New Local Move Operators for Learning the Structure of Bayesian Networks

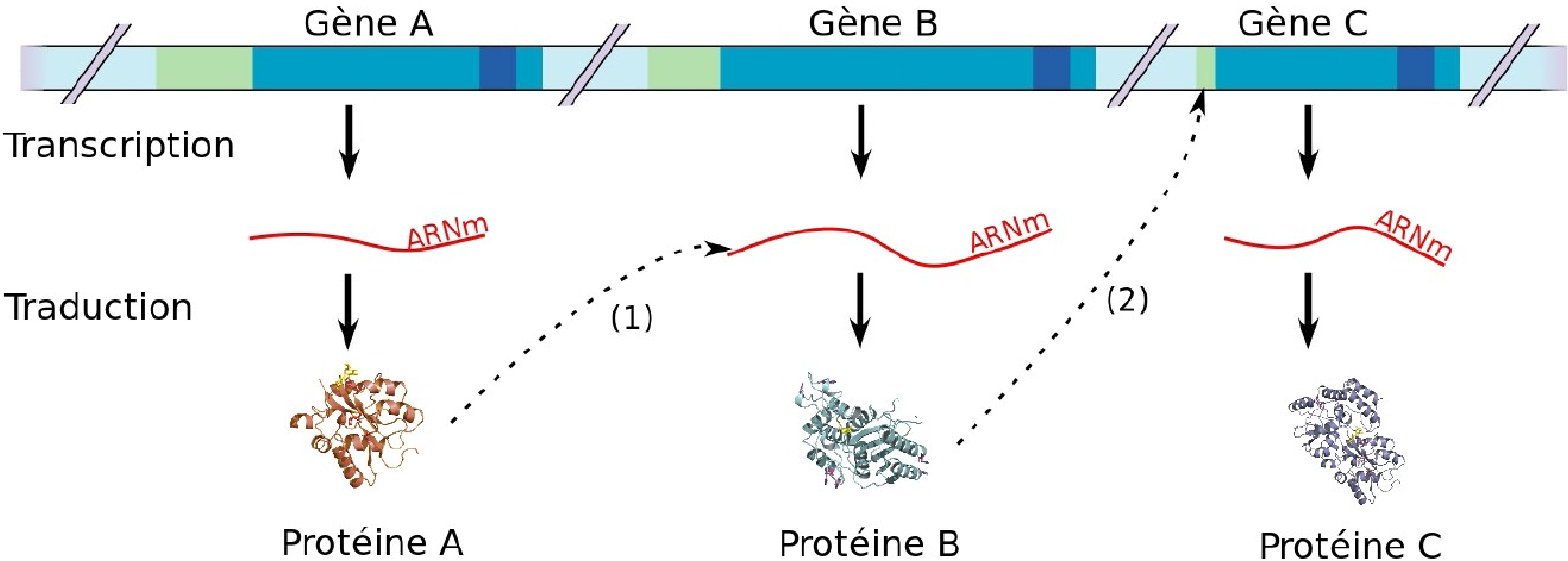
Jimmy Vandel, Brigitte Mangin & Simon de Givry

UR875 UBIA, INRA Toulouse

Plan

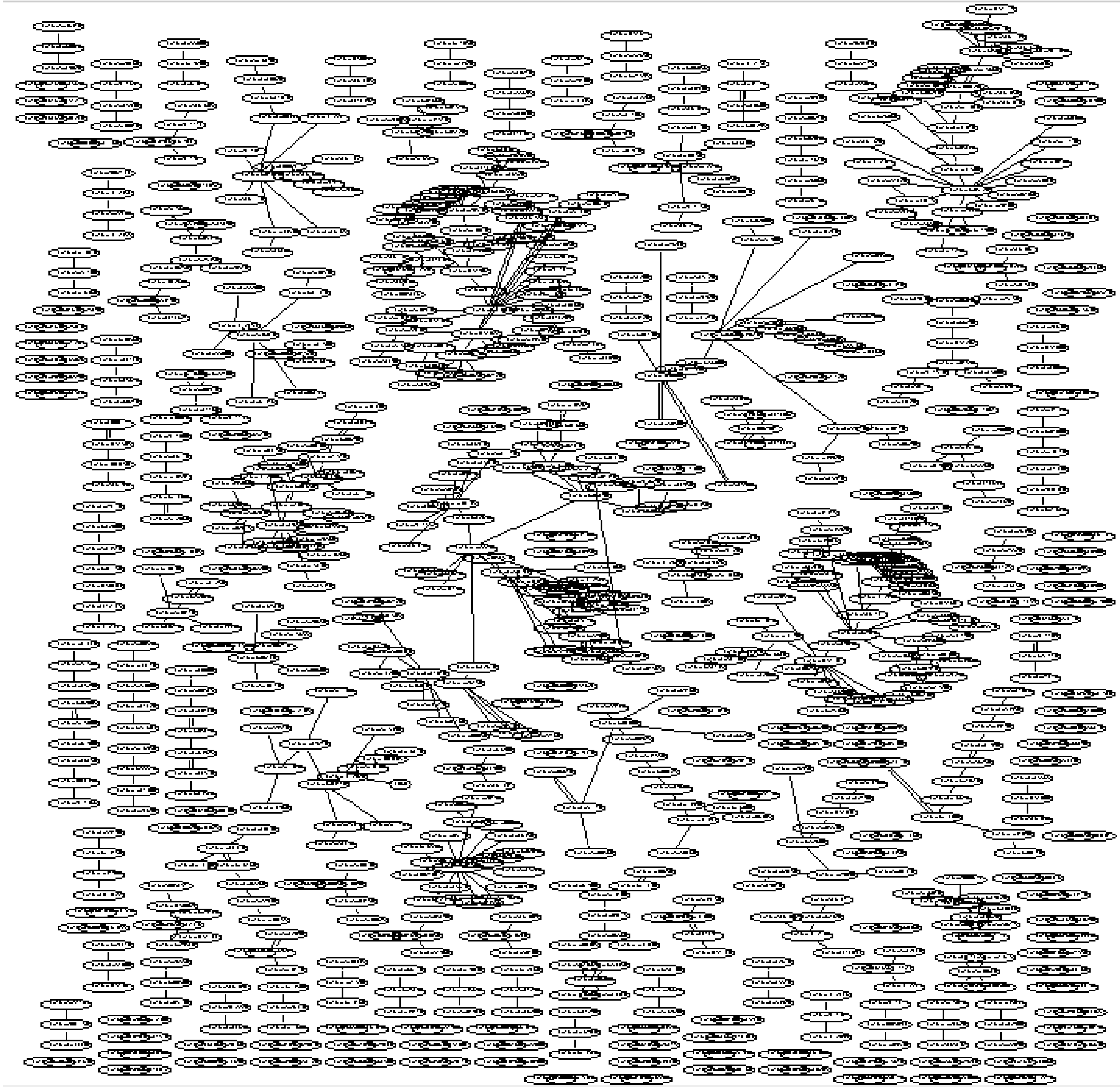
- Context
- Bayesian Network
- Stochastic Local Search (Greedy Search)
- New local move operators (*swap* and *swap**)
- Experiments on standard BN benchmarks
- Experiments in genetical genomics (DREAM 2012)
- Conclusion & perspectives

Different levels of regulation



Gene network of *trans regulations* for 2775 transcripts with high *eQTL* (LOD≥3) measured on 158 RIL Arabidopsis thaliana.

Bootstrap threshold = 0.3

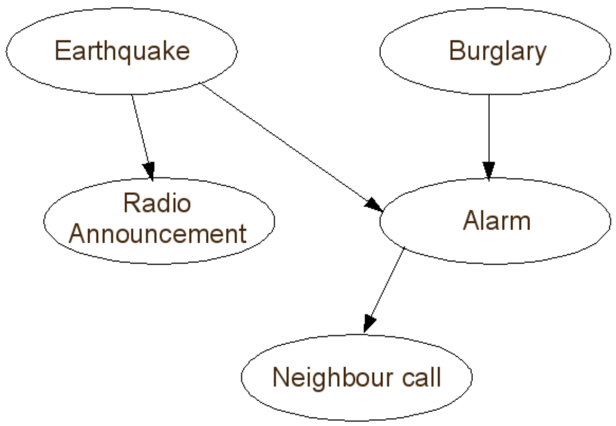


Discrete Bayesian network

Static Bayesian networks *(Friedman et al., Plos comput. bio., 2000)*

- Directed Acyclic Graph (DAG)
- Conditional probability distribution of X_i , given its parents Pa_i in G: $P_G(X_i / Pa_i^j) = \theta_i^j$

  (independence of these local probabilities)



Probability Distribution for the Alarm Node given the events of "Earthquakes" and "Burglaries"			
E	B	P(A | E,B)	P(!A | E,B)
E	B	0.90	0.10
E	! B	0.20	0.80
! E	B	0.90	0.10
! E	! B	0.01	0.99

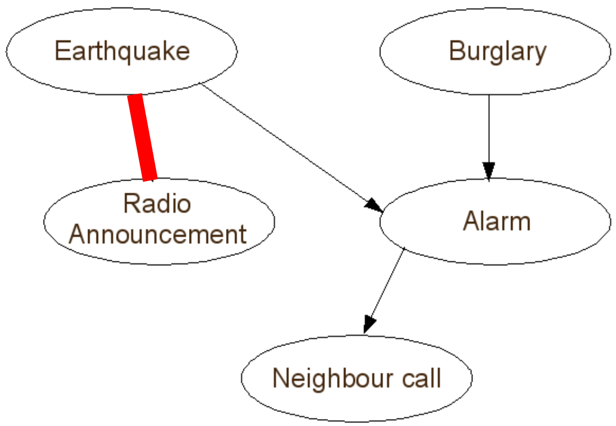
Graphic representation of a joint probability distribution:

$$P_G(X) = \prod_{i=1}^{n} P_G(X_i / Pa_i)$$

Discrete Bayesian network

$\{ Earthquake \perp Burglary , \ldots \}$

Partial DAG (PDAG)



- Directed Acyclic Graph (DAG)
- Conditional probability distribution of X_i , given its parents Pa_i in G: $P_G(X_i / Pa_i^j) = \theta_i^j$

  (independence of these local probabilities)

Graphic representation of a joint probability distribution:

$$P_G(X) = \prod_{i=1}^{n} P_G(X_i / Pa_i)$$

Probability Distribution for the Alarm Node given the events of "Earthquakes" and "Burglaries"			
E	B	P(A | E,B)	P(!A | E,B)
E	B	0.90	0.10
E	! B	0.20	0.80
! E	B	0.90	0.10
! E	! B	0.01	0.99

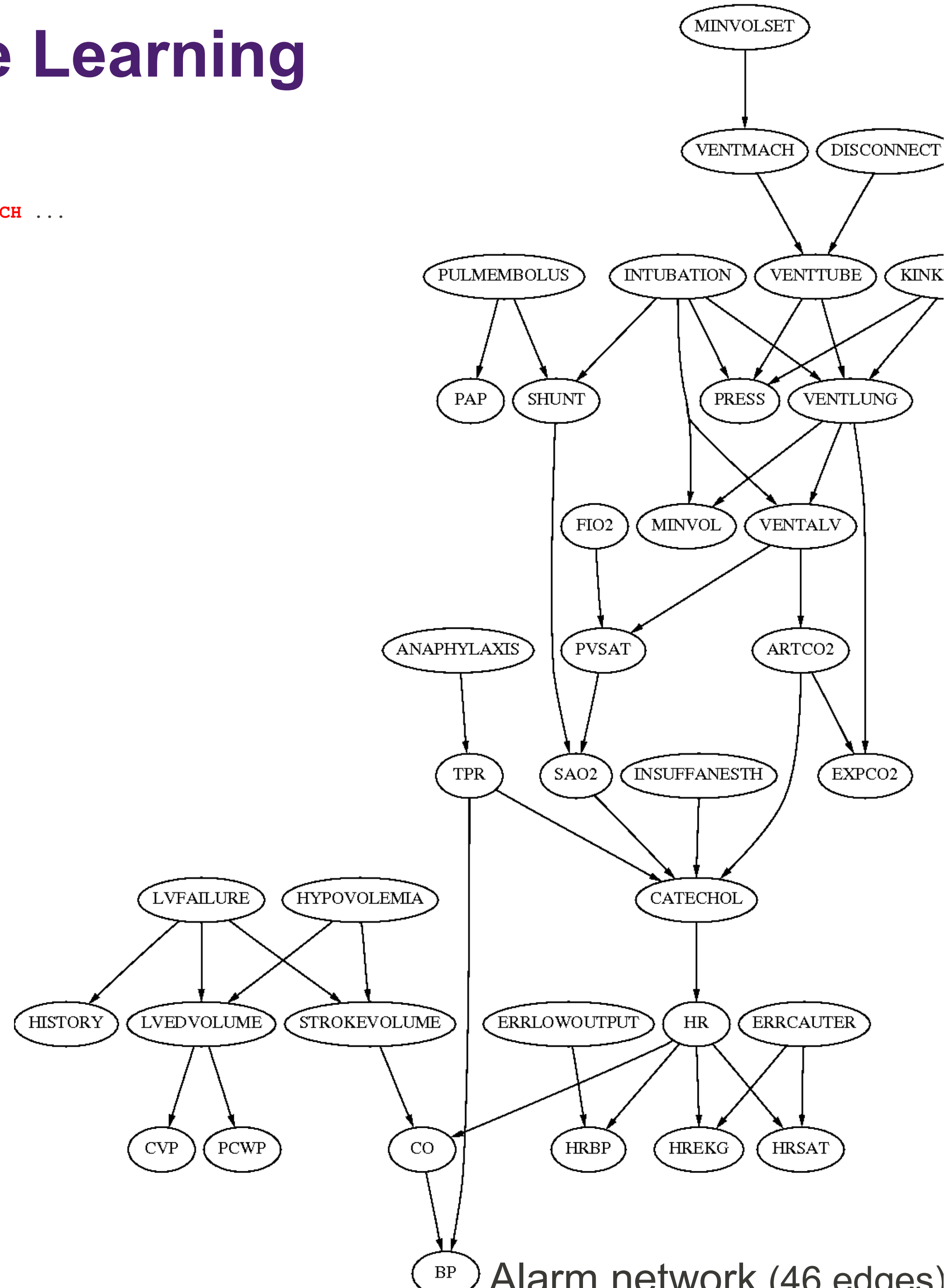
Structure Learning

SAO2	FIO2	PRESS	EXPCO2	MINVOL	MIVOLS	HYPOV	LVFAI	ANAPH	INSUF	VENTMACH	...
NORMAL	LOW	HIGH	ZERO	HIGH	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
NORMAL	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	HIGH	ZERO	NORMAL	FALSE	FALSE	FALSE	TRUE	NORMAL	
LOW	LOW	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
NORMAL	NORMAL	HIGH	ZERO	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	HIGH	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	TRUE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
HIGH	NORMAL	HIGH	LOW	HIGH	LOW	FALSE	FALSE	FALSE	FALSE	LOW	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
HIGH	NORMAL	LOW	LOW	HIGH	HIGH	FALSE	FALSE	FALSE	FALSE	HIGH	
LOW	NORMAL	NORMAL	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	HIGH	HIGH	HIGH	FALSE	FALSE	FALSE	FALSE	HIGH	
HIGH	NORMAL	HIGH	LOW	HIGH	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	HIGH	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	ZERO	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	LOW	NORMAL	LOW	FALSE	FALSE	FALSE	FALSE	LOW	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	NORMAL	LOW	NORMAL	FALSE	TRUE	FALSE	TRUE	NORMAL	
LOW	NORMAL	HIGH	NORMAL	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	LOW	NORMAL	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
NORMAL	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	ZERO	LOW	LOW	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
HIGH	NORMAL	HIGH	LOW	HIGH	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	LOW	NORMAL	FALSE	FALSE	TRUE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	HIGH	FALSE	FALSE	FALSE	FALSE	HIGH	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
...											

Alarm dataset (37 variables)

Structure Learning

SAO2	FIO2	PRESS	EXPCO2	MINVOL	MIVOLS	HYPOV	LVFAI	ANAPH	INSUF	VENTMACH	...
NORMAL	LOW	HIGH	ZERO	HIGH	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
NORMAL	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	HIGH	ZERO	NORMAL	FALSE	FALSE	FALSE	TRUE	NORMAL	
LOW	LOW	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
NORMAL	NORMAL	HIGH	ZERO	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	HIGH	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	TRUE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
HIGH	NORMAL	HIGH	LOW	HIGH	LOW	FALSE	FALSE	FALSE	FALSE	LOW	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
HIGH	NORMAL	LOW	LOW	HIGH	HIGH	FALSE	FALSE	FALSE	FALSE	HIGH	
LOW	NORMAL	NORMAL	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	HIGH	HIGH	HIGH	FALSE	FALSE	FALSE	FALSE	HIGH	
HIGH	NORMAL	HIGH	LOW	HIGH	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	HIGH	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	ZERO	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	LOW	NORMAL	LOW	FALSE	FALSE	FALSE	FALSE	LOW	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	NORMAL	LOW	NORMAL	FALSE	TRUE	FALSE	TRUE	NORMAL	
LOW	NORMAL	HIGH	NORMAL	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	LOW	NORMAL	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
NORMAL	NORMAL	LOW	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	ZERO	LOW	LOW	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
HIGH	NORMAL	HIGH	LOW	HIGH	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	LOW	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	LOW	ZERO	LOW	NORMAL	FALSE	FALSE	TRUE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	FALSE	FALSE	FALSE	FALSE	NORMAL	
LOW	NORMAL	HIGH	ZERO	ZERO	HIGH	FALSE	FALSE	FALSE	FALSE	HIGH	
LOW	NORMAL	NORMAL	ZERO	ZERO	NORMAL	TRUE	FALSE	FALSE	FALSE	NORMAL	
...											



Alarm network (46 edges)

Score based learning

- We look for the graph maximizing an objective function
  - easy to evaluate, avoids over-fitting and Markov-equivalent
    - decomposable, penalized and equivalent scores
      - BDe score *(**D.Heckerman** Machine learning 1995)*
      - *BIC score (**G.Schwartz** Annals of statistics 1978)*

- Local score change from G to G' after operation OP_i modifying Pa_i

$$\Delta_{score} OP_i = f(G') - f(G) = f_{X_i}(G') - f_{X_i}(G)$$

(assuming G' is a DAG)

Local search components

1. Search space

   - Directed Acyclic Graph
   - Partial DAG (PDAG)

   - variable orders

Local search components

1. Search space
   - Directed Acyclic Graph
   - Partial DAG (PDAG)

   - variable orders

2. Initial structure
   - empty structure
   - random structure
   - informed structure (MWST, expert...)

Local search components

1. Search space
   - Directed Acyclic Graph
   - Partial DAG (PDAG)

   - variable orders

2. Initial structure
   - empty structure
   - random structure
   - informed structure (MWST, expert...)

3. Neighborhood operators
   - addition of an edge
   - deletion of an edge
   - reversal of an edge
   - k look-ahead
   - optimal reinsertion

Local search components

1. Search space
   - Directed Acyclic Graph
   - Partial DAG (PDAG)

   - variable orders

2. Initial structure
   - empty structure
   - random structure
   - informed structure (MWST, expert...)

3. Neighborhood operators
   - addition of an edge
   - deletion of an edge
   - reversal of an edge
   - k look-ahead
   - optimal reinsertion

4. Meta-heuristics
   - greedy search (**GS**) (with restarts)
   - tabu search
   - simulated annealing
   - MCMC
   - genetic algorithms
   - ...

Local search components

1. Search space
   - Directed Acyclic Graph
   - Partial DAG (PDAG)
   - variable orders

2. Initial structure
   - empty structure
   - random structure
   - informed structure (MWST, expert...)

3. Neighborhood operators
   - addition of an edge
   - deletion of an edge
   - reversal of an edge
   - k look-ahead
   - optimal reinsertion

---

4. Meta-heuristics
   - greedy search (**GS**)

PDAG	empty structure	addition & deletion	**GES**
		(*Greedy Equivalence Search*, Chickering 2002)	
DAG	empty structure	restricted 2 look-ahead	**LAGD**
		(*k Look-Ahead in l Good Directions*, Holland 2008)	

Local search components

1. Search space
   - Directed Acyclic Graph
   - Partial DAG (PDAG)
   - **Directed Cyclic Graph**

   - variable orders

2. Initial structure
   - empty structure
   - random structure
   - informed structure (MWST, expert...)

3. Neighborhood operators
   - addition of an edge *
   - deletion of an edge
   - reversal of an edge *
   - k look-ahead
   - optimal reinsertion
   - **swap of an edge ***
   - **Iterative operators ***

---

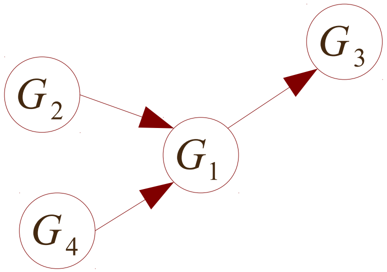
4. Meta-heuristics
   - greedy search (**GS**)
   - **Stochastic Greedy Search**

PDAG empty structure addition & deletion **GES**
(*Greedy Equivalence Search*, Chickering 2002)

DAG empty structure restricted 2 look-ahead **LAGD**
(*k Look-Ahead in l Good Directions*, Holland 2008)

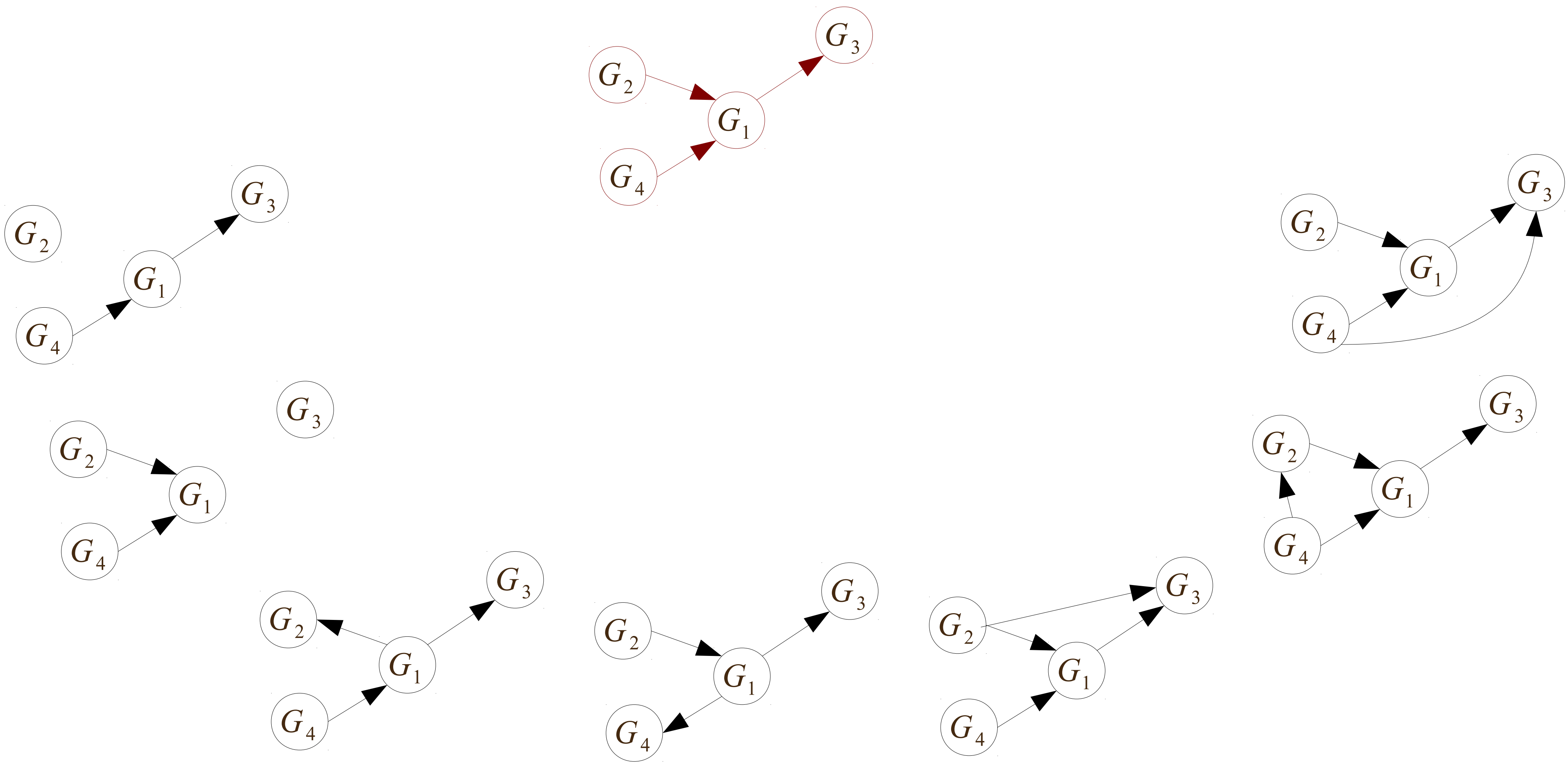
GS Algorithm

- **Greedy search**
  - *Start with an initial network (empty graph, a priori graph)*
  - *Score all possible local modifications (addition / deletion / reversal of one edge) and select the best of them (if it exist)*



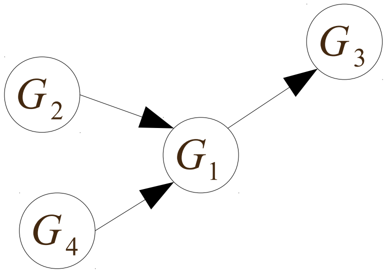
GS Algorithm

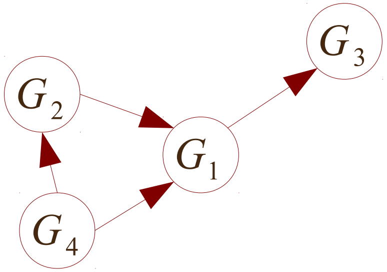
- **Greedy search**
  - *Start with an initial network (empty graph, a priori graph)*
  - *Score all possible local modifications (addition / deletion / reversal of one edge) and select the best of them (if it exist)*



GS Algorithm

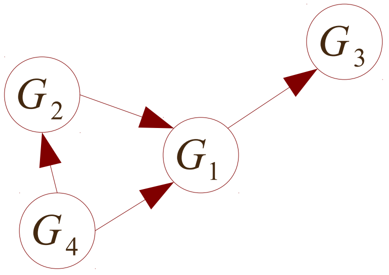
- **Greedy search**
  - *Start with an initial network (empty graph, a priori graph)*
  - *Score all possible local modifications (addition / deletion / reversal of one edge) and select the best of them (if it exist)*





GS Algorithm

- **Greedy search**
  - *Start with an initial network (empty graph, a priori graph)*
  - *Score all possible local modifications (addition / deletion / reversal of one edge) and select the best of them (if it exist)*



SGS Algorithm

- **Sochastic Greedy Search (SGS)**

classical Greedy Search (GS)
+
random edge orientation for Marvov-equivalent structures

- Markov-equivalent structures in Bayesian networks

G_1 G_2 G_3

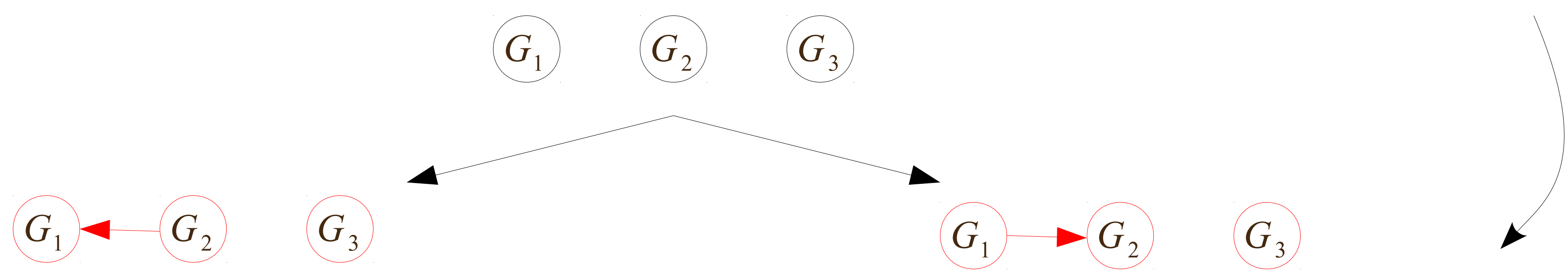
SGS Algorithm

- **Sochastic Greedy Search (SGS)**

classical Greedy Search (GS)
+
random edge orientation for Marvov-equivalent structures

- Markov-equivalent structures in Bayesian networks

$G_1 \not\perp G_2$

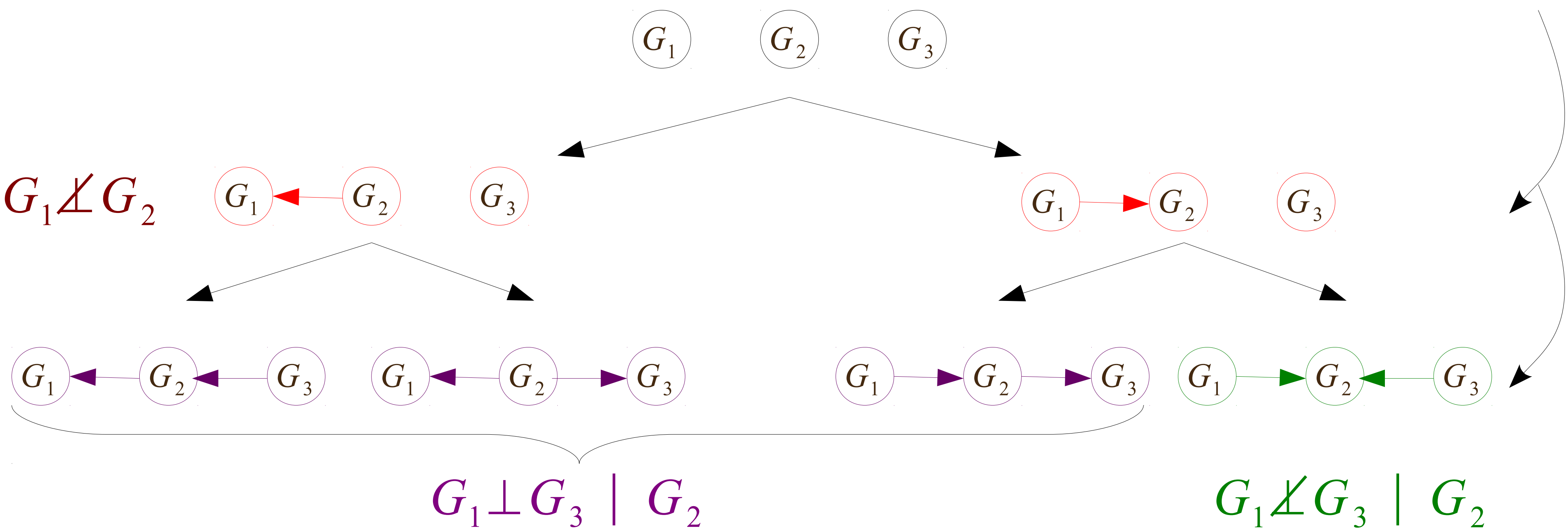


SGS Algorithm

- **Sochastic Greedy Search (SGS)**

classical Greedy Search (GS)
+
random edge orientation for Marvov-equivalent structures

- Markov-equivalent structures in Bayesian networks

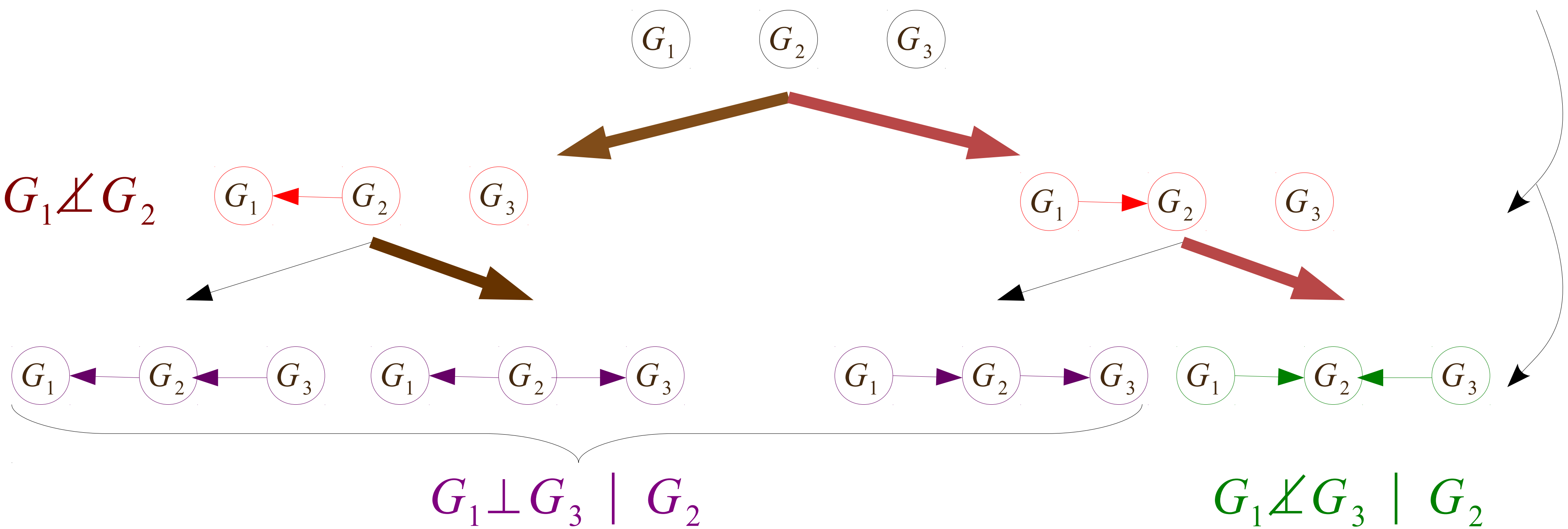


SGS Algorithm

- **Sochastic Greedy Search (SGS)**

classical Greedy Search (GS)
+
random edge orientation for Marvov-equivalent structures

- Markov-equivalent structures in Bayesian networks



Swap Operator

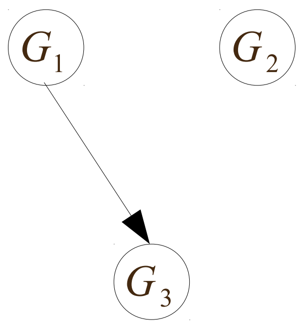
- addition
- deletion
- reversal (deletion + addition on the same pair)
- **swap (deletion + addition including an extra node)**

Swap Operator

- addition
- deletion
- reversal (deletion + addition on the same pair)
- **swap (deletion + addition including an extra node)**

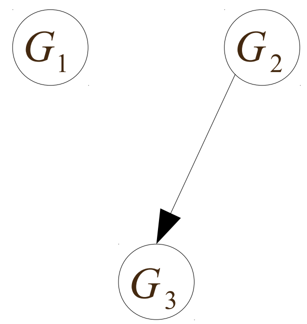
  <u>Example</u>:

Current situation



$$\Delta_{score} Add(G_{2,} G_3) > \Delta_{score} Add(G_{1,} G_3) > 0$$

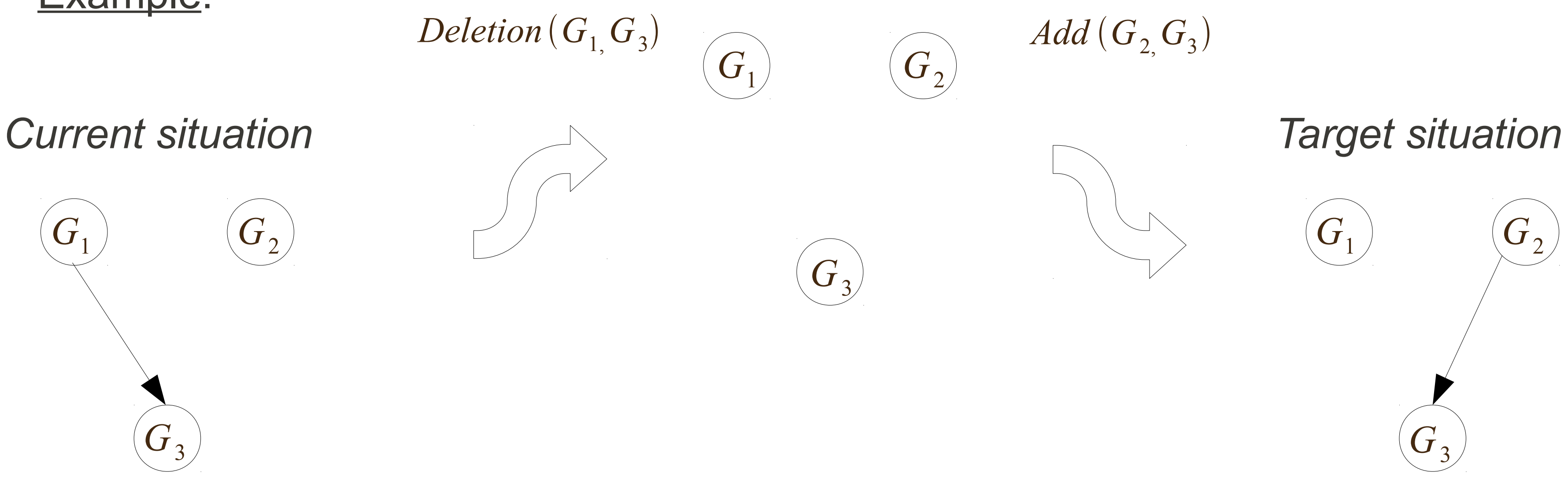
Target situation



Swap Operator

- addition
- deletion
- reversal (deletion + addition on the same pair)
- **swap (deletion + addition including an extra node)**

<u>Example</u>:

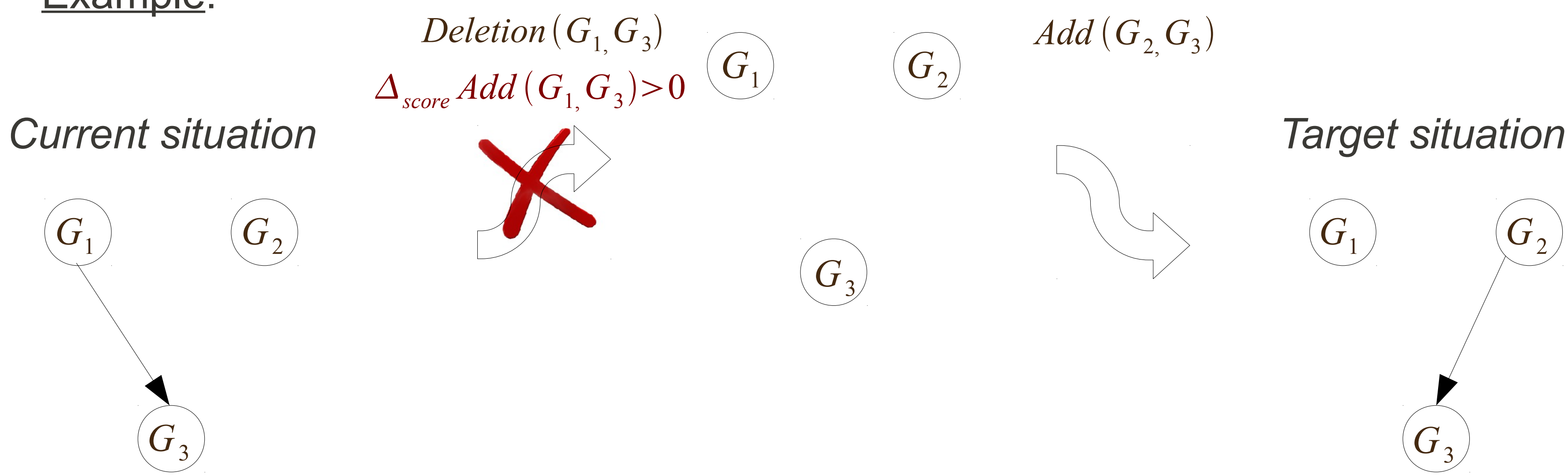


$$\Delta_{score} Add(G_2, G_3) > \Delta_{score} Add(G_1, G_3) > 0$$

Swap Operator

- addition
- deletion
- reversal (deletion + addition on the same pair)
- **swap (deletion + addition including an extra node)**

Example:

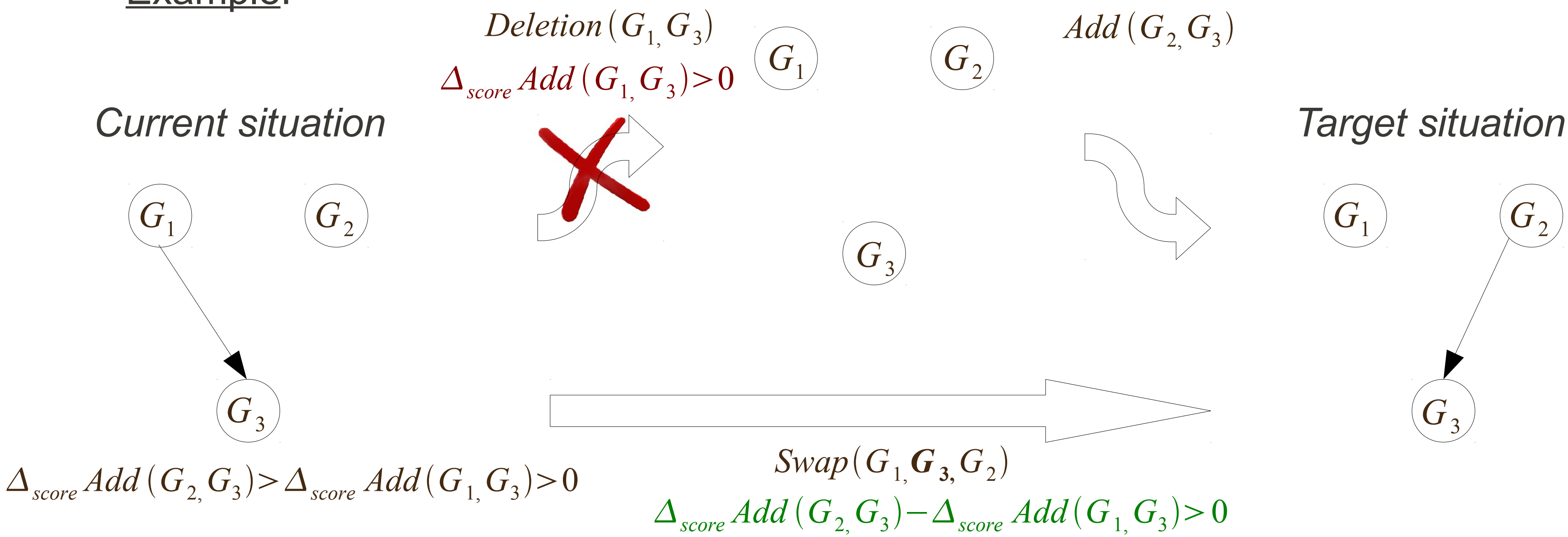


$\Delta_{score} Add(G_{2,} G_3) > \Delta_{score} Add(G_{1,} G_3) > 0$

Swap Operator

- addition
- deletion
- reversal (deletion + addition on the same pair)
- **swap (deletion + addition including an extra node)**

<u>Example</u>:



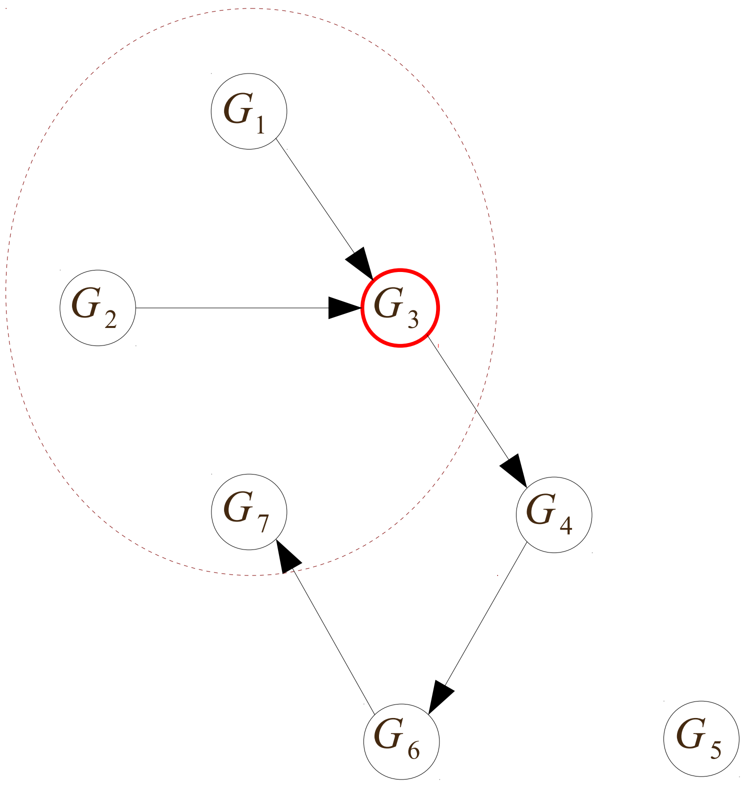
→ escape from some local maxima

Swap★ Operator

$Swap(G_{2,} \boldsymbol{G_{3,}} G_7)?$

Current situation

$\Delta_{score} Add(G_{7,} G_3|G_1) > \Delta_{score} Add(G_{2,} G_3|G_1) > 0$



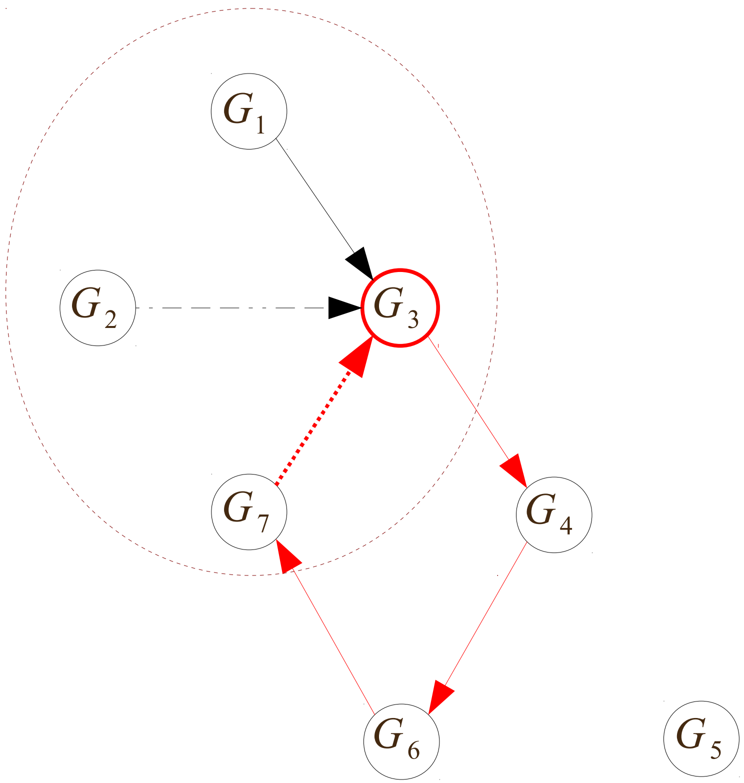
Swap★ Operator

$$Swap(G_{2,} \mathbf{G_{3,}} G_7)? \longrightarrow Cycle\{G_{3,} G_{4,} G_{6,} G_7\}$$

Current situation

$$\Delta_{score} Add(G_{7,} G_3|G_1) > \Delta_{score} Add(G_{2,} G_3|G_1) > 0$$

Objective : delete the cycles

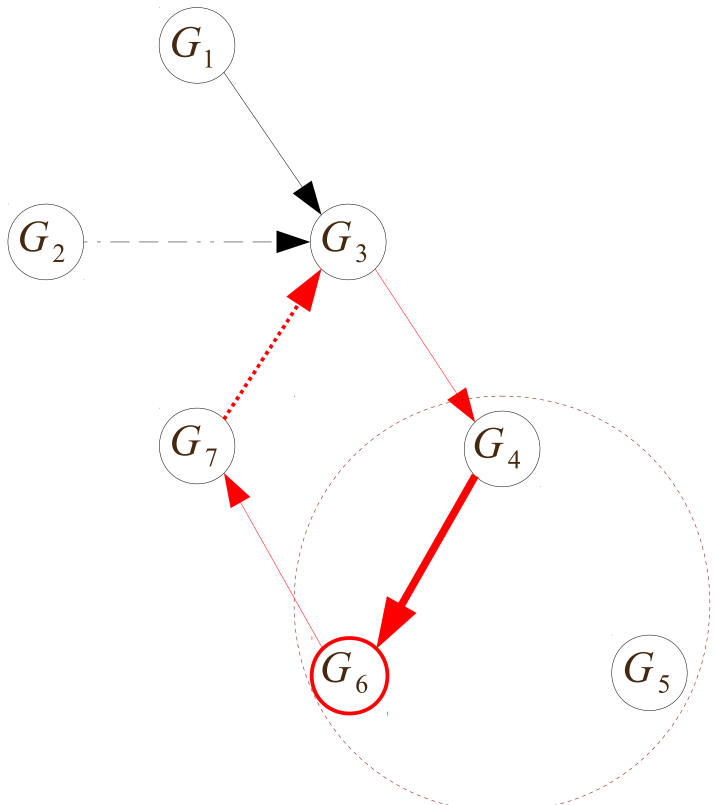


Swap* Operator

$$Swap(G_2, \boldsymbol{G_3}, G_7)? \longrightarrow Cycle\{G_3, G_4, G_6, G_7\}$$

Current situation

$$\Delta_{score} Add(G_7, G_3|G_1) > \Delta_{score} Add(G_2, G_3|G_1) > 0$$



Objective : delete the cycles

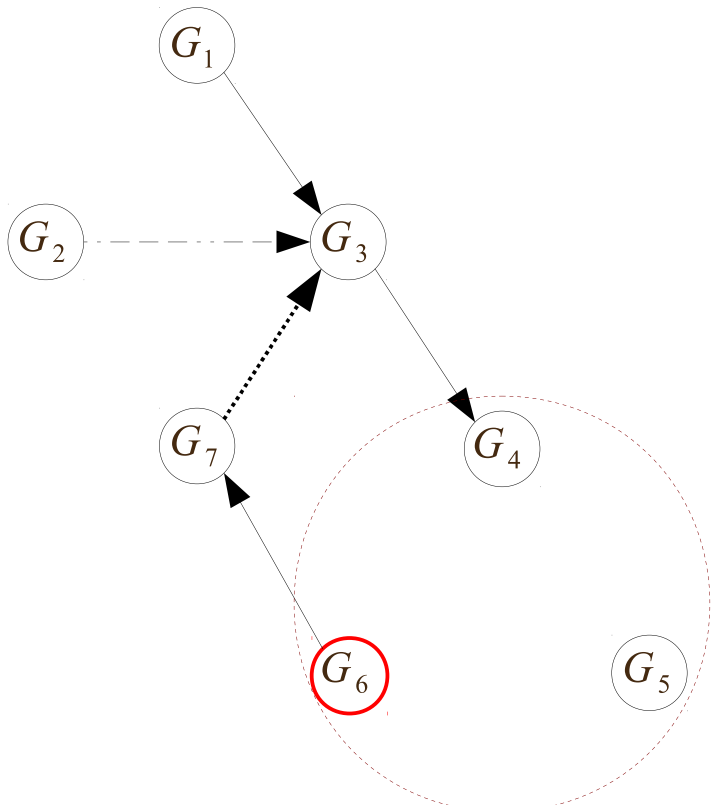
Step 1. Try to delete the edge minimizing $\Delta_{score} Add$

Swap* Operator

$Swap(G_2, \mathbf{G_3}, G_7)? \longrightarrow Cycle\{G_3, G_4, G_6, G_7\}$

Current situation

$\Delta_{score} Add(G_7, G_3 | G_1) > \Delta_{score} Add(G_2, G_3 | G_1) > 0$



Objective : delete the cycles

Step 1. Try to delete the edge minimizing $\Delta_{score} Add$

Improving score ?

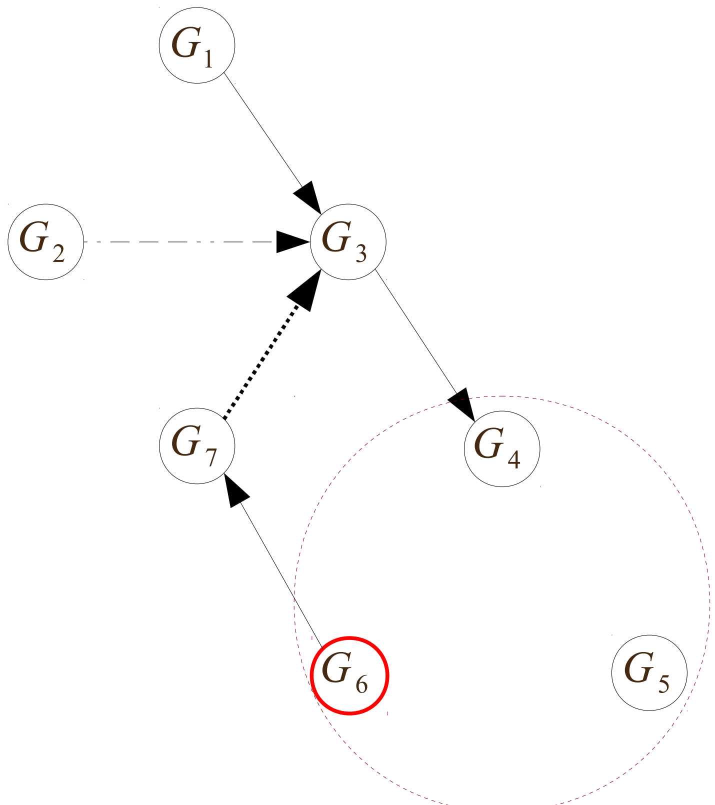
Yes

Swap★ Operator

$$Swap(G_2, \mathbf{G_3}, G_7)? \longrightarrow Cycle\{G_3, G_4, G_6, G_7\}$$

Current situation

$$\Delta_{score} Add(G_7, G_3|G_1) > \Delta_{score} Add(G_2, G_3|G_1) > 0$$



Objective : delete the cycles

Step 1. Try to delete the edge minimizing $\Delta_{score} Add$

Improving score ?

Yes → **Acyclic ?**

Yes → OK !

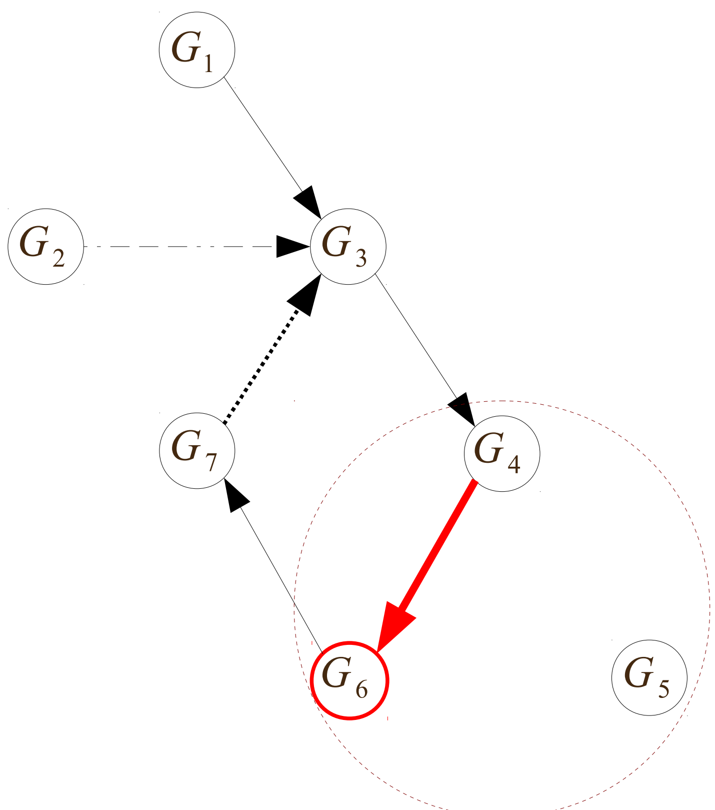
No → Return to Step 1

Swap* Operator

$Swap(G_2, \mathbf{G_3}, G_7)?$ $\longrightarrow$ $Cycle\{G_3, G_4, G_6, G_7\}$

Current situation

$\Delta_{score} Add(G_7, G_3 | G_1) > \Delta_{score} Add(G_2, G_3 | G_1) > 0$



Objective : delete the cycles

Step 1. Try to delete the edge minimizing $\Delta_{score} Add$

Improving score ?

Yes $\rightarrow$ Acyclic ?

Yes $\rightarrow$ OK !

No $\rightarrow$ Return to Step 1

No $\rightarrow$ Continue to Step 2

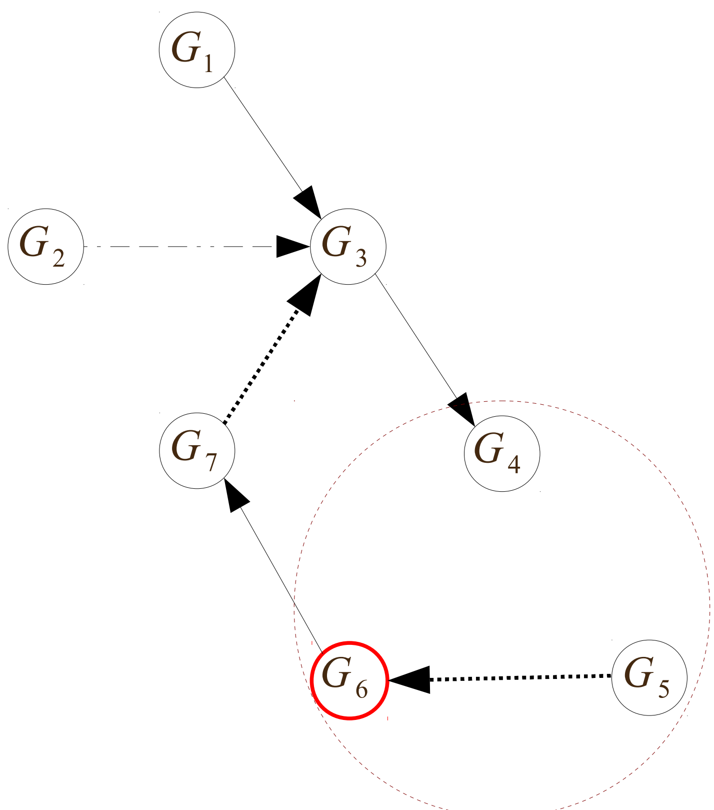
Step 2. Try to swap this edge

Swap* Operator

$Swap(G_2, \boldsymbol{G_3}, G_7)?$ $\longrightarrow$ $Cycle\{G_3, G_4, G_6, G_7\}$

Current situation

$\Delta_{score}Add(G_7, G_3|G_1) > \Delta_{score}Add(G_2, G_3|G_1) > 0$



Objective : delete the cycles

Step 1. Try to delete the edge minimizing $\Delta_{score}Add$

Improving score ?

Yes → Acyclic ?

Yes → OK !

No → Return to Step 1

No → Continue to Step 2

Step 2. Try to swap this edge

Improving score ?

Yes → Acyclic ?

Yes → OK !

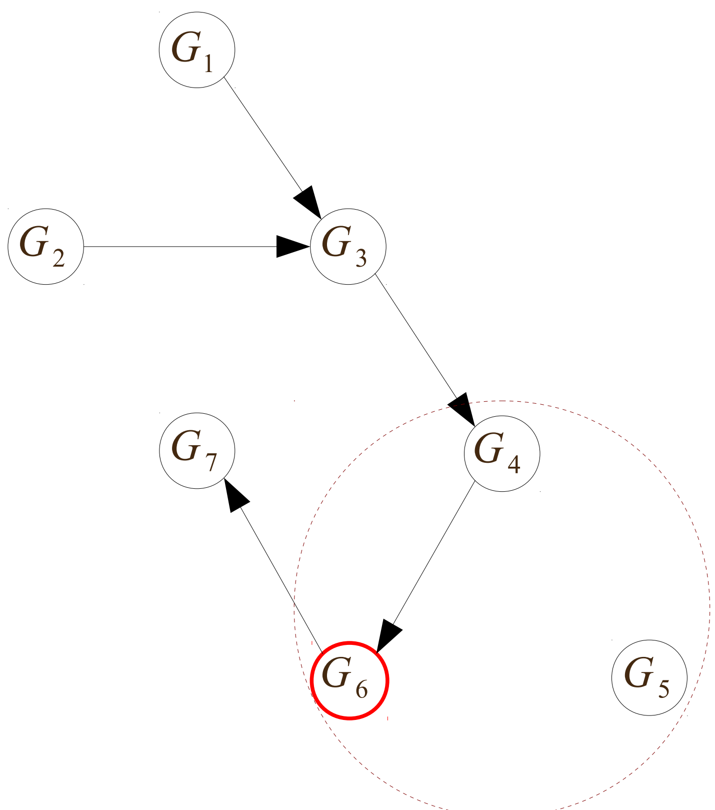
No → Return to Step 1

Swap* Operator

$Swap(G_2, \boldsymbol{G_3}, G_7)?$ $\longrightarrow$ $Cycle\{G_3, G_4, G_6, G_7\}$

Current situation

$\Delta_{score} Add(G_7, G_3|G_1) > \Delta_{score} Add(G_2, G_3|G_1) > 0$



Objective : delete the cycles

Step 1. Try to delete the edge minimizing $\Delta_{score} Add$

Improving score ?

- Yes → Acyclic ?
  - Yes → OK !
  - No → Return to Step 1
- No → Continue to Step 2

Step 2. Try to swap this edge

Improving score ?

- Yes → Acyclic ?
  - Yes → OK !
  - No → Return to Step 1
- **No → Game Over !**

SGS algorithms

- **SGS^1**: Addition + Deletion + Reversal
- **SGS^2**: Addition + Deletion + Reversal + Swap
- **SGS^3**: Addition* + Deletion + Reversal* + Swap*

- One parameter: number of restarts **r**

Experimental settings

- 4 benchmark networks:

	Alarm	*Insurance*	*Hailfinder*	*Pigs*
Nodes	37	27	56	441
Edges	46	52	66	592
In-degree	4	3	4	2

- Data generated from conditional probabilities:
  100 datasets with 500 and 5 000 sample sizes

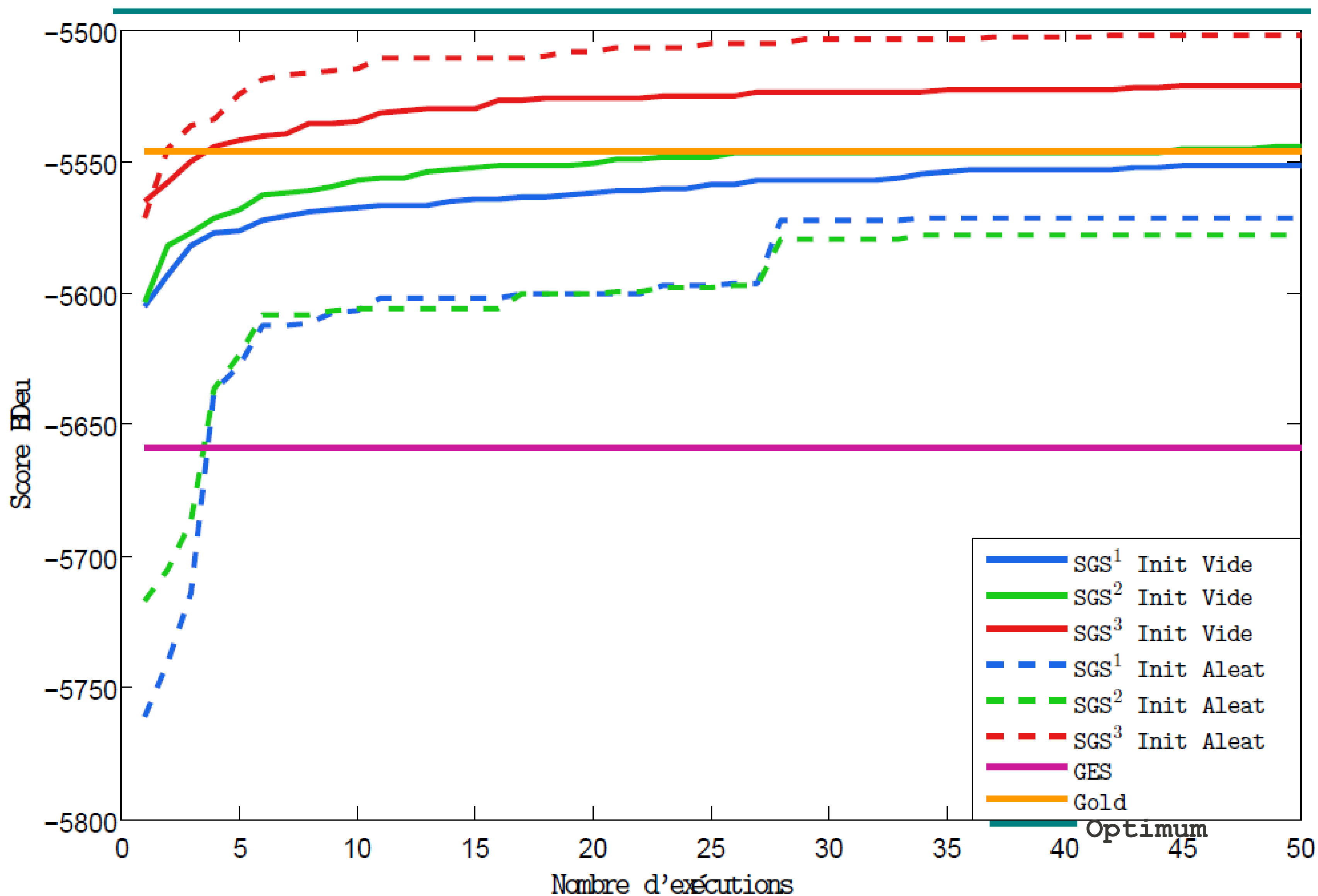
- **SGS** compared to: **LAGD** (2 look-ahead in 5 directions)
  GES

- Limit number of parents : 5

- Pre-filtering candidate parents under condition for Pigs network with SGS
  $$\Delta\, Add(Parent\,, Target) > 0$$

Results (1/4)

Impact of the number of restarts

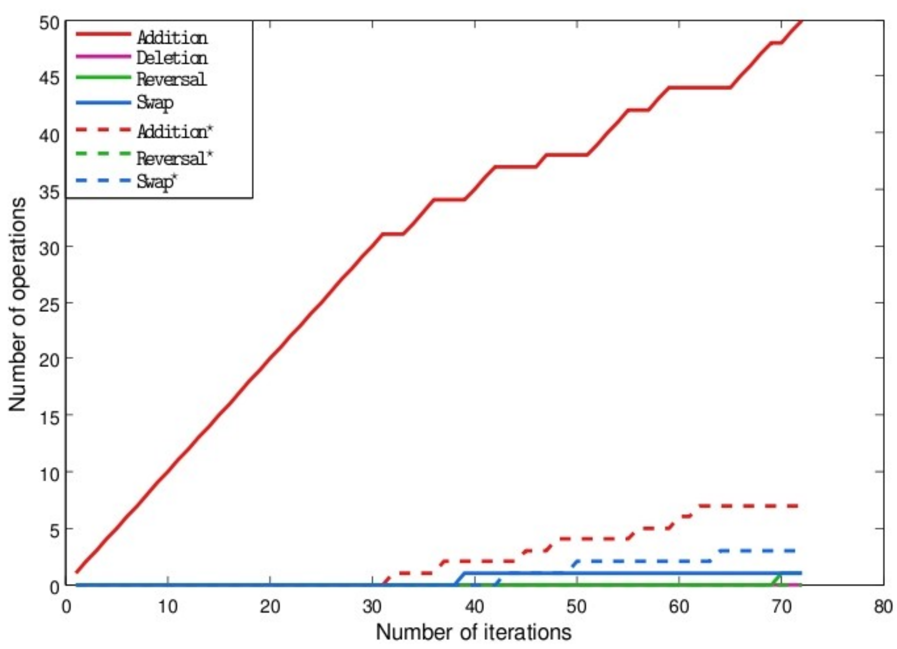


Alarm network (37 variables). Mean over 30 datasets with 500 samples

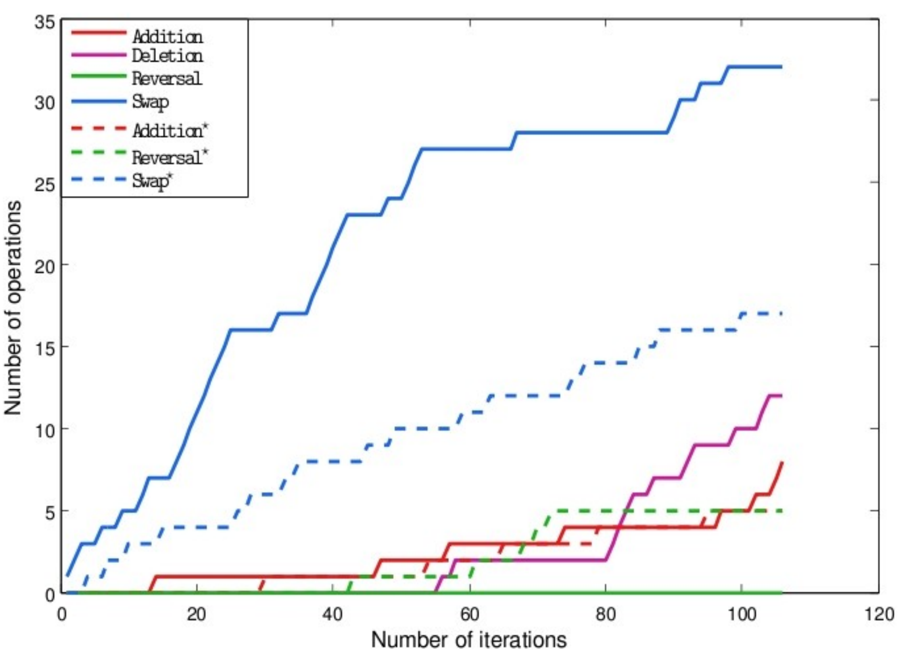
Results (2/4)

- Number of applied operators by type during the search
- *Alarm* network
- **1** run of **SGS³** (r=1) with 500 samples
- **SGS³** Initialized with **empty** and **random network** (2 parents max)

empty network



random network



Results (2/4)

- Comparison of BDeu scores reached by **SGS³**, **LAGD** and **GES**
- ***4*** benchmark networks, **500** and **5 000** samples
- Best of **10** runs for **SGS** and **LAGD** (r=10)
- **All methods** Initialized with a **empty network**

Wilcoxon test 5%	*Alarm*		*Insurance*		*Hailfinder*		*Pigs*	
	500	5 000	500	5 000	500	5 000	500	5 000
SGS³ vs **GES**	+	+	+	+	+	+	+	-
SGS³ vs **LAGD**	+	+	+	+	~	+	n/a	n/a
LAGD vs **GES**	+	~	+	+	+	+	n/a	n/a

Results (3/4)

- Comparison of **Hamming distances** for **SGS³**, **LAGD** and **GES**

Hamming distance = False Positive + False Negative

	Alarm		*Insurance*		*Hailfinder*		*Pigs*	
	500	5 000	500	5 000	500	5 000	500	5 000
SGS³	**11***	8	**24***	**10***	41	**29***	32	41
LAGD	15	10	**24***	16	47	39	n/a	n/a
GES	**11***	**6***	25	15	**39***	33	**9***	**0***

* best result

Results (4/4)

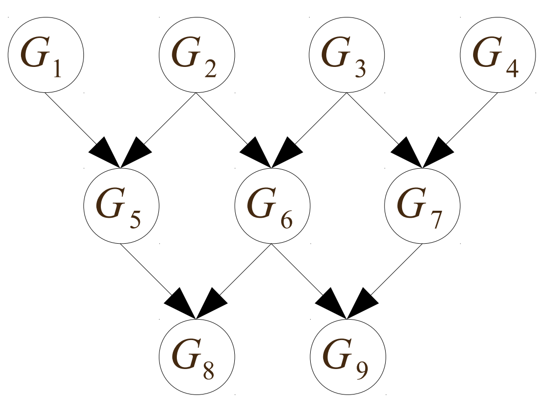
- Comparison of **Hamming distances** for **SGS³**, **LAGD** and **GES**

Hamming distance = False Positive + False Negative

	Alarm		*Insurance*		*Hailfinder*		*Pigs*	
	500	5 000	500	5 000	500	5 000	500	5 000
SGS³	**11***	8	**24***	**10***	41	**29***	32	41
LAGD	15	10	**24***	16	47	39	n/a	n/a
GES	**11***	**6***	25	15	**39***	33	**9***	**0***

* best result

Pigs network



Results (4/4)

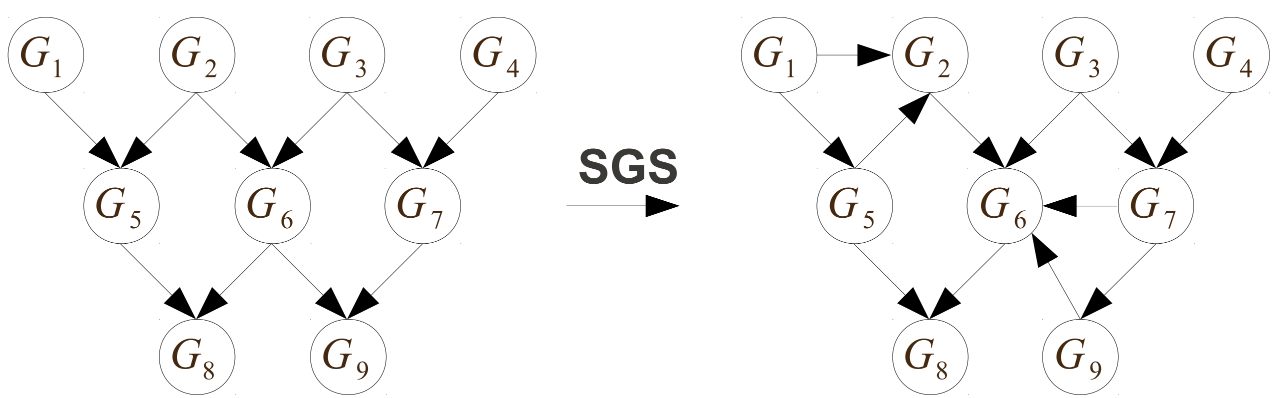
- Comparison of **Hamming distances** for **SGS**, **LAGD** and **GES**

Hamming distance = False Positive + False Negative

	Alarm		*Insurance*		*Hailfinder*		*Pigs*	
	500	5 000	500	5 000	500	5 000	500	5 000
SGS[3]	**11***	8	**24***	**10***	41	**29***	32	41
LAGD	15	10	**24***	16	47	39	n/a	n/a
GES	**11***	**6***	25	15	**39***	33	**9***	**0***

* best result

Pigs network



Results (4/4)

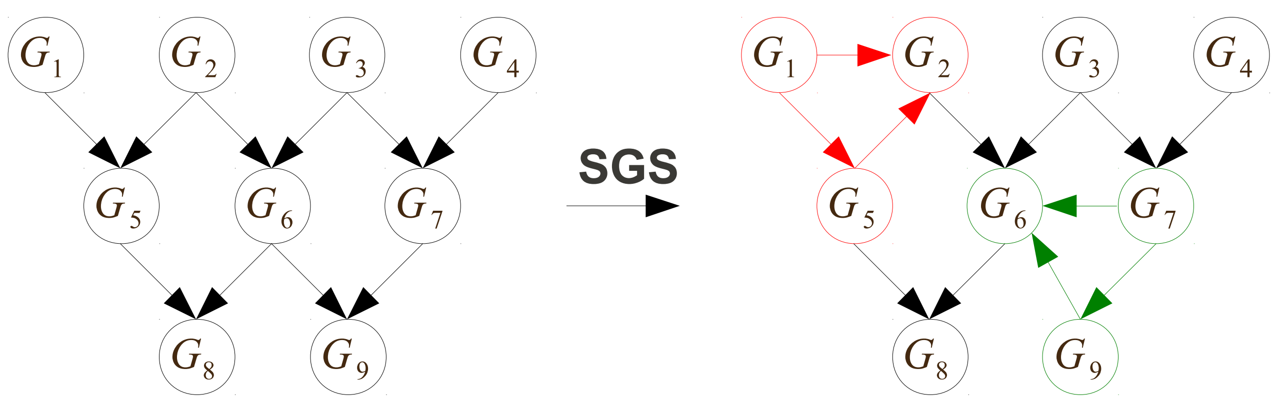
- Comparison of **Hamming distances** for **SGS**, **LAGD** and **GES**

Hamming distance = False Positive + False Negative

	Alarm		*Insurance*		*Hailfinder*		*Pigs*	
	500	5 000	500	5 000	500	5 000	500	5 000
SGS³	**11***	8	**24***	**10***	41	**29***	32	41
LAGD	15	10	**24***	16	47	39	n/a	n/a
GES	**11***	**6***	25	15	**39***	33	**9***	**0***

* best result

Pigs network



Results (4/4)

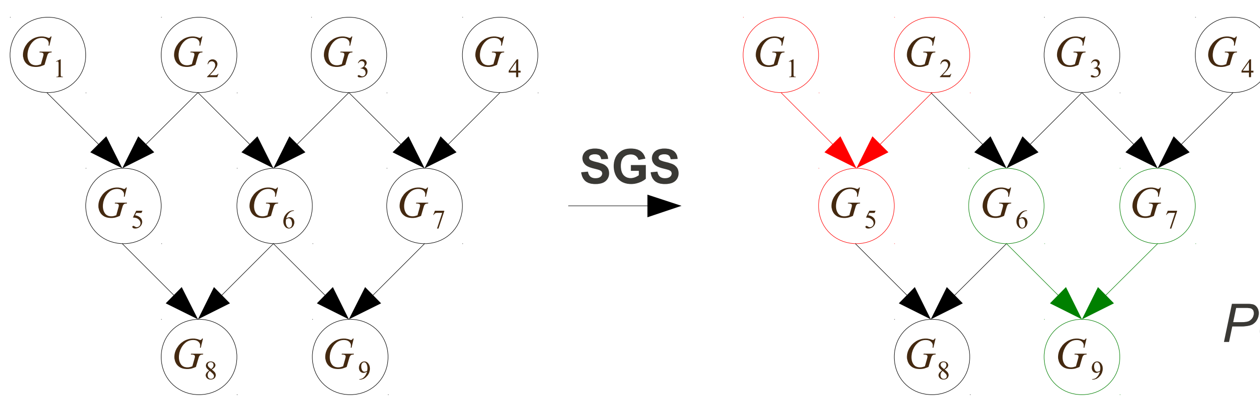
- Comparison of **Hamming distances** for **SGS**, **LAGD** and **GES**

Hamming distance = False Positive + False Negative

	Alarm		*Insurance*		*Hailfinder*		*Pigs*	
	500	5 000	500	5 000	500	5 000	500	5 000
SGS[3]	**9***	**4***	**24***	**9***	40	**26***	**1***	2
LAGD	15	10	**24***	16	47	39	n/a	n/a
GES	11	6	25	15	**39***	33	**9**	**0***

* best result

Pigs network



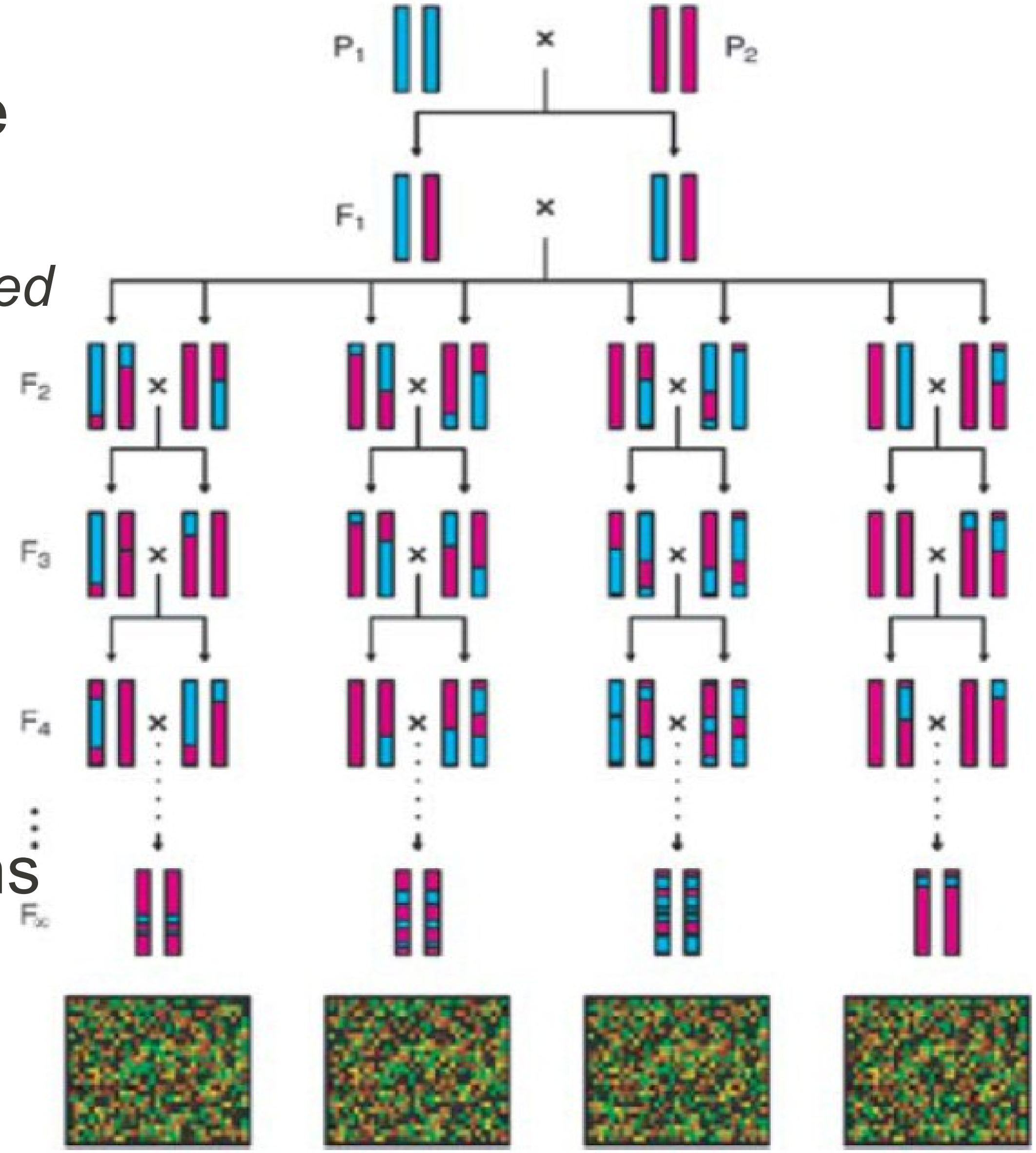
Post-processing

Genetical genomics

- Gene expressions vary due to polymorphisms

(stationary phenomenon in controlled environment)

- Data
  - expression levels
  - genotypes
  - marker/gene localisations on the genome



(Jansen & Nap, Trends in Gen. 2001)

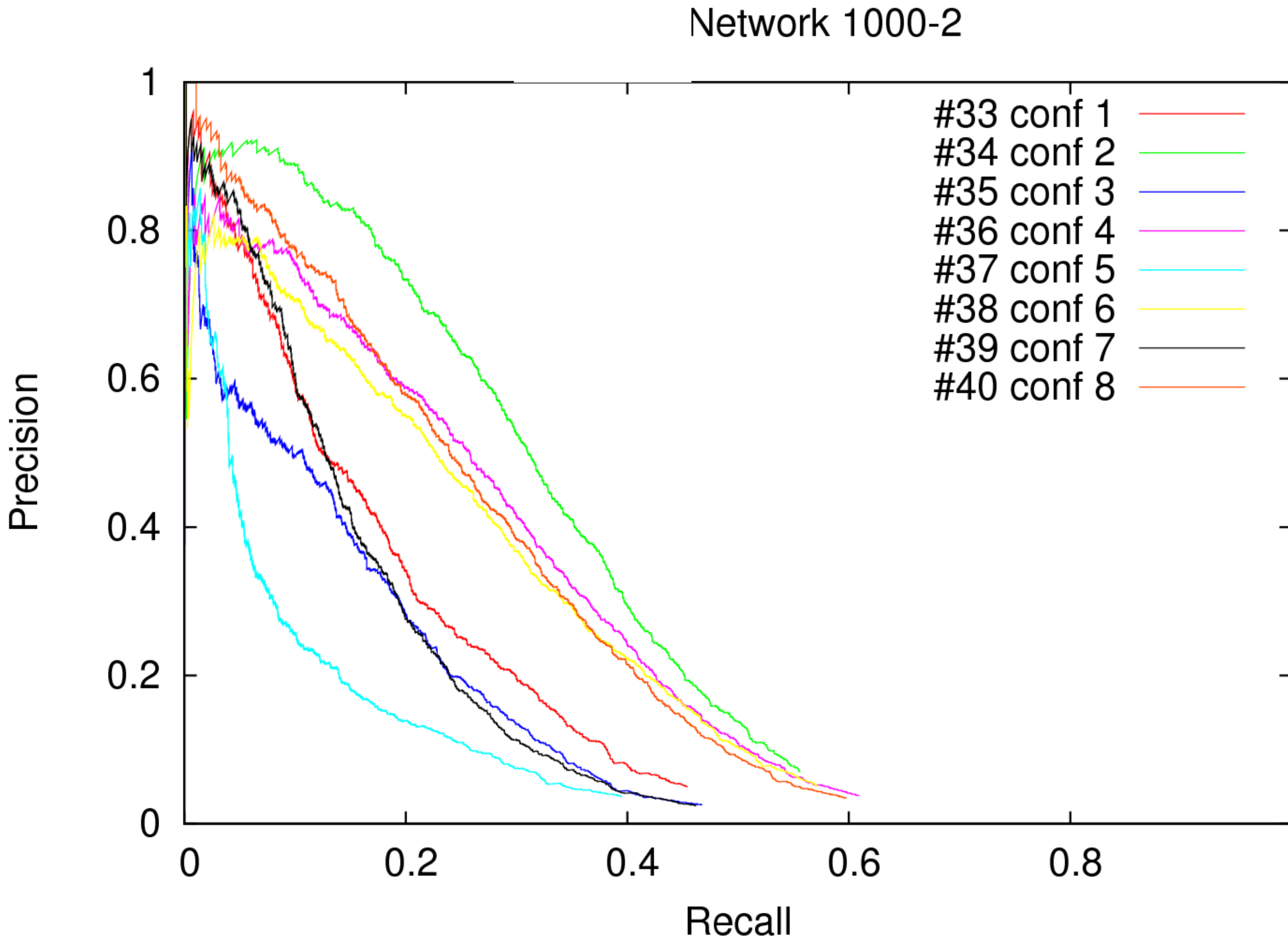
DREAM 2012
StatSeq Systems Genetics Benchmark

- 72 datasets: 9 gene networks (p=100,1000,5000, e~6p) x 8 configurations

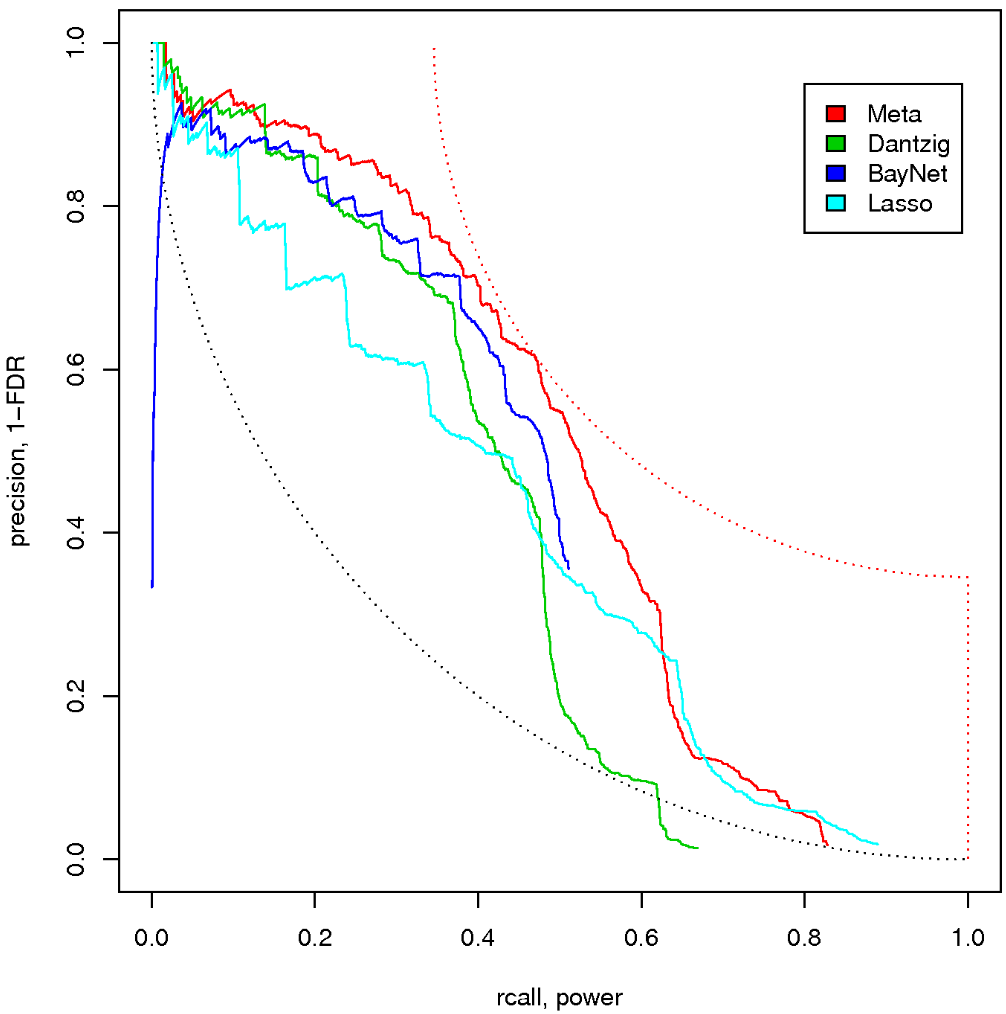
Configuration	Marker Distance	Biological Variance	Heritability	Population Size
1	N(5,1)	N(1,0.1)	High	300
2	N(5,1)	N(1,0.1)	High	900
3	N(5,1)	N(1,0.25)	Low	300
4	N(5,1)	N(1,0.25)	Low	900
5	N(1,0.1)	N(1,0.1)	High	300
6	N(1,0.1)	N(1,0.1)	High	900
7	N(1,0.1)	N(1,0.25)	Low	300
8	N(1,0.1)	N(1,0.25)	Low	900

SysGenSIM (de la Fuente et al, Bioinformatics 2011)

SGS^3 (BDeu alpha=1) with 10 restarts and 100 bootstraps

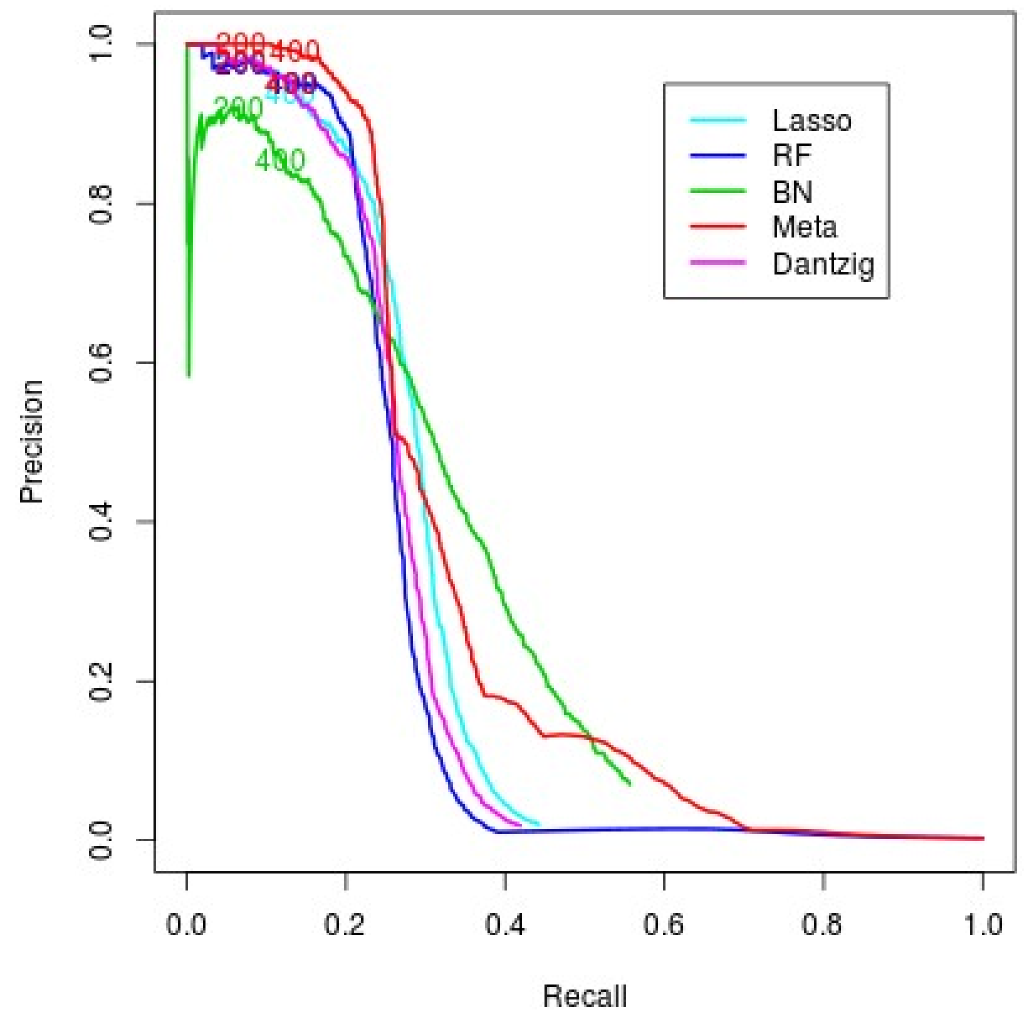


DREAM 2010 / 2012

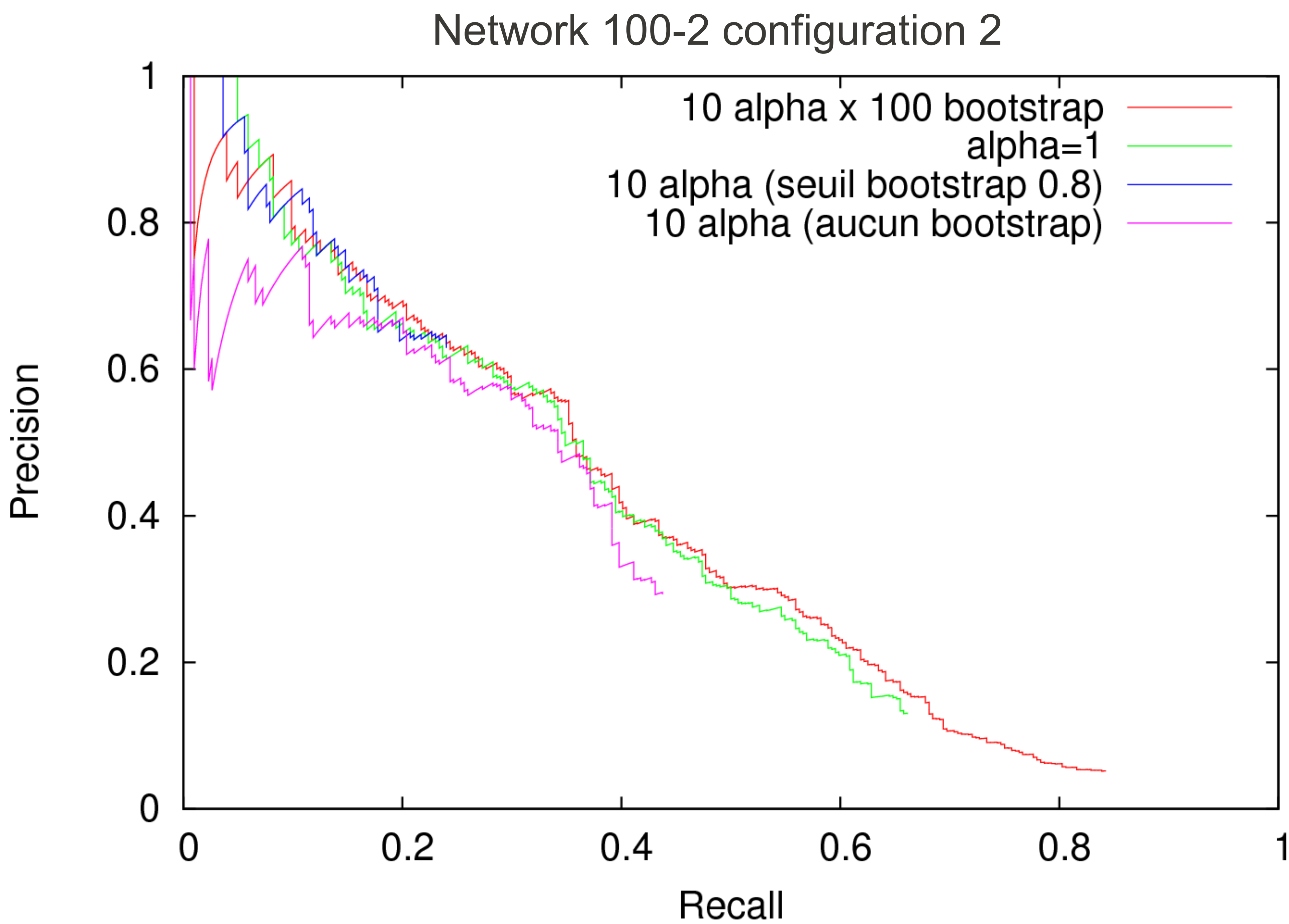


DREAM 2010 Net A1 (p=1000, 999 sample)

(Vignes et al., Plos One, 2011)

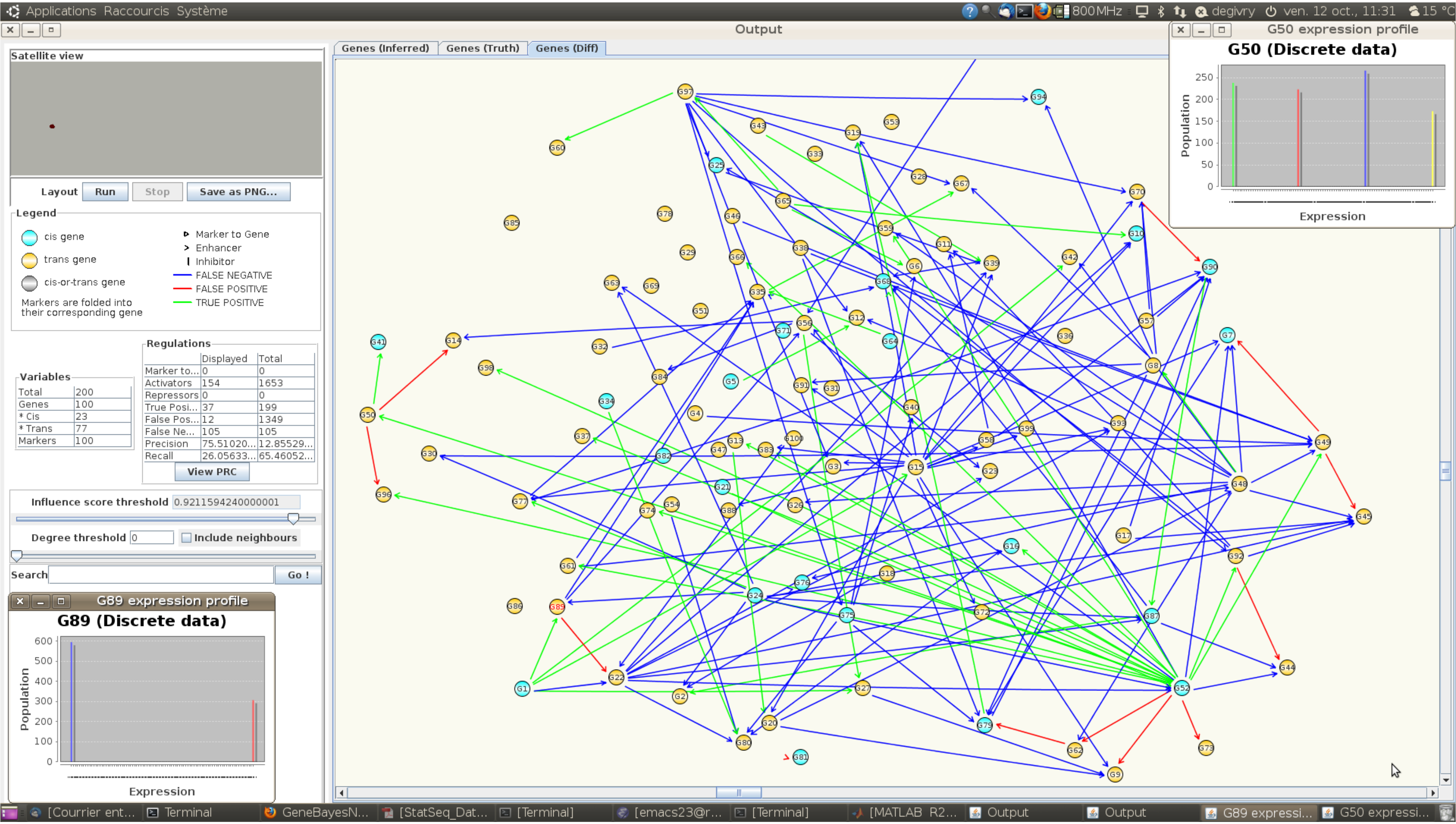


DREAM 2012 Net 1000-2-2 (p=1000, 900 s.)


Network 100-2 configuration 2
Precision
Recall
10 alpha x 100 bootstrap
alpha=1
10 alpha (seuil bootstrap 0.8)
10 alpha (aucun bootstrap)
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1

GeneBayesNet

http://carlit.toulouse.inra.fr/genebayesnet/



Conclusion & Perspectives

We

- Propose a new algorithm SGS
- Propose a new local operator SWAP and iterative extensions for breaking cycles
- Improve BDeu scores of learned networks with these operators
- Compare with other methods on standard BN and genetical genomics benchs

TODO list:

- Reduce the number of restarts **r** required
- Try other meta-heuristics
- Try on real data (arabidopsis thaliana)
-
- Integrate other data sources (bibliome)