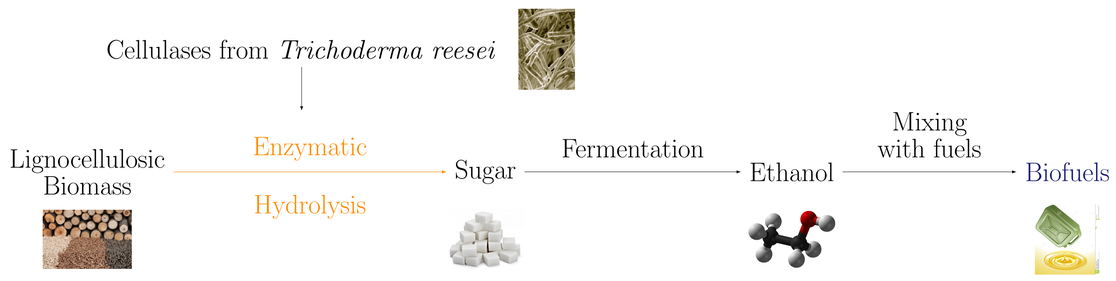
# Incorporating structural *a priori* in Gene Regulatory Network Inference using Graph cuts

Aurélie Pirayre[1,2], Camille Couprie[1], Laurent Duval[1], Frédérique Bidard[1], Jean-Christophe Pesquet[2]

[1] IFP Energies nouvelles
[2] Univ. Paris-Est, LIGM, UMR CNRS 8049

## Introduction




**Energetical context:**

Improving the production efficiency of the second generation biofuels by optimizing the enzymatic hydrolysis phase

**Biological context:**

Genetic target identification in *Trichoderma reesei* to improve the cellulase production, involved in the biofuel production process

**Mathematical context:**

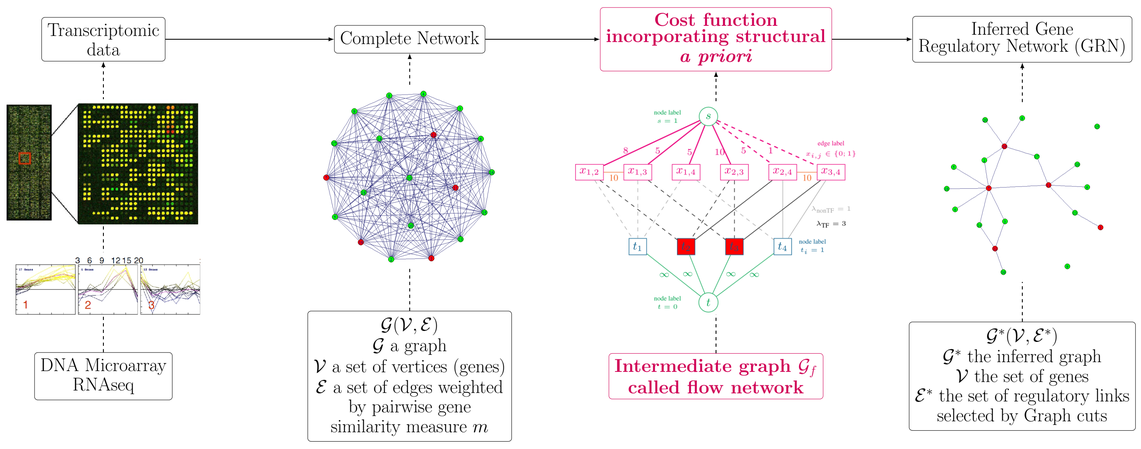
Novel algorithmic method based on graph optimization to infer Gene Regulatory Networks (GRNs) and identify new target genes

GRN: powerful tool to visualize gene interaction relationships from high-throughput data
Difficult problem: thousands of genes expressed in only few conditions

This last decade, very active community with DREAM challenge and many inference methods (RN, ARACNE, SIMoNe, NARROMI, CLR, GENIE3 ...)

## Global strategy

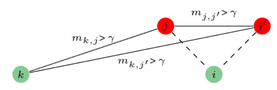
Inferring a GRN: recovering the interactions between the transcription factors and their target genes i.e. in the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, find a set of edges $\mathcal{E}^*(\subseteq \mathcal{E})$ reflecting regulatory links




## Proposed cost function

### Structural *a priori*

- Differential degree according to the nature of the nodes favoring TF-nonTF interactions
- Enforces co-regulation relationships




- Let $\mathcal{T} \subset \mathcal{V}$ a set of transcription factors (TFs) and $x_{i,j}$ be the binary label of the edges $e_{i,j}$ such that:

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

- Inference problem re-expressed as a cost function to be minimized:

### Cost function

$$\underset{\mathbf{x}\in\{0;1\}^n}{\text{minimize}} \underbrace{\sum_{\substack{(i,j)\in\mathcal{V}\times\mathcal{V} \\ i\neq j}} m_{i,j}|x_{i,j}-1|}_{\text{Disfavors the deletion of strongly weighted edges}} + \underbrace{\sum_{\substack{(i,j)\in\mathcal{V}\times\mathcal{V} \\ i\neq j}} \lambda_{i,j}x_{i,j}}_{\text{Favors the selection of edges linked to a transcription factor (TF)}} + \underbrace{\sum_{\substack{i\in\mathcal{V}\setminus\mathcal{T}, \\ (j,j')\in\mathcal{T}\times\mathcal{T}}} \alpha_{i,j,j'}|x_{i,j}-x_{i,j'}|}_{\text{Enforces the coupling of regulatory relationships}}.$$

## Optimization strategy

Thanks to the min-cut/max-flow duality, computing the optimal labeling $\mathbf{x}^*$ minimizing the above equation may be performed by a maximal flow algorithm on a flow network $\mathcal{G}_f$.

A flow $f$ is a function assigning a real value at each edge under two main constraints:

- Capacity constraint: the flow in each edge is less than the capacity (weight) of the edges
- Flow conservation: at each node, the entering flow equals the leaving flow

### The flow network $\mathcal{G}_f$

We used construction rules given by [3] to build the flow network $\mathcal{G}_f$ allowing us to compute $\mathbf{x}^*$:

- Two specific nodes: the source $s$ (0-in-degree) and the sink $t$ (0-out-degree)
- $n = |\mathcal{E}|$ nodes $v_{i,j}$ linked to the source $s$ and $p = |\mathcal{V}|$ nodes $g_i$ linked to the sink $t$

The capacities of the different edges in $\mathcal{G}_f$ are given by the different weights $m_{i,j}$, $\lambda_{i,j}$ and $\alpha_{i,j,j'}$ of the above equation. The edge saturation allows us to label the nodes $v_{i,j}$ of $\mathcal{G}_f$ with binary labels $x_{i,j}$:

- nodes $v_{i,j}$ linked to the source $s$ *via* a non-saturated path: $x_{i,j} = 1$
- nodes $v_{i,j}$ linked to the think $t$ *via* a non-saturated path: $x_{i,j} = 0$

With respect to the two constraints on the flow, finding the maximal flow from $s$ to $t$ in the flow network $\mathcal{G}_f$, give us the optimal labeling $\mathbf{x}^*$ according to the cost function

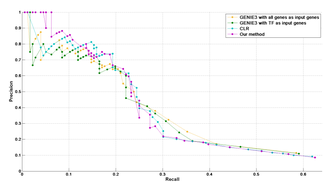
## Results

Comparison to two state-of-the-art methods: CLR [1] and GENIE3 [2] on two kinds of dataset: DREAM4 [4] (in silico multifactorial challenge) and a real dataset of *Escherichia coli* also used in [1]. The evaluation is performed computing Precision and Recall for each inferred graph.

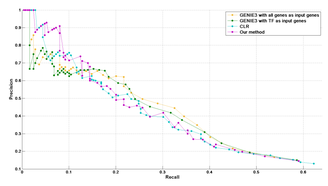
$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \qquad \text{and} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}},$$

where TP: True Positive, FP: False Positive and FN: False Negative. Results are given in terms of AUPR: Area Under the Precision-Recall curve.

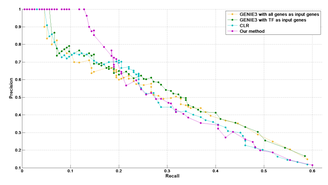
### *In silico* data: multifactorial DREAM4 Challenge



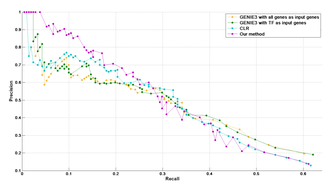

| | Network 1 | | | |
|---|---|---|---|---|
| Method | GENIE3[1] | GENIE3[2] | CLR | Our method |
| AUPR | 0.246 | 0.239 | 0.249 | **0.256** |



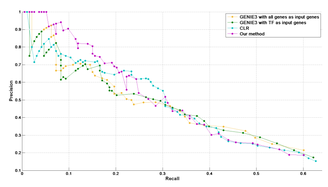

| | Network 2 | | | |
|---|---|---|---|---|
| Method | GENIE3[1] | GENIE3[2] | CLR | Our method |
| AUPR | 0.258 | **0.260** | 0.258 | **0.261** |




| | Network 3 | | | |
|---|---|---|---|---|
| Method | GENIE3[1] | GENIE3[2] | CLR | Our method |
| AUPR | 0.300 | **0.316** | 0.294 | **0.317** |



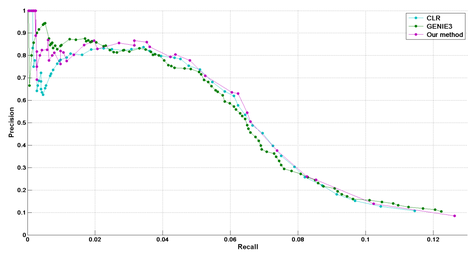

| | Network 4 | | | |
|---|---|---|---|---|
| Method | GENIE3[1] | GENIE3[2] | CLR | Our method |
| AUPR | 0.292 | 0.301 | 0.296 | **0.317** |




| | Network 5 | | | |
|---|---|---|---|---|
| Method | GENIE3[1] | GENIE3[2] | CLR | Our method |
| AUPR | 0.294 | 0.295 | 0.299 | **0.316** |

GENIE3[1]: all genes used as input genes GENIE3[2]: TF genes used as input genes

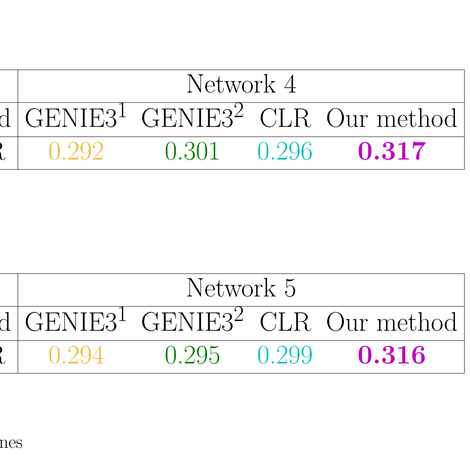
### *Real* data: *Escherichia coli* compendium




| Method | GENIE3 | CLR | Our method |
|---|---|---|---|
| AUPR ($\times 10^{-2}$) | 6.28 | 6.11 | **6.45** |
| AUPR Gain (%) | 2.2 | 5.6 | |

| Method | GENIE3 | CLR | Our method |
|---|---|---|---|
| Total. comp. time (min) | 420 | 30 | 30.05 |
| Comp. time Gain | 14 × faster | none | |

| **Precision (%)** | **Recall (%)** | | |
|---|---|---|---|
| | GENIE3 | CLR | Our method |
| 83.8 | 2.24 | 3.43 | **3.61** |
| 80 | 3.70 | 3.95 | **4.37** |
| 78 | 3.89 | 4.52 | **4.80** |
| 63.6 | 5.62 | 5.83 | **6.23** |
| **Precision (%)** | **TP edges** | | |
| | GENIE3 | CLR | Our method |
| 83.8 | 74 | 113 | **119** |
| 80 | 122 | 130 | **145** |
| 78 | 125 | 149 | **158** |
| 63.6 | 185 | 192 | **205** |



## Conclusion

- Our formulation taking into account structural *a priori* and the fast optimization *via* Graph cuts allow us to outperform state-of-the-art methods
- Existing GRN methods may benefit from our approach, as it takes a weighted graph as an input

## References

[1] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner. Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, 5:8, 2007.

[2] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring regulatory networks from expression data using tree-based methods. *Plos ONE*, 5(9):e12776, sept 2010.

[3] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. Patt. Anal. Mach. Int.*, 26:65–81, 2004.

[4] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proc. Nat. Acad. Sci. U.S.A.*, 107(14):6286–6291, Apr 2010.