

# 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

TEST - Telecom ParisTech, January 13th, 2012



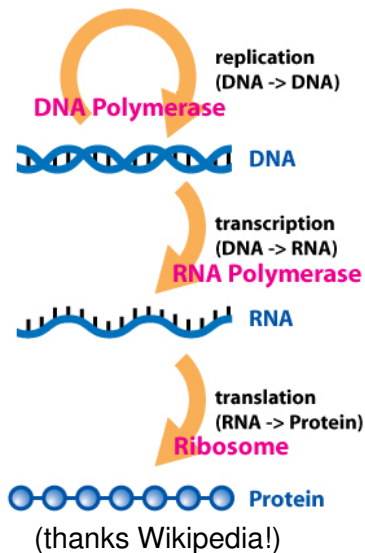
# Outline

- 1 Introduction
- 2 TIGRESS: Trustful Inference of Gene REgulation using Stability Selection
- 3 Results
  - In silico network results
  - E. coli network results
- 4 Conclusions and discussion

# Outline

- 1 Introduction
- 2 TIGRESS: Trustful Inference of Gene REgulation using Stability Selection
- 3 Results
  - In silico network results
  - E. coli network results
- 4 Conclusions and discussion

# The central dogma of molecular biology



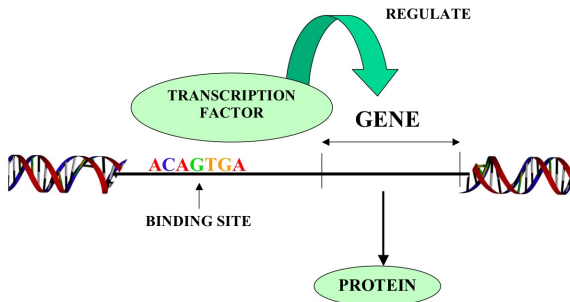
Information transfer to transform a gene into a protein:

- 1 **Transcription:** DNA is copied into messenger RNA (mRNA) that contains the same genetic information.
- 2 **Translation:** mRNA leaves the nucleus and is transformed into a protein by the ribosomes.

## Zoom on transcription

- In order for transcription to occur, one needs **transcription factors**.
- They are either promoting or inhibiting transcription of other genes.

Legend: A transcription factor molecule binds to the DNA at its binding site, and thereby regulates the production of a protein from a gene.



(Borrowed from <http://howardhughes.trinity.duke.edu/>)

# Gene Regulatory Networks

- Gene Regulatory Network (GRN):

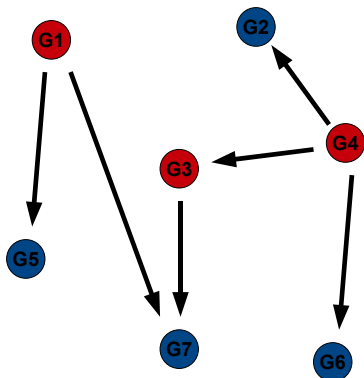
- ▶ Complex set of interactions between genes
- ▶ Transcription factors (TF) activate or repress target genes (TG).
- ▶ *Note that*  $\{TFs\} \in \{TGs\}$ .

- Example

- ▶ G4 regulates G2, G3 and G6.
- ▶ G7 is regulated both by G3 and G1.

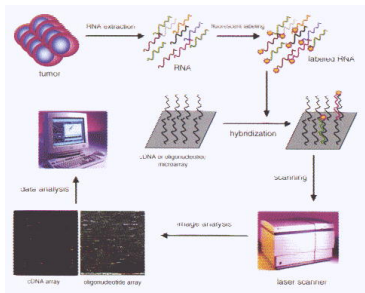
- Why reconstruct it?

- ▶ Understand the structure of regulation better (causality, patterns,...)
- ▶ Applications such as drug target identification.

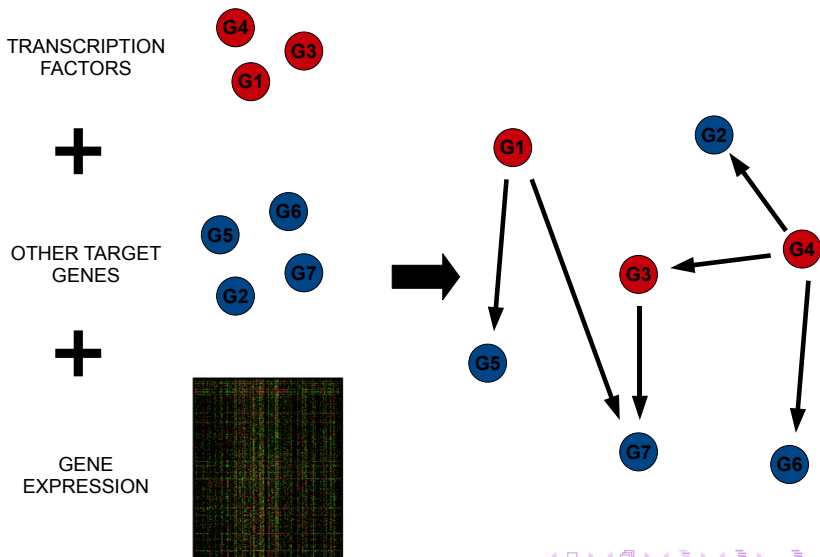


# Gene expression data and microarrays

- The more activated a gene, the more quantity of mRNA in the nucleus, the more **expressed** the gene.
- Therefore **measuring gene expression** amounts to measuring the quantity of mRNA.
- **Microarrays** are chips on which RNAs are **hybridized**. The more RNA in the cell, the more on the chip.
- They are scanned and transformed into an image. Levels of grey represent gene expression.



# Reconstruction of a GRN using gene expression data





# Regression-based inference

For 10+ years, many methods have been proposed using, e.g.:

- Static and dynamic bayesian networks,
- Boolean networks,
- Correlation-based methods,
- Information-theoretic based methods,
- ...
- Regression-based methods.
  - ▶ It is assumed that the expression level of a TG is a function of the expression levels of the TFs that regulate it.
  - ▶ We will focus here on linear regression.

# Outline

- 1 Introduction
- 2 TIGRESS: Trustful Inference of Gene REgulation using Stability Selection
- 3 Results
  - In silico network results
  - E. coli network results
- 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 → 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	<b>TG 3</b>	...	TG $n_{tg}$
TF 1	-	0.23	<b>0</b>		
TF 2	0.97	-	<b>0.03</b>		
...	...	...	...		
TF $n_{tf}$	0	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	TG 3	...	<b>TG <math>n_{tg}</math></b>
TF 1	-	0.23	0	...	<b>0.11</b>
TF 2	0.97	-	0.03	...	<b>0</b>
...	...	...	...	...	<b>...</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  $\rightarrow$  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.

# 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):

$$X_g = X_T \beta^g + \epsilon = \sum_{t \in TFs(g)} X_t \beta_t^g + \epsilon$$

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

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

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 exist)
- **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 exist)
- **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)

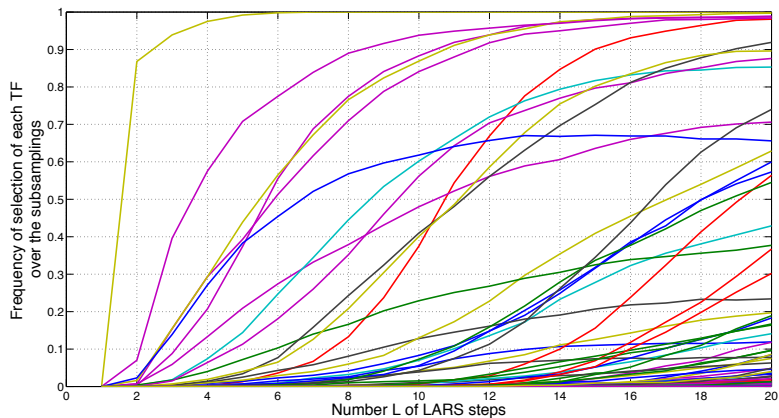
$$X_{it} \leftarrow W_t X_{it} \quad (1)$$

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:

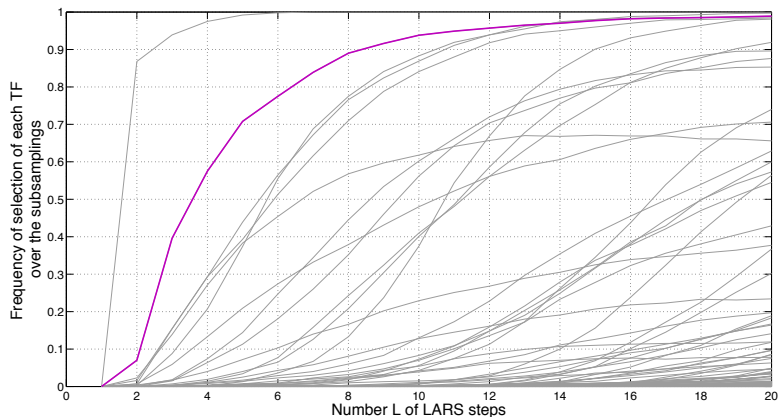


(example for one target gene)

# Scoring

How to transform this matrix into a vector of scores?

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

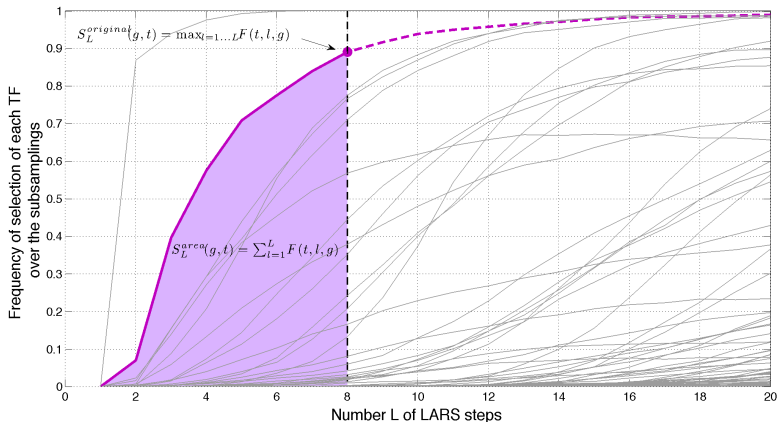




# Scoring

How to transform this matrix into a vector of scores?

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



## How to choose the right $L$ ?

Simple heuristic:

- Let  $N$  be the total number of edges one wants to predict.
- For each value of  $L$ , put all scores together into a vector  $S_L$  (of size  $n_{tf} \times n_{tg}$ ).

- Then,

$$\hat{L}^* = \arg \min_{L=1 \dots L_{\max}} |\#\{s \in S_L, s \neq 0\} - N|$$

- $\hat{L}^*$  is the value for which TIGRESS predicts the number of interactions closest to  $N$ .
- What is  $N$ ? Assume that each TG is regulated by 3 TFs in average. Then fix  $N = 3n_{tg}/prec$  where  $prec$  is the expected precision for a recall of 1.  $N$  is the necessary number of predictions for all true edges to be retrieved.

# 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

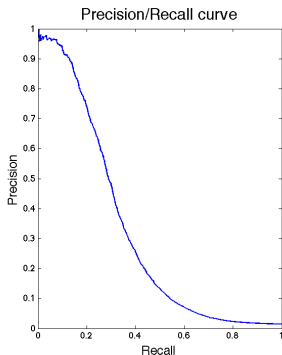
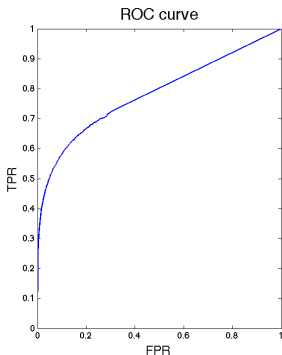
- 2 **Rank** the scores altogether:

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.

# Evaluation

- **AUROC**: Area Under the ROC curve
- **AUPR**: Area Under the Precision/Recall curve
- p-values  $p_{AUPR}$  and  $p_{AUROC}$ : probability that a given or higher AUPR (resp. AUROC) could be achieved by chance. The smaller the better.



# Data

- **DREAM 5 Challenge 4 *in silico* dataset**: 805 experiments, 1643 genes, 195 TFs
- ***E. Coli* network** from *Faith et al, 2007*: 907 experiments, 4297 genes, 3812 verified interactions among 1525 of the genes present in the microarrays experiments.

# Outline

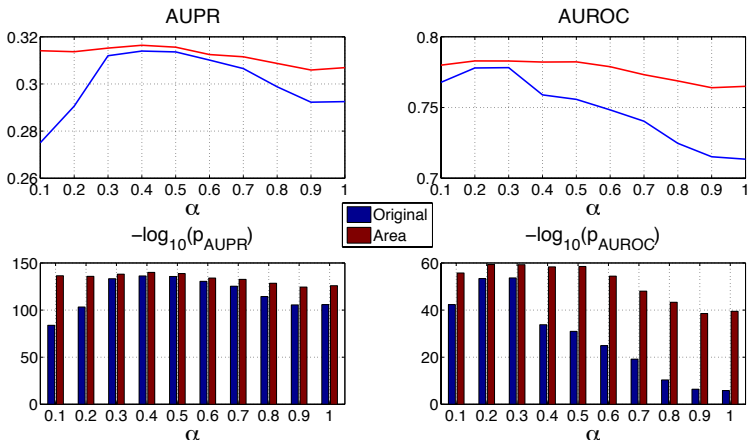
- 1 Introduction
- 2 TIGRESS: Trustful Inference of Gene REgulation using Stability Selection
- 3 **Results**
  - In silico network results
  - E. coli network results
- 4 Conclusions and discussion



# Outline

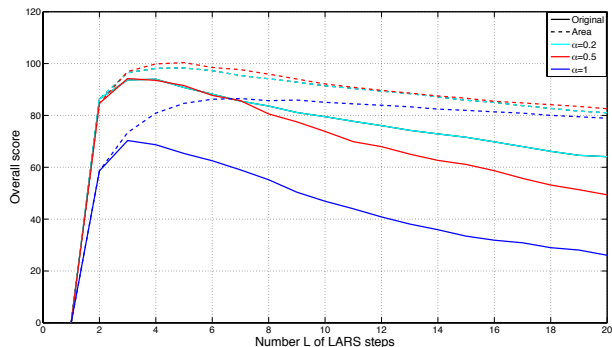
- 1 Introduction
- 2 TIGRESS: Trustful Inference of Gene REgulation using Stability Selection
- 3 Results**
  - **In silico network results**
  - E. coli network results
- 4 Conclusions and discussion

# Level of randomization



- *Area* less sensitive than *original* to level of randomization.
- *Area* systematically outperforms *original*.
- Best values for  $\alpha$ : between 0.1 and 0.5.

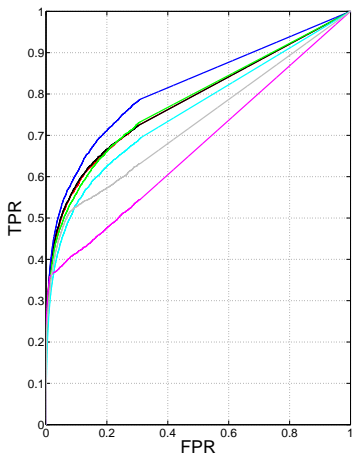
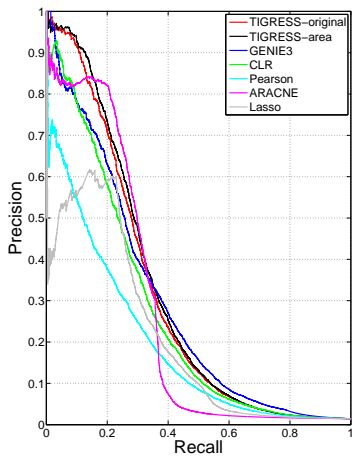
# Number of LARS steps



Area		
$\alpha$	$\hat{L}^*$	$L^*$
0.1	3	4
0.2	4	4
0.3	4	4
0.4	4	4
0.5	5	5
0.6	6	5
0.7	6	5
0.8	7	6
0.9	7	8
1	8	7

- The larger  $\alpha$ , the most critical the value of  $L$ .
- *Area* less sensitive than *original* to value of  $L$ .
- *Area* systematically outperforms *original*.
- Our estimates of  $L^*$  are close to the truth.

## TIGRESS vs ...



## TIGRESS vs ...

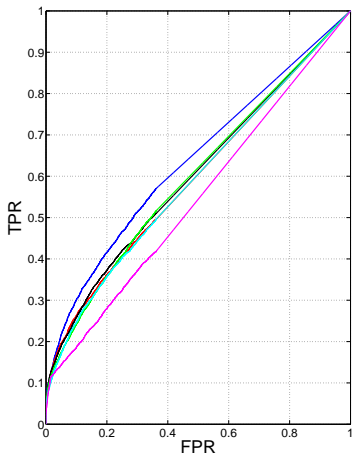
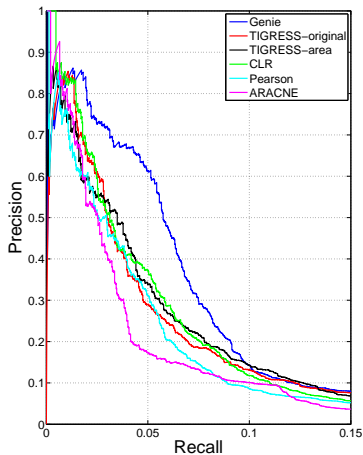
Algorithm	AUPR	$p_{AUPR}$	AUROC	$p_{AUROC}$
TIGRESS	0.3152	8.01e-139	0.7829	5.43e-60
GENIE3	0.2915	2.91e-105	0.8155	2.30e-107
CLR	0.2654	1.82e-73	0.7817	1.41e-58
Pearson	0.1887	3.71e-13	0.7568	1.44e-32
ARACNE	0.2758	1.73e-85	0.6715	9.82e-01
Lasso	0.2079	1.38e-23	0.7280	1.06e-12

**Table:** AUPR, AUROC and p-values obtained by several methods on the *in silico* dataset.

# Outline

- 1 Introduction
- 2 TIGRESS: Trustful Inference of Gene REgulation using Stability Selection
- 3 Results**
  - In silico network results
  - E. coli network results**
- 4 Conclusions and discussion

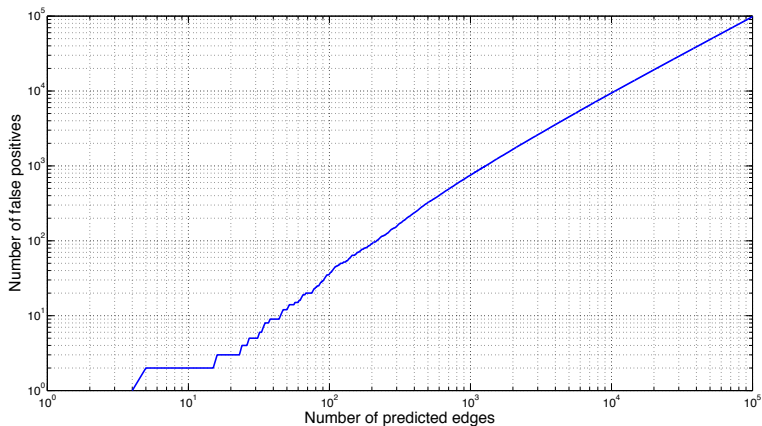
# Results on *E. coli* network



- TIGRESS is **competitive** with the best GRN inference networks on *in vivo* data.

# False discovery analysis

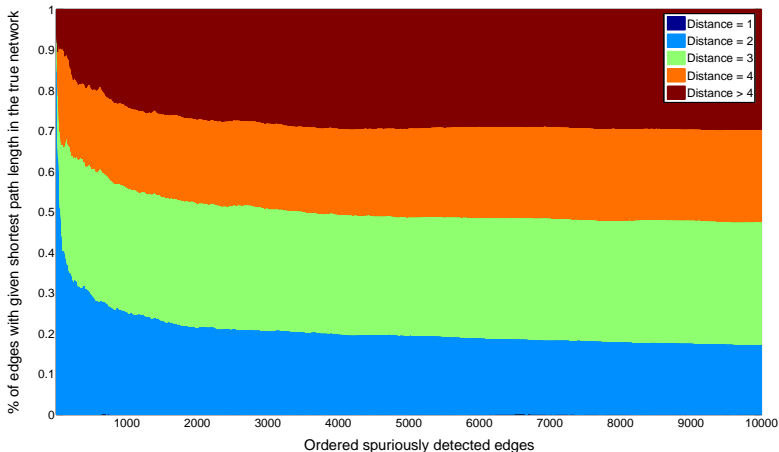
Very high proportion of **false positives** even in the top edges:



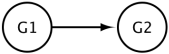
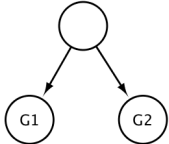
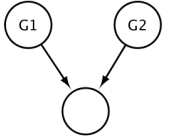
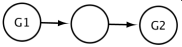


## How far the Fps from the truth?

Length of the shortest path in the true network between nodes in spuriously discovered edges:

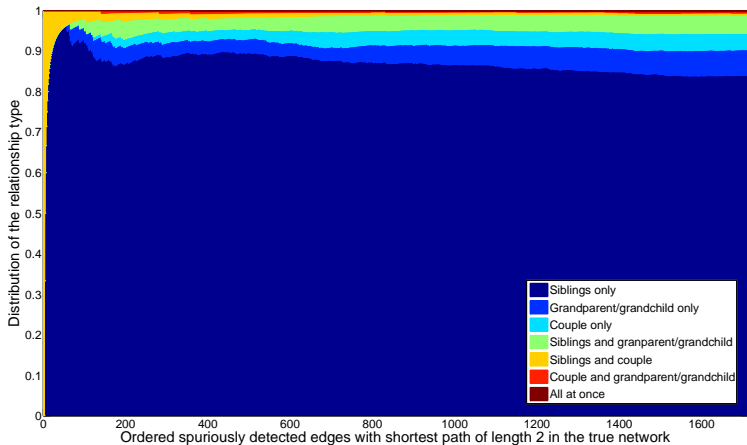


# Patterns

Distance	Name	Illustration	Description
1	Parent/ Child		G1 is a <b>parent</b> of G2.
2	Siblings		G1 and G2 hav a common parent. They are <b>siblings</b> .
	Couple		G1 and G2 have a common child. They are a <b>couple</b> .
	Grandparent/ Grandchild		G1 has a child that is a parent of G2. G1 is a <b>grandparent</b> of G2.

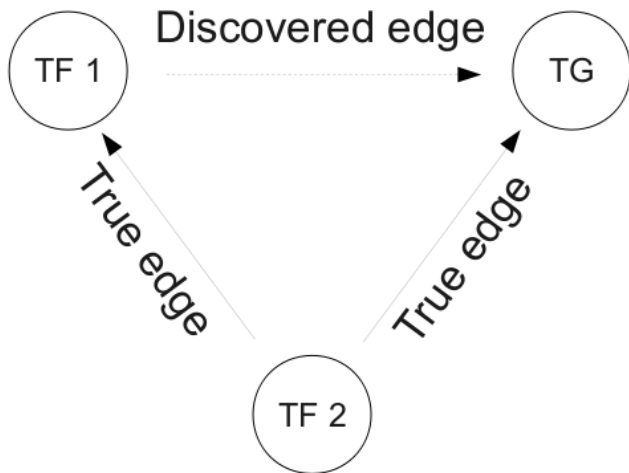
## Who are distance 2 FPs?

Type of patterns for distance 2 FPs:



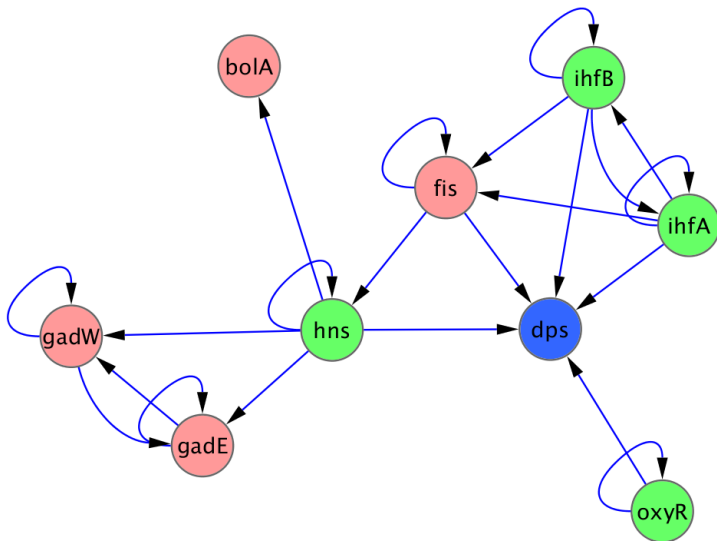
## The special case of siblings

We look for parent/child relationships. Instead, we find many **siblings**:



## The special case of siblings

We look for parent/child relationships. Instead, we find many **siblings**:



# Outline

- 1 Introduction
- 2 TIGRESS: Trustful Inference of Gene REgulation using Stability Selection
- 3 Results
  - In silico network results
  - E. coli network results
- 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.
  - ▶ Nice results (3rd best performer at DREAM5, confirmed second best on both *in silico* and *E. coli* 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 (but not that bad: FPs are not far apart in the true graph)
- ▶ It has **high bias** and **low variance**.
- ▶ It is also **easily interpretable**.
- ▶ Best method (GENIE3) does not achieve great scores in general: there is something more to the problem than a wrong model choice.

## 2 Could further information be used?

- ▶ We assumed in this work that expression data contains all the necessary information. Probably not true.
- ▶ Some groups of genes are **regulated by the same TFs** (e.g. operons): prior information?
- ▶ The experiments are not i.i.d (replicates, different situations...).
- ▶ Adding **priors on motifs** (e.g. feed-forward loops) is an option for future work.



# Acknowledgments



Fantine Mordelet



Paola Vera-Licona



Jean-Philippe Vert

Thank you for your attention!