* and *BRANE Relax*
Joint inference and clustering: *BRANE Clust*

- *The BRA- in BRANE*: integrating biological *a priori* constrains the search of relevant edges
- *The -NE in BRANE*: proposed graph inference methods lead to promising results and outperforms state-of-the-art methods

Conclusions

Inference: *BRANE Cut* and *BRANE Relax*
Joint inference and clustering: *BRANE Clust*

- *The BRA- in BRANE*: integrating biological *a priori* constrains the search of relevant edges
- *The -NE in BRANE*: proposed graph inference methods lead to promising results and outperforms state-of-the-art methods

$\Rightarrow$ Average improvements around 10 %
$\Rightarrow$ Biological relevant inferred networks
$\Rightarrow$ Negligible time complexity with respect to graph weight computation

Conclusions

Inference: *BRANE Cut* and *BRANE Relax*
Joint inference and clustering: *BRANE Clust*

- *The BRA- in BRANE*: integrating biological *a priori* constrains the search of relevant edges
- *The -NE in BRANE*: proposed graph inference methods lead to promising results and outperforms state-of-the-art methods

$\Rightarrow$ Average improvements around 10 %
$\Rightarrow$ Biological relevant inferred networks
$\Rightarrow$ Negligible time complexity with respect to graph weight computation

- Biological *a priori* relevance for network inference

BRANE Clust $\succ$ *BRANE Cut* $\succ$ *BRANE Relax*

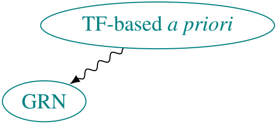
Perspectives

From biological graphs...

GRN

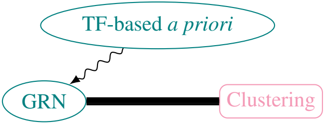
Perspectives

From biological graphs...



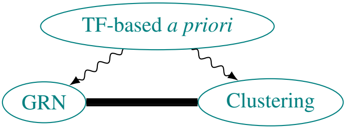
Perspectives

From biological graphs...



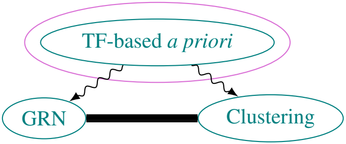
Perspectives

From biological graphs...



Perspectives

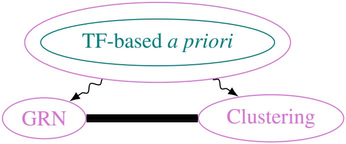
From biological graphs...



- Extend TF-based *a priori* for

Perspectives

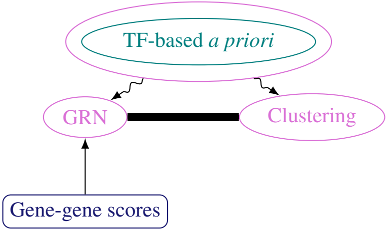
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering

Perspectives

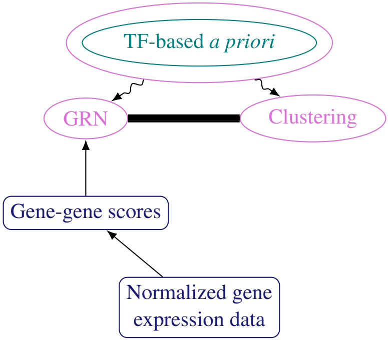
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering

Perspectives

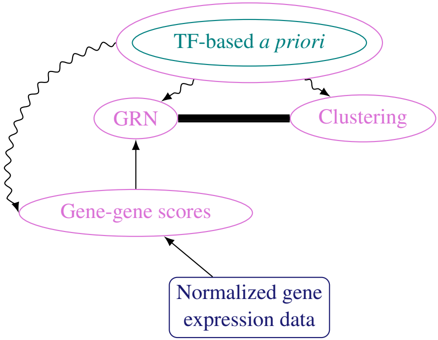
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering

Perspectives

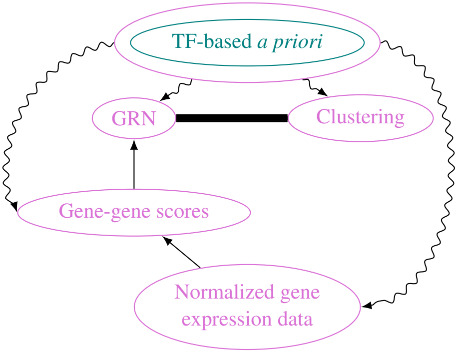
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering , graph weighting,

Perspectives

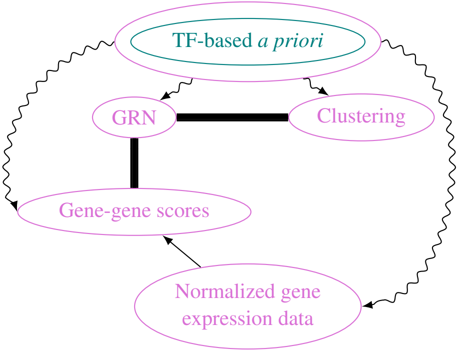
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering , graph weighting, data normalization...

Perspectives

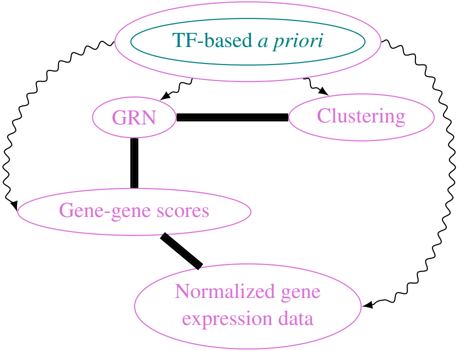
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering , graph weighting, data normalization...
- Integrate transcriptomic data treatment

Perspectives

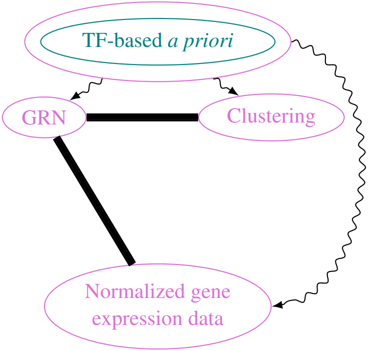
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering , graph weighting, data normalization...
- Integrate transcriptomic data treatment

Perspectives

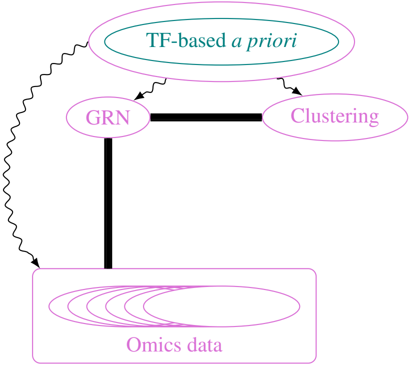
From biological graphs...



- Extend TF-based *a priori* for GRN, clustering , graph weighting, data normalization...
- Integrate transcriptomic data treatment

Perspectives

From biological graphs...



- Extend TF-based *a priori* for GRN, clustering , graph weighting, data normalization...
- Integrate transcriptomic data treatment
- Integrate *a priori*, omics- data and treatments

Perspectives

... to general graphs

- *BRANE*-like applications for non biological graphs
  - Coupled edge inference: social networks
  - Node-degree constraint: telecommunication
  - Coupling between inference and clustering: temperature networks, brain networks

Perspectives

... to general graphs

- *BRANE*-like applications for non biological graphs
  - Coupled edge inference: social networks
  - Node-degree constraint: telecommunication
  - Coupling between inference and clustering: temperature networks, brain networks
- Topological constraint in graph inference
  - Expected node degree distribution
  - Scale-free networks: webgraphs, financial networks, social networks...

Perspectives

... to general graphs

- *BRANE*-like applications for non biological graphs
  - Coupled edge inference: social networks
  - Node-degree constraint: telecommunication
  - Coupling between inference and clustering: temperature networks, brain networks
- Topological constraint in graph inference
  - Expected node degree distribution
  - Scale-free networks: webgraphs, financial networks, social networks...
- Laplacian-based approach for graph comparison
  - Spectral view of the graph
  - Modularity
  - Local and topological-based criteria

Publications

Journal papers — published

D. Poggi-Parodi, F. Bidard, A. Pirayre, T. Portnoy. C. Blugeon, B. Seiboth, C.P. Kubicek, S. Le Crom and A. Margeot
Kinetic transcriptome reveals an essentially intact induction system in a cellu- lase hyper-producer Trichoderma reesei strain
Biotechnology for Biofuels, 7:173, Dec. 2014

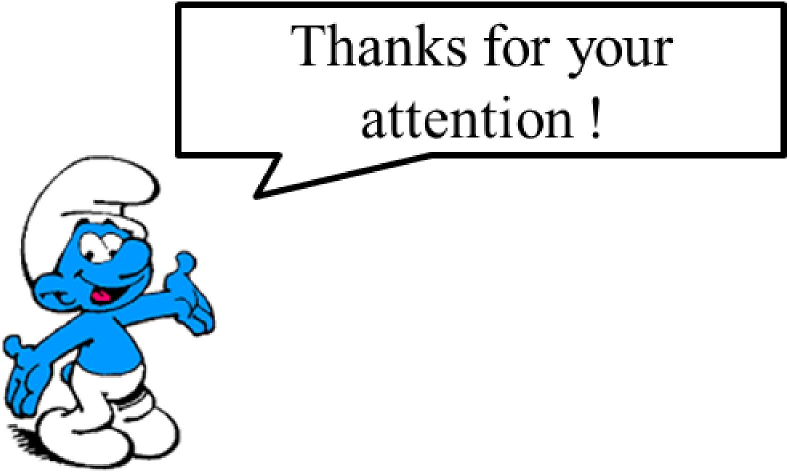
A. Pirayre, C. Couprie, F. Bidard, L. Duval, and J.-C. Pesquet.
BRANE Cut: biologically-related *a priori* network enhancement with graph cuts for gene regulatory network inference
BMC Bioinformatics, 16(1):369, Dec. 2015.

A. Pirayre, C. Couprie, L. Duval, and J.-C. Pesquet.
BRANE Clust: Cluster-Assisted Gene Regulatory Network Inference Refinement
IEEE/ACM Transactions on Computational Biology and Bioinformatics, Mar. 2017.

Journal papers — in preparation

Y. Zheng, A. Pirayre, L. Duval and J.-C. Pesquet
Joint restoration/segmentation of multicomponent images with variational Bayes and higher-order graphical models (HOGMep)
To be submitted to *IEEE Transactions on Computational Imaging*, Jul. 2017.

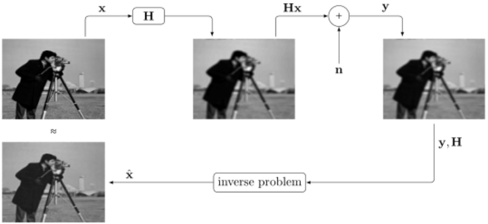
A. Pirayre, D. Ivanoff, L. Duval, C. Blugeon, C. Firmo, S. Perrin, E. Jourdier, A. Margeot and F. Bidard
Growing *Trichoderma reesei* on a mix of carbon sources suggests links between development and cellulase production
To be submitted to *BMC Genomics*, Jul. 2017.


Thanks for your
attention !

HOGMep for non-blind inverse problems

$$\boxed{\mathbf{y} = \mathbf{H}\,\mathbf{x} + \mathbf{n}}$$

- $\mathbf{x}$: unknown signal to be recovered
- $\mathbf{H}$: known degradation operator
- $\mathbf{n}$: additive noise
- $\mathbf{y}$: observations



HOGMep — Bayesian framework

- Estimation of $\mathbf{x}$ from the knowledge of the posterior pdf $p(\mathbf{x}|\mathbf{y})$

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})}$$

- $p(\mathbf{x})$: the marginal pdf encoding information about $\mathbf{x}$
- $p(\mathbf{y}|\mathbf{x})$: the likelihood highlighting the uncertainty in $\mathbf{y}$
- $p(\mathbf{y})$: the marginal pdf of $\mathbf{y}$

HOGMep — Variational Bayesian Approximation

- $q(\mathbf{x})$: approximation of $p(\mathbf{x}|\mathbf{y})$

$$q^{opt}(\mathbf{x}) = \underset{q(\mathbf{x})}{\operatorname{argmin}} \ \mathcal{KL}(q(\mathbf{x}) \,||\, p(\mathbf{x} \,|\, \mathbf{y}))$$

- Separable distribution:

$$q(\mathbf{x}) = \prod_{j=1}^{J} q_j(\mathbf{x}_j),$$

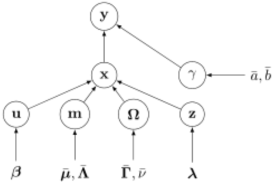
with

$$q_j^{opt}(\mathbf{x}_j) \ \propto \ \exp\left(\langle \ln p(\mathbf{y},\mathbf{x})\rangle_{\prod_{i\neq j} q_i(\mathbf{x}_i)}\right)$$

- Estimation of the distributions in an iterative manner

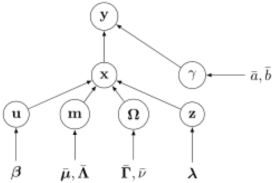
HOGMep — Bayesian formulation

- Likelihood prior: $p(\mathbf{y} \mid \mathbf{x}, \gamma) = \mathcal{N}(\mathbf{Hx}, \gamma^{-1}\mathbf{I})$
- $p(\mathbf{z})$: prior on hidden variables $\mathbf{z} \Rightarrow$ generalized Potts model
- $p(\mathbf{x}|\mathbf{z})$: prior on $\mathbf{x}$ conditionally to $\mathbf{z} \Rightarrow$ MEP distribution restricted to Gaussian Scale Mixtures $\mathcal{GSM}(\mathbf{m}, \boldsymbol{\Omega}, \beta)$
- Hyperpriors: $p(\gamma)$, $p(\mathbf{m}_l)$ and $p(\boldsymbol{\Omega}_l)$



$\mathcal{HOGM}ep$ — Bayesian formulation

- Likelihood prior: $p(\mathbf{y} \mid \mathbf{x}, \gamma) = \mathcal{N}(\mathbf{Hx}, \gamma^{-1}\mathbf{I})$
- $p(\mathbf{z})$: prior on hidden variables $\mathbf{z} \Rightarrow$ generalized Potts model
- $p(\mathbf{x}|\mathbf{z})$: prior on $\mathbf{x}$ conditionally to $\mathbf{z} \Rightarrow$ MEP distribution restricted to Gaussian Scale Mixtures $\mathcal{GSM}(\mathbf{m}, \boldsymbol{\Omega}, \beta)$
- Hyperpriors: $p(\gamma)$, $p(\mathbf{m}_l)$ and $p(\boldsymbol{\Omega}_l)$



- Joint posterior distribution

$$p(\mathbf{y} \mid \mathbf{x}, \gamma) \prod_{i=1}^{N} \Big(p(\mathbf{x}_i \mid z_i, u_i, \mathbf{m}, \boldsymbol{\Omega}) p(u_i \mid \beta) \Big) p(\mathbf{z}) p(\gamma) \prod_{l=1}^{L} p(\mathbf{m}_l) p(\boldsymbol{\Omega}_l)$$

$\mathcal{HOGM}ep$ — VBA strategy

- Separable form for the approximation:

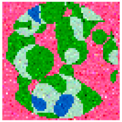
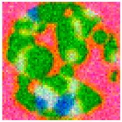
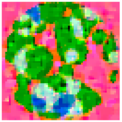
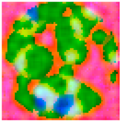
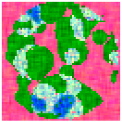
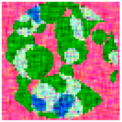
$$q(\Theta) = \prod_{i=1}^{N} \left(q(\mathbf{x}_i, z_i) q(u_i)\right) q(\gamma) \prod_{l=1}^{L} \left(q(\mathbf{m}_l) q(\boldsymbol{\Omega}_l)\right)$$

with

$$\begin{aligned} q(\mathbf{x}_i | z_i = l) &= \mathcal{N}(\boldsymbol{\eta}_{i,l}, \boldsymbol{\Xi}_{i,l}), \\ q(z_i = l) &= \pi_{i,l}, \\ q(\mathbf{m}_l) &= \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Lambda}_l), \\ q(\boldsymbol{\Omega}_l) &= \mathcal{W}(\boldsymbol{\Gamma}_l, \nu_l), \\ q(\gamma) &= \mathcal{G}(a, b). \end{aligned}$$

HOGMep — Some restoration results

- Restoration

Original	Degraded	DR	3MG	VB-MIG	*HOGMep*
					
SNR	6.655	9.467	6.744	12.737	**12.895**

Original	Degraded	DR	3MG	VB-MIG	*HOGMep*
					
SNR	19.659	18.728	17.188	15.486	**19.555**

HOGMep — Some segmentation results

- Segmentation