

# TIGRESS: Trustful Inference of Gene Regulation using Stability Selection

Anne-Claire Haury, Fantine Mordelet, Paola Vera-Licona and  
Jean-Philippe Vert

*Centre for Computational Biology*  
Mines ParisTech, INSERM U900, Institut Curie

November 19th, 2012



# Outline

- 1 Introduction
- 2 Methods
  - Regression-based inference
  - TIGRESS
  - Material
- 3 Results
  - In silico network results
  - In vitro networks results
  - Undirected case: DREAM4
- 4 Conclusions and discussion

# Outline

## 1 Introduction

## 2 Methods

- Regression-based inference
- TIGRESS
- Material

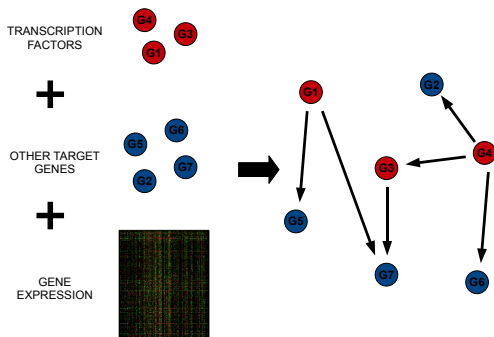
## 3 Results

- In silico network results
- In vitro networks results
- Undirected case: DREAM4

## 4 Conclusions and discussion

# DREAM network inference challenge

- **DREAM**: Dialogue on Reverse Engineering Assessments and Methods.
- **Network inference challenge**: infer *in silico* and *in vivo networks*, given list of TFs (transcription factors) and gene expression data



## DREAM challenge, continued

- **The challenge:** teams are asked to predict the 100,000 most probable interactions, along with confidence scores.

TF 12	→	TG 17	1
TF 23	→	TG 5	0.99
TF 2	→	TG 1	0.97
...	...	...	...

- **Ground truth:** blinded and revealed at the end.
- **Evaluation:** score based on AUROC and AUPR over all networks.
- **2010 results (DREAM5):**

Method	Network 1		Network 3		Network 4		Overall
	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	
GENIE3 <sup>1</sup>	0.291	0.815	0.093	0.617	0.021	0.518	40.28
ANOVerece <sup>2</sup>	0.245	0.780	0.119	0.671	0.022	0.519	34.02
<b>Naive TIGRESS</b>	<b>0.301</b>	<b>0.782</b>	<b>0.069</b>	<b>0.595</b>	<b>0.020</b>	<b>0.517</b>	<b>31.1</b>

<sup>1</sup> Huynh-Thu et al., 2010

<sup>2</sup> Kueffner et al., 2012

# This work

Three main purposes:

- **introduce** TIGRESS: Trustful Inference of Gene REgulation using Stability Selection;
- **assess** the impact of the parameters, provide guidelines as to how to choose them;
- **test** and **benchmark** TIGRESS on further datasets.

Availability:

- Paper to appear in BMC Systems Biology
- Code available: <http://cbio.ensmp.fr/tigress>

# Outline

## 1 Introduction

## 2 **Methods**

- Regression-based inference
- TIGRESS
- Material

## 3 Results

- In silico network results
- In vitro networks results
- Undirected case: DREAM4

## 4 Conclusions and discussion

# Outline

- 1 Introduction
- 2 **Methods**
  - **Regression-based inference**
  - TIGRESS
  - Material
- 3 Results
  - In silico network results
  - In vitro networks results
  - Undirected case: DREAM4
- 4 Conclusions and discussion



# Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, score all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1					
TF 2					
...					
TF $n_{tf}$					

- 2 Rank the scores altogether:

TF 12	→	TG 17	1
TF 23	→	TG 5	0.99
TF 2	→	TG 1	0.97
...		...	...

- 3 Threshold to a value or a given number  $N$  of edges.

## Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, score all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1					
TF 2					
...					
TF $n_{tf}$					

- 2 Rank the scores altogether:

TF 12  $\rightarrow$  TG 17 1  
 TF 23  $\rightarrow$  TG 5 0.99  
 TF 2  $\rightarrow$  TG 1 0.97  
 ... ..

- 3 Threshold to a value or a given number  $N$  of edges.

# Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, score all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1	-				
TF 2	<b>0.97</b>				
...	...				
TF $n_{tf}$	<b>0</b>				

- 2 Rank the scores altogether:

TF 12  $\rightarrow$  TG 17 1  
 TF 23  $\rightarrow$  TG 5 0.99  
 TF 2  $\rightarrow$  TG 1 0.97  
 ... ..

- 3 Threshold to a value or a given number  $N$  of edges.

## Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, **score** all  $n_{tf}$  candidate interactions:

	TG 1	<b>TG 2</b>	TG 3	...	TG $n_{tg}$
TF 1	-	<b>0.23</b>			
TF 2	0.97	-			
...	...	...			
TF $n_{tf}$	0	<b>0</b>			

- 2 Rank the scores altogether:

TF 12  $\rightarrow$  TG 17 1  
 TF 23  $\rightarrow$  TG 5 0.99  
 TF 2  $\rightarrow$  TG 1 0.97  
 ... ..

- 3 Threshold to a value or a given number  $N$  of edges.

## Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, **score** all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	<b>TG 3</b>	...	TG $n_{tg}$
TF 1	-	0.23	<b>0</b>		
TF 2	0.97	-	<b>0.03</b>		
...	...	...	...		
TF $n_{tf}$	<b>0</b>	<b>0</b>	<b>0</b>		

- 2 Rank the scores altogether:

TF 12  $\rightarrow$  TG 17    1  
 TF 23  $\rightarrow$  TG 5    0.99  
 TF 2  $\rightarrow$  TG 1    0.97  
 ...    ...    ...    ...

- 3 Threshold to a value or a given number  $N$  of edges.

# Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, **score** all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	<b>TG</b> $n_{tg}$
TF 1	-	0.23	0	...	<b>0.11</b>
TF 2	0.97	-	0.03	...	<b>0</b>
...	...	...	...	...	...
TF $n_{tf}$	0	0	0	...	<b>0.76</b>

- 2 Rank the scores altogether:

TF 12  $\rightarrow$  TG 17 1  
 TF 23  $\rightarrow$  TG 5 0.99  
 TF 2  $\rightarrow$  TG 1 0.97  
 ... ..

- 3 Threshold to a value or a given number  $N$  of edges.

# Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, score all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1	-	0.23	0	...	0.11
TF 2	0.97	-	0.03	...	0
...	...	...	...	...	...
TF $n_{tf}$	0	0	0	...	0.76

- 2 Rank the scores altogether:

TF 12  $\rightarrow$  TG 17 1  
 TF 23  $\rightarrow$  TG 5 0.99  
 TF 2  $\rightarrow$  TG 1 0.97  
 ... ..

- 3 Threshold to a value or a given number  $N$  of edges.

## Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, **score** all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1	-	0.23	0	...	0.11
TF 2	0.97	-	0.03	...	0
...	...	...	...	...	...
TF $n_{tf}$	0	0	0	...	0.76

- 2 **Rank** the scores altogether:

TF 12	→	TG 17	1
TF 23	→	TG 5	0.99
TF 2	→	TG 1	0.97
...		...	...

- 3 **Threshold** to a value or a given number  $N$  of edges.



# Regression-based inference: main steps

Idea: consider as many problems as TGs ( $n_{tg}$  subproblems)  
 subproblem  $g \Leftrightarrow$  find regulators  $TFs(g)$  of gene  $g$

- 1 For each TG, **score** all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1	-	0.23	0	...	0.11
TF 2	0.97	-	0.03	...	0
...	...	...	...	...	...
TF $n_{tf}$	0	0	0	...	0.76

- 2 **Rank** the scores altogether:

TF 12  $\rightarrow$  TG 17 1  
 TF 23  $\rightarrow$  TG 5 0.99  
 TF 2  $\rightarrow$  TG 1 0.97  
 ...      ...      ...      ...

- 3 **Threshold** to a value or a given number  $N$  of edges.

# GRN Inference through feature selection

## Notations

- $n_{tf}$  transcription factors (TF),  $n_{tg}$  target genes (TG)
- Expression data:  $X$  ( $n_{exp} \times n_{tg}$ ).
- $X_g$ : expression levels of gene  $g$ .
- $X_G$ : expression levels of genes in  $G$ .
- $\mathcal{T}_g$ : candidate TFs for gene  $g$ .

## Hypotheses

- 1 The expression level  $X_g$  of a TG  $g$  is a function of the expression levels  $X_{\mathcal{T}_g}$  of  $\mathcal{T}_g$ :

$$X_g = f_g(X_{\mathcal{T}_g}) + \varepsilon.$$

- 2 A score  $s_g(t)$  can be derived from  $f_g$ , for all  $t \in \mathcal{T}_g$  to assess the probability of the interaction  $(t, g)$ .

# Outline

- 1 Introduction
- 2 **Methods**
  - Regression-based inference
  - **TIGRESS**
  - Material
- 3 Results
  - In silico network results
  - In vitro networks results
  - Undirected case: DREAM4
- 4 Conclusions and discussion

# A linear model

- Base idea:

$$X_g = f_g(X_{\mathcal{T}_g}) + \varepsilon = X_{\mathcal{T}_g}\beta^g + \epsilon$$

- If  $\beta_t^g = 0$ , no edge between  $g$  and  $t$ .

## A sparse problem requires a sparsity-inducing method

- Safe to assume: few TFs regulate each TG in general. The solution is **sparse** (few edges in general):

$$X_g = X_{\mathcal{T}_g} \beta^g + \epsilon = \sum_{t \in \text{TFs}(g)} X_t \beta_t^g + \epsilon$$

- **Lasso** is one of the most common sparsity-inducing algorithms:

$$\hat{\beta}^g = \arg \min_{\beta \in \mathbb{R}^{n_{tf}}} \left\| \underbrace{X_g}_{\text{TG } g} - \underbrace{X_{\mathcal{T}_g}}_{\text{Candidate TFs (all but } g)}} \beta^g \right\|_2^2 + \lambda \|\beta^g\|_1.$$

Then,  $\hat{\beta}_t^g \neq 0 \Leftrightarrow t$  regulates  $g$ .

- Alternatively to choosing a value for  $\lambda$ , one can control the sparsity of  $\beta^g$  by a **number of LARS steps**. Roughly, after  $L$  steps in the algorithm,  $L$  TFs are chosen, which makes it **easier to compare the subproblems**.

# Stability Selection

- **Problem:** Lasso efficiency is limited:
  - ▶ when TFs are correlated, i.e. different training sets will lead to different solutions.
  - ▶ it does not provide a confidence score for each TF (no probability that the edge exists)
- **Solution:** *Meinshausen and Bühlmann, 2009* introduced Stability Selection with randomized Lasso:
  - ▶ **Resample the experiments:** run Lasso many (e.g. 1,000) times with different training sets.
  - ▶ **"Resample" the variables:** in each run, also weight the variables differently (randomized Lasso)

$$X_{it} \leftarrow W_t X_{it} \quad (1)$$

where  $W_j \sim \mathcal{U}([\alpha, 1])$  for all  $t = 1 \dots n_{tf}$ . The smaller  $\alpha$ , the more randomized the variables;  $\alpha = 1$ : no randomization.

- ▶ Get a frequency of selection for each TF.

# Stability Selection

- **Problem:** Lasso efficiency is limited:
  - ▶ when TFs are correlated, i.e. different training sets will lead to different solutions.
  - ▶ it does not provide a confidence score for each TF (no probability that the edge exists)
- **Solution:** *Meinshausen and Bühlmann, 2009* introduced Stability Selection with randomized Lasso:
  - ▶ **Resample the experiments:** run Lasso many (e.g. 1,000) times with different training sets.
  - ▶ **“Resample” the variables:** in each run, also weight the variables differently (randomized Lasso)

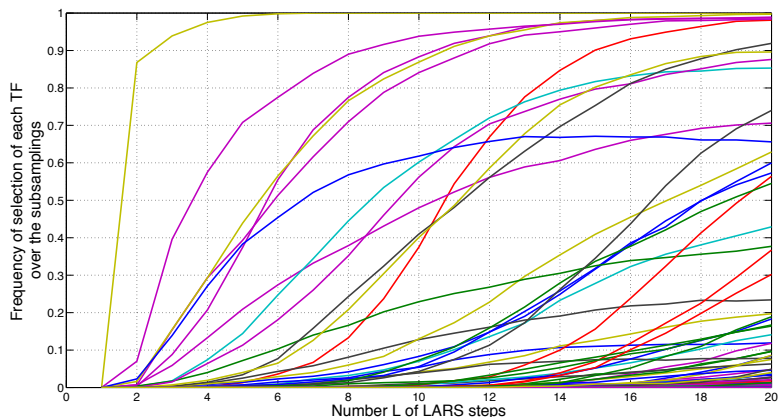
$$X_{it} \leftarrow W_t X_{it} \quad (1)$$

where  $W_j \sim \mathcal{U}([\alpha, 1])$  for all  $t = 1 \dots n_{tf}$ . **The smaller  $\alpha$ , the more randomized the variables;**  $\alpha = 1$ : no randomization.

- ▶ Get a frequency of selection for each TF.

## Stability Selection path

For each TG, Stability Selection returns such a frequency path:



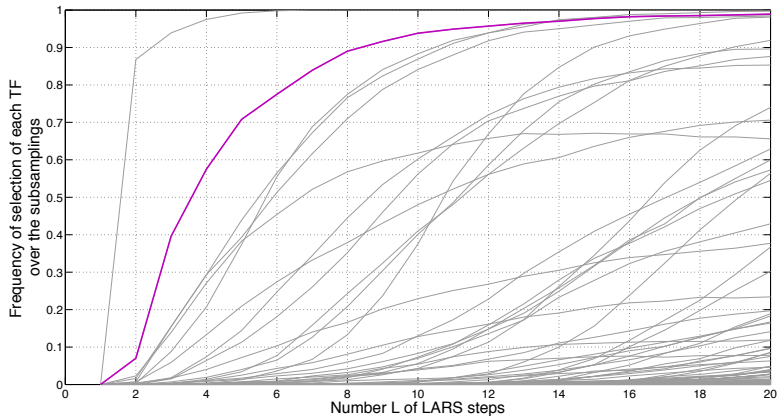
(example for one target gene)



# Scoring

How to transform this matrix into a vector of scores?

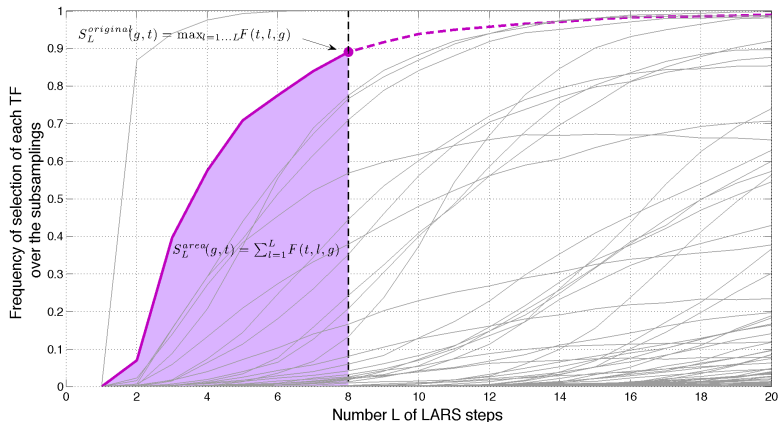
- *Original* scoring (from original paper)
- *Area* scoring (contribution)



# Scoring

How to transform this matrix into a vector of scores?

- *Original* scoring (from original paper)
- *Area* scoring (contribution)



# Get the final network

Finally,

- Rank all edges by decreasing score  $s_{L^*}$ .
- Threshold to  $N$  edges.

# TIGRESS summary

- 1 For each TG, **score** all  $n_{tf}$  candidate interactions:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1	-	0.23	0	...	0.11
TF 2	0.97	-	0.03	...	0
...	...	...	...	...	...
TF $n_{tf}$	0	0	0	...	0.76

- 2 **Rank** the scores altogether:

TF 12 → TG 17 1  
 TF 23 → TG 5 0.99  
 TF 2 → TG 1 0.97  
 ...    ...    ...    ...

- 3 **Threshold** to a value or a given number  $N$  of edges.

# TIGRESS summary

- 1 For each TG, **score** all  $n_{tf}$  candidate interactions:
  - 1 Run Stability Selection many times, get frequencies.
  - 2 Score for each value of  $L$ .
  - 3 Choose  $\hat{L}^*$ .
  - 4 Keep  $s_{\hat{L}^*}$  scores:

	TG 1	TG 2	TG 3	...	TG $n_{tg}$
TF 1	-	0.23	0	...	0.11
TF 2	0.97	-	0.03	...	0
...	...	...	...	...	...
TF $n_{tf}$	0	0	0	...	0.76

- 2 **Rank** the scores altogether:

TF 12	→	TG 17	1
TF 23	→	TG 5	0.99
TF 2	→	TG 1	0.97
...		...	...

- 3 **Threshold** to a value or a given number  $N$  of edges.

# Parameters

TIGRESS needs four parameters to be set:

- scoring method (original, area, ...)
- number of runs  $R$ : as large as computationally affordable
- randomization level  $\alpha$ : between 0 and 1
- number of LARS steps  $L$ : not obvious

# Outline

## 1 Introduction

## 2 Methods

- Regression-based inference
- TIGRESS
- **Material**

## 3 Results

- In silico network results
- In vitro networks results
- Undirected case: DREAM4

## 4 Conclusions and discussion

# Data

<b>Network</b>	<b># TF</b>	<b># Genes</b>	<b># Chips</b>	<b># Edges</b>
DREAM5 Net 1 (in-silico)	195	1643	805	4012
DREAM5 Net 3 ( <i>E. coli</i> )	334	4511	805	2066
DREAM5 Net 4 ( <i>S. cerevisiae</i> )	333	5950	536	3940
<i>E. coli</i> Net from <i>Faith et al., 2007</i>	180	1525	907	3812
DREAM4 Multifactorial Net 1	100	100	100	176
DREAM4 Multifactorial Net 2	100	100	100	249
DREAM4 Multifactorial Net 3	100	100	100	195
DREAM4 Multifactorial Net 4	100	100	100	211
DREAM4 Multifactorial Net 5	100	100	100	193



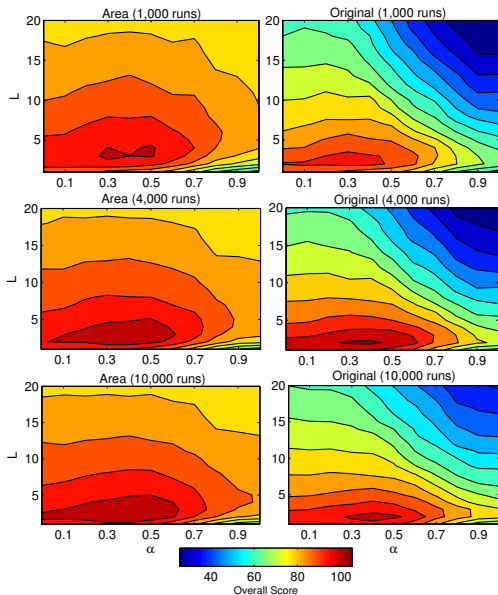
# Outline

- 1 Introduction
- 2 Methods
  - Regression-based inference
  - TIGRESS
  - Material
- 3 Results**
  - In silico network results
  - In vitro networks results
  - Undirected case: DREAM4
- 4 Conclusions and discussion

# Outline

- 1 Introduction
- 2 Methods
  - Regression-based inference
  - TIGRESS
  - Material
- 3 Results**
  - In silico network results**
  - In vitro networks results
  - Undirected case: DREAM4
- 4 Conclusions and discussion

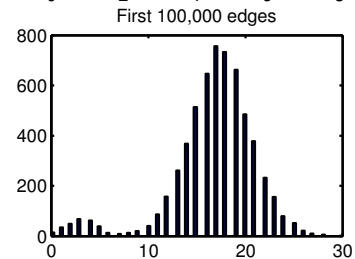
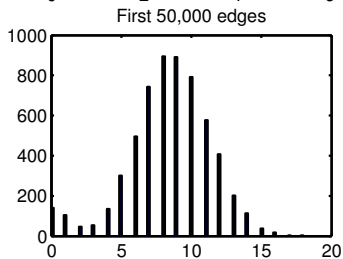
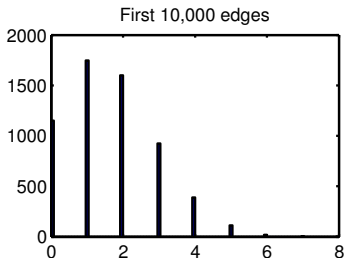
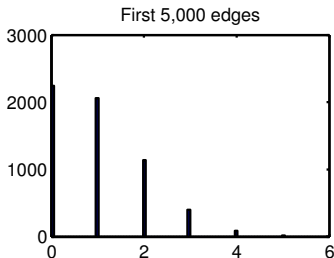
# Impact of the parameters



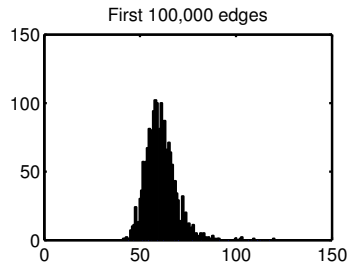
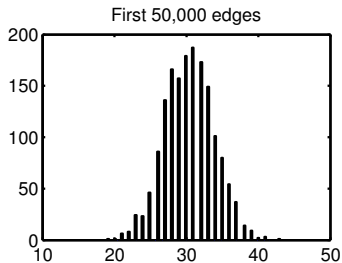
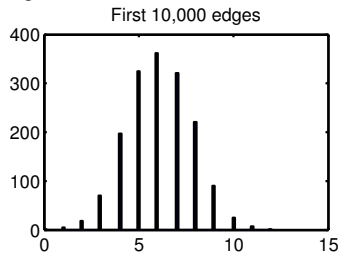
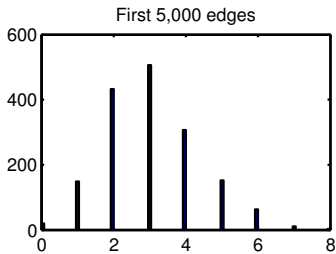
- $score = \frac{1}{2} \log_{10}(p_{AUROC} p_{AUPR})$
- Area less sensitive than *original* to  $\alpha$  and  $L$ .
- Area systematically outperforms *original*.
- *The more runs, the better*
- Best values:  $\alpha = 0.4$ ,  $L = 2$ ,  $R = 10,000$ .

# Number of TFs per TG

$$L = 2$$



# Number of TFs per TG

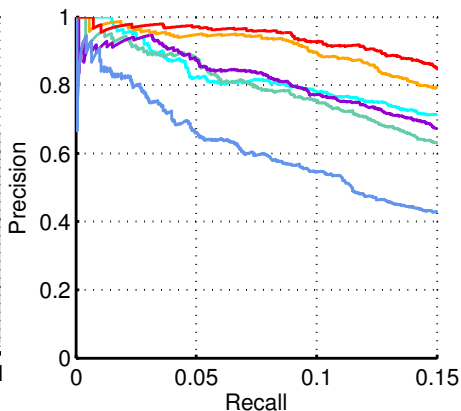
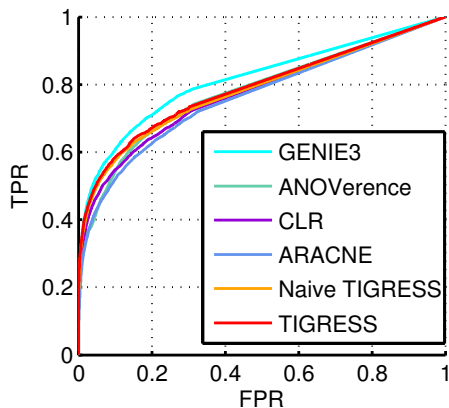
 $L = 20$ 

# Number of TFs per TG

- When  $L$  is small: more variability, more sparsity.
- When  $L$  is large: greater number of interactions per TG, less variance.

=>  $L$  should depend on the expected network's topology

# TIGRESS vs state-of-the-art

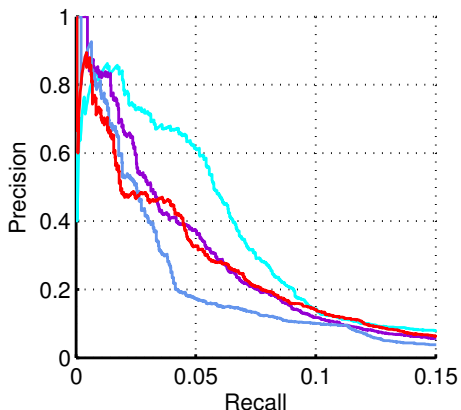
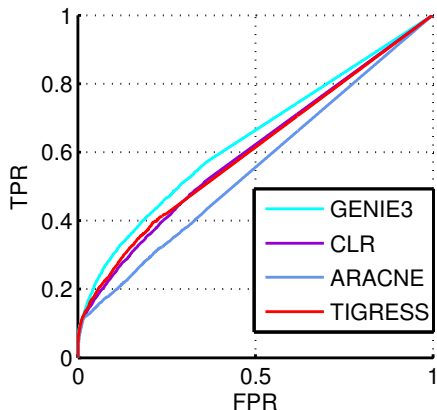


# Outline

- 1 Introduction
- 2 Methods
  - Regression-based inference
  - TIGRESS
  - Material
- 3 Results**
  - In silico network results
  - In vitro networks results**
  - Undirected case: DREAM4
- 4 Conclusions and discussion

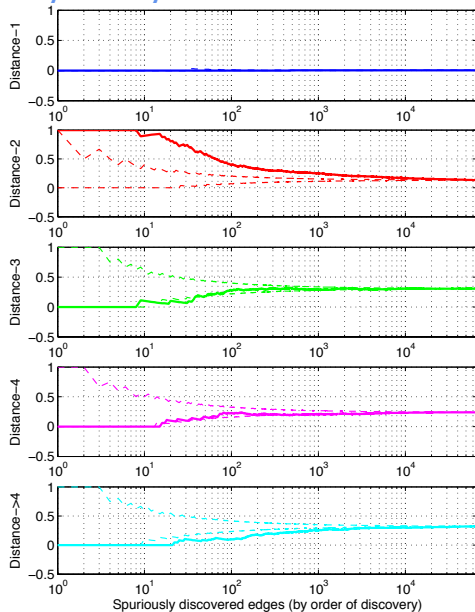


## Results on *E. coli* network

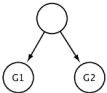
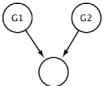



- TIGRESS is **competitive** with the best GRN inference networks on *in vitro* data.
- However: outperformed by **random forests**-based GENIE3.

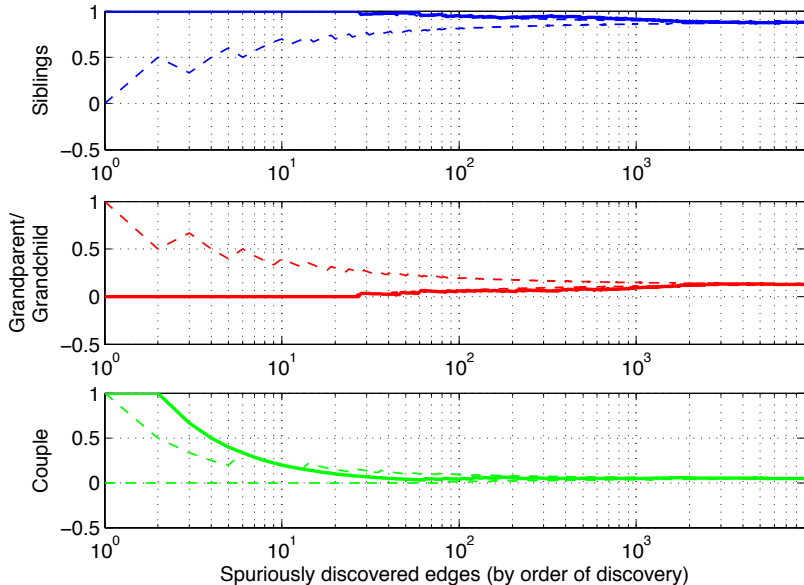
# False discovery analysis on *E. coli*



# False discovery analysis on *E. coli*

Name	Illustration	Description
Siblings		G1 and G2 have a common parent. They are <i>siblings</i> .
Couple		G1 and G2 have a common child. They are a <i>couple</i> .
Grandparent/Grandchild		G1 has a child that is a parent of G2. G1 is a <i>grandparent</i> of G2.

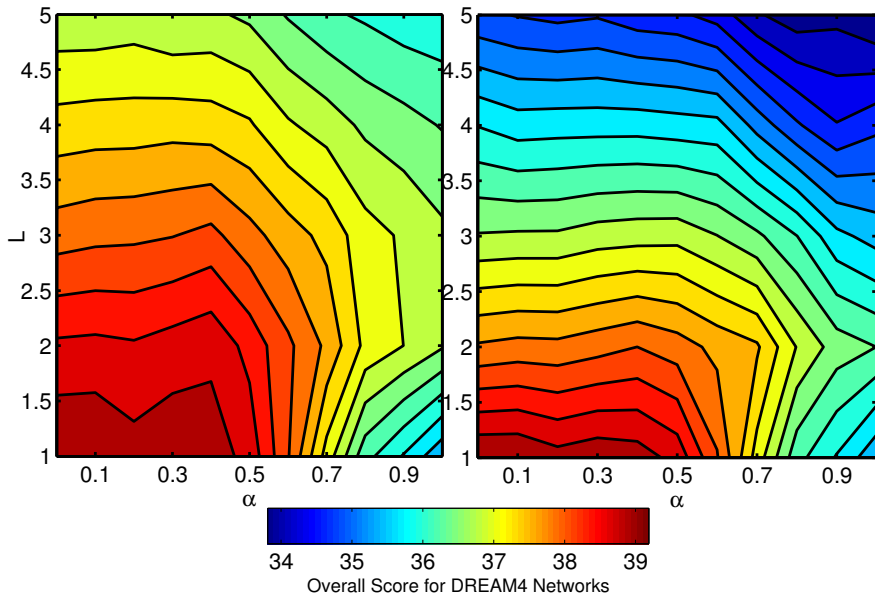
# False discovery analysis on *E. coli*



# Outline

- 1 Introduction
- 2 Methods
  - Regression-based inference
  - TIGRESS
  - Material
- 3 Results**
  - In silico network results
  - In vitro networks results
  - Undirected case: DREAM4**
- 4 Conclusions and discussion

# Undirected case: DREAM4 challenge



## Undirected case: DREAM4 challenge

*A posteriori* comparison to GENIE3 (TIGRESS run using "best" parameters from *in silico* network):

Method	Network 1		Network 2		Network 3		Network 4		Network 5	
	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC
GENIE3	0.154	0.745	0.155	0.733	0.231	0.775	0.208	0.791	0.197	0.798
TIGRESS	0.165	0.769	0.161	0.717	0.233	0.781	0.228	0.791	0.234	0.764

Overall scores:

- GENIE3: 37.48
- TIGRESS: 38.85

# Outline

- 1 Introduction
- 2 Methods
  - Regression-based inference
  - TIGRESS
  - Material
- 3 Results
  - In silico network results
  - In vitro networks results
  - Undirected case: DREAM4
- 4 Conclusions and discussion



# Conclusion

- TIGRESS provides:
  - ▶ **Automatization** and adaptation of the Stability Selection procedure to the GRN inference problem.
  - ▶ **Area scoring setting**: better results and less elasticity to parameters.
  - ▶ 3rd best performer at DREAM5, confirmed second best on both *in silico* and *E. coli* networks. "Best" performer *a posteriori* on undirected DREAM4 networks.
  - ▶ Code, demos and data available (MATLAB). Fast (SPAMS toolbox, *Mairal et al., 2009*) and parallelizable.
- **However: outperformed by GENIE3**
  - ▶ TIGRESS uses essentially the same global framework as GENIE3...
  - ▶ ... but GENIE3 is not linear (random forests).
  - ▶ Overall: confirmation that **regression-based methods** belong to the state-of-the-art.

# Discussion

## 1 How to choose the right model?

- ▶ The linear model is clearly not correct.
- ▶ It has **high bias** and **low variance**.
- ▶ It is also **easily interpretable**.
- ▶ **Simple and false vs obscure and performant?**

## 2 Perspectives

- ▶ Use chip information?
- ▶ Group situations (operons): group Lasso may be able to solve it.

# Acknowledgments



Fantine Mordelet



Paola Vera-Licona



Jean-Philippe Vert

Thank you for your attention!