ISC Operator for reconstructing Bayesian Network in gene networks context.

Jimmy Vandel & Simon de Givry

Université Paul Sabatier TOULOUSE III

Unité BIA Toulouse
Biométrie et Intelligence Artificielle

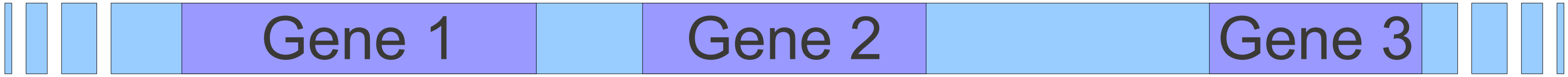
INRA
Institut National de la Recherche Agronomique

Outlines:

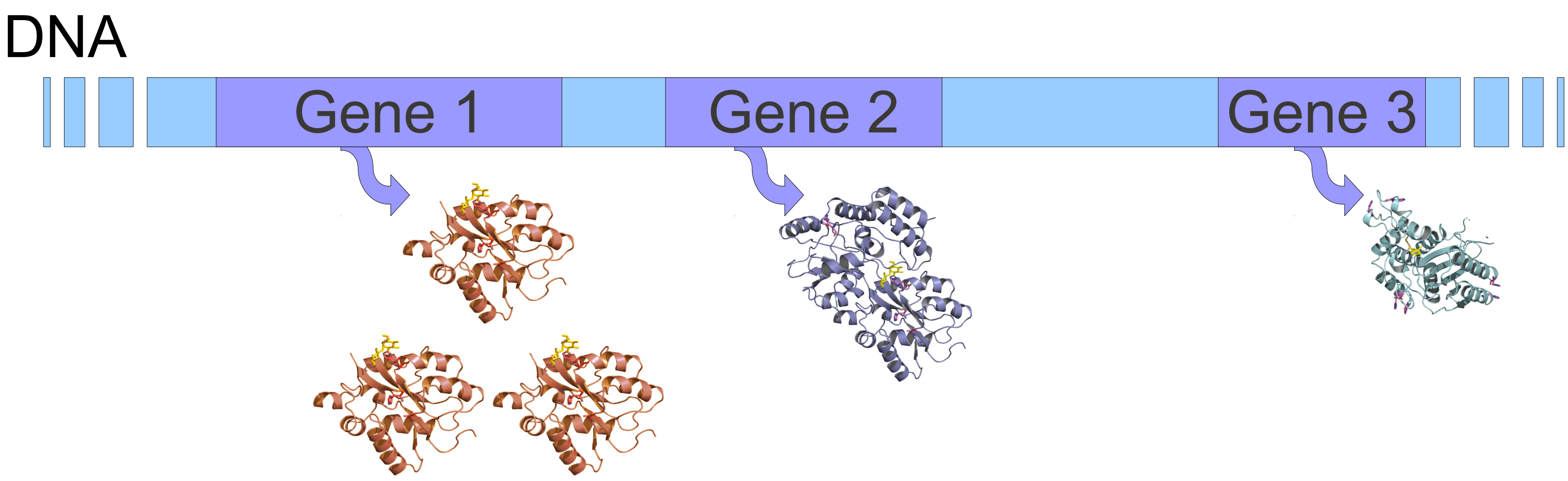
- Biological motivation
- Bayesian Networks framework
- Learning Algorithms
- Local Operators
- Comet language
- Experimentation

Biological motivation

DNA

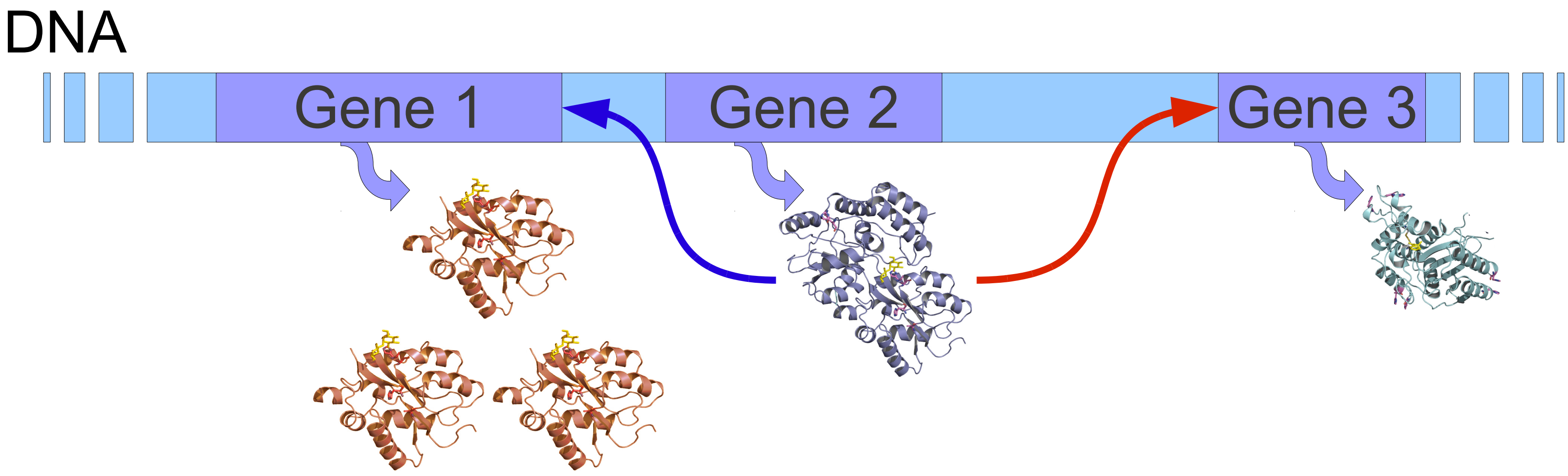


Biological motivation



→ gene expressions (mRNA concentrations)

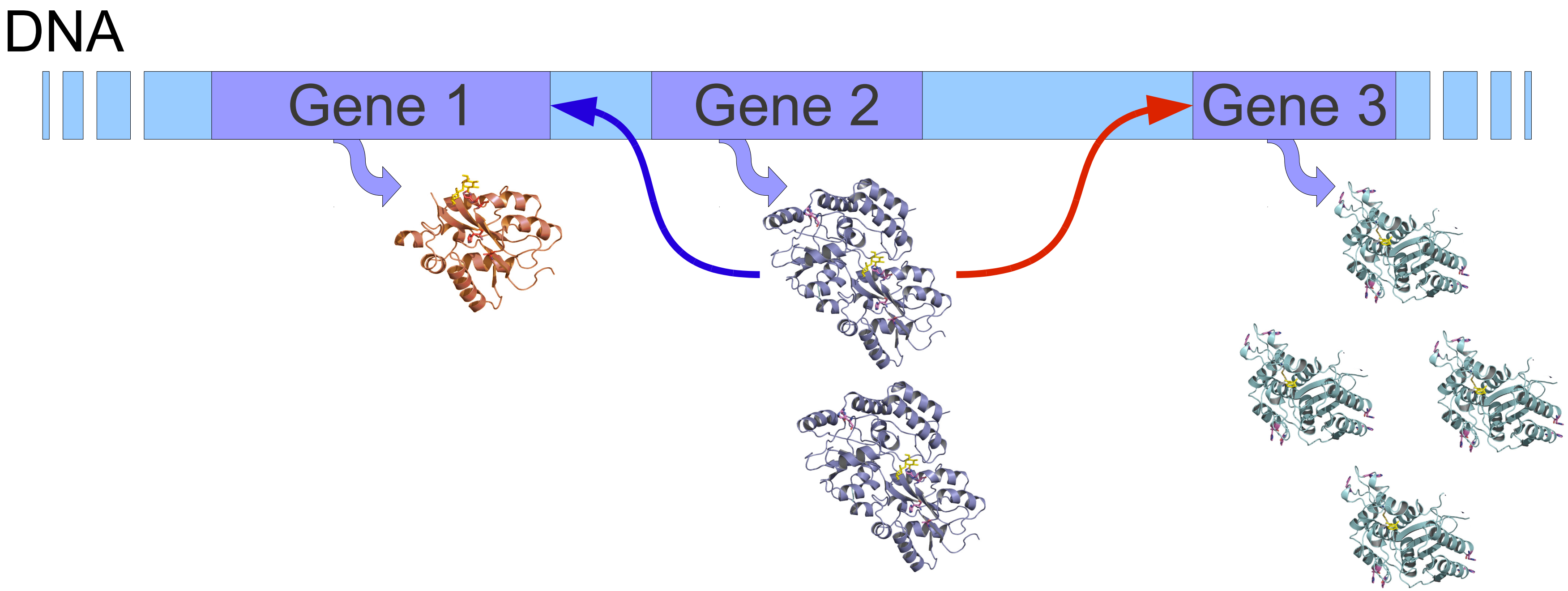
Biological motivation



→ gene expressions (mRNA concentrations)

→ gene regulations

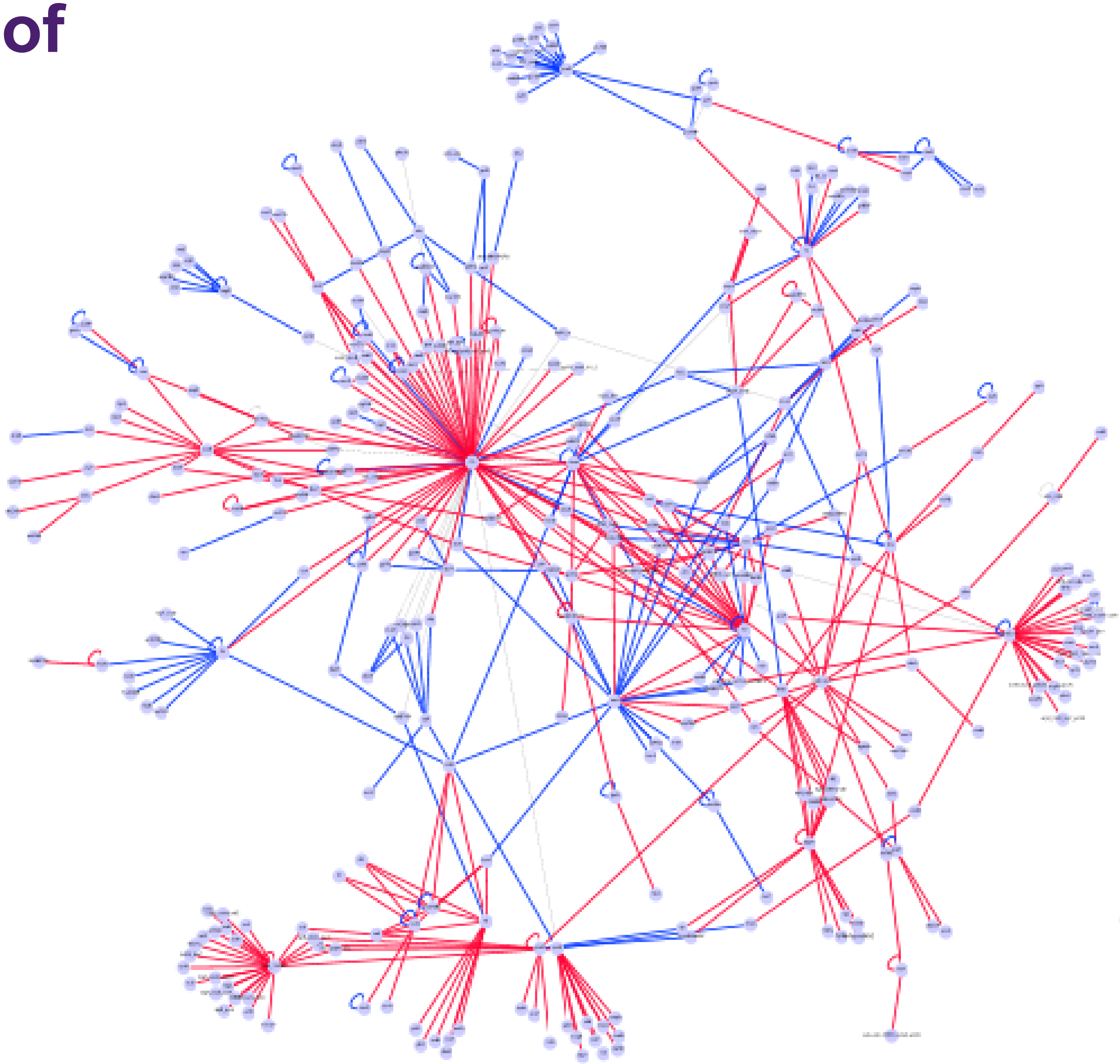
Biological motivation



→ gene expressions (mRNA concentrations)

→ gene regulations

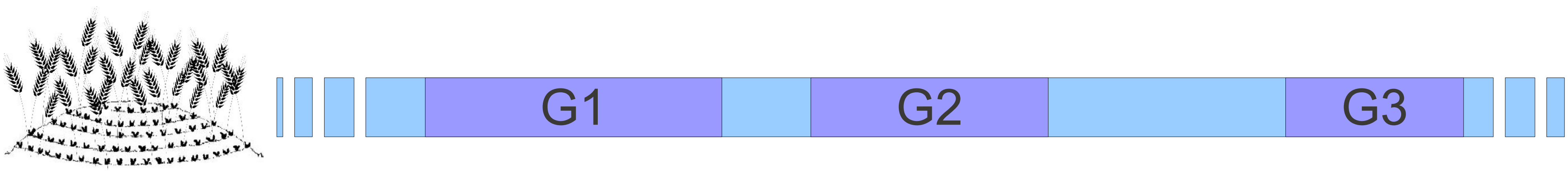
<u>Goal</u> :
Reconstruction of gene regulatory network.



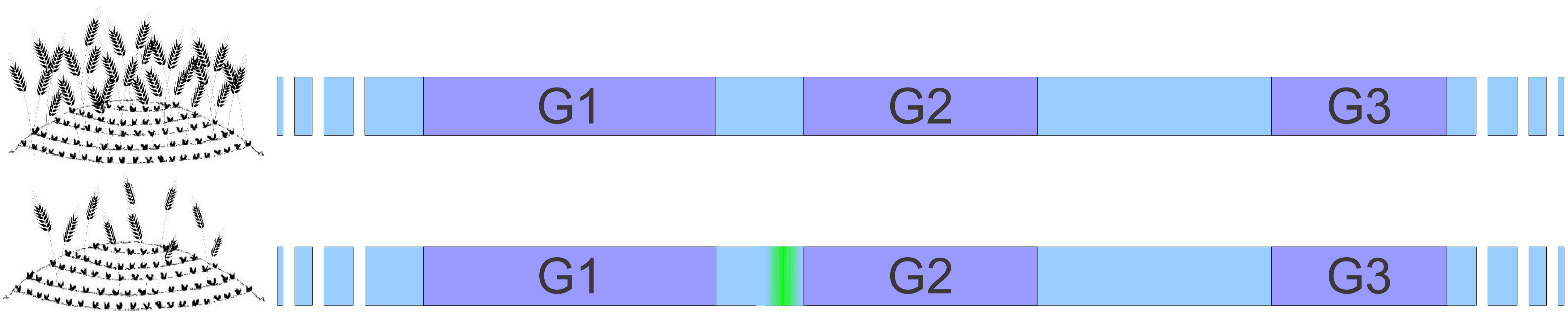
Escherichia coli
(423 genes, 578 regulations)
(SS. Shen-Orr and al., 2002)

Polymorphism

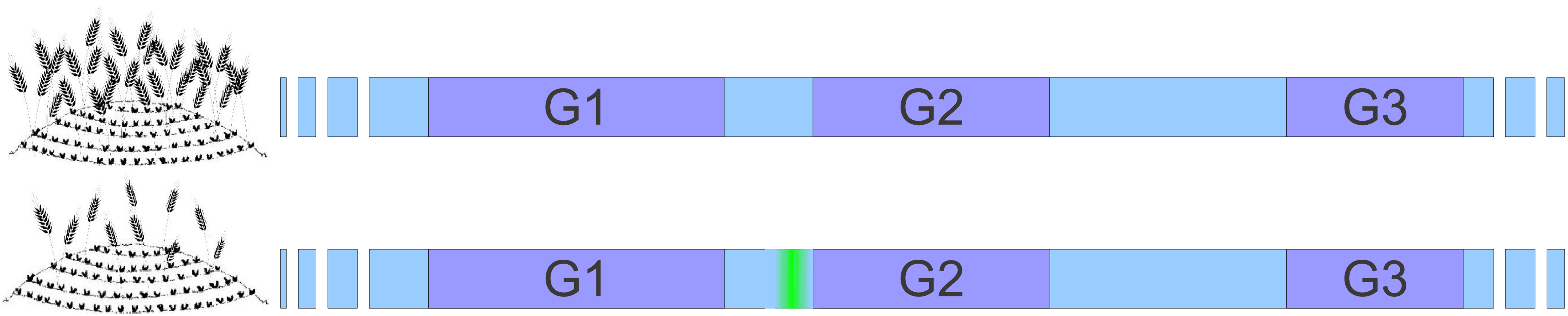
Polymorphism



Polymorphism



Polymorphism

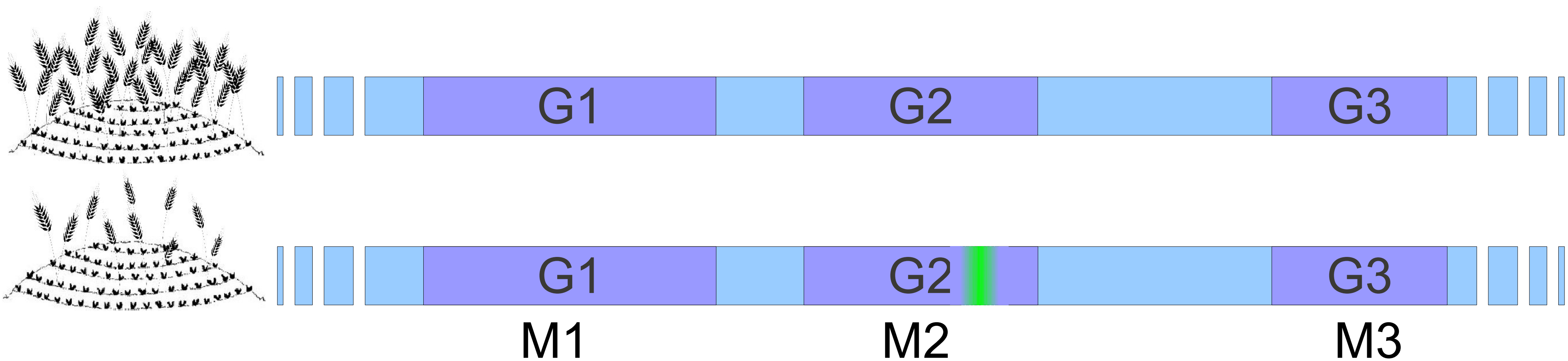


DNA mutations in genes : in promoter region → impact on its gene activity

Polymorphism

DNA mutations in genes: in promoter region → impact on its gene activity
in coding region → impact on others gene activities

Polymorphism

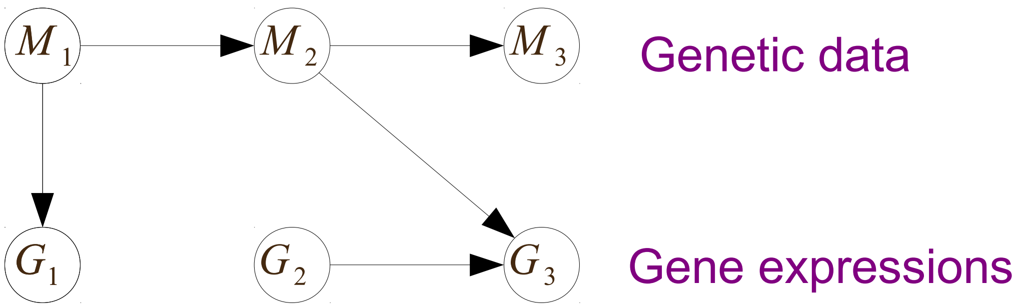


DNA mutations in genes: in promoter region → impact on its gene activity
in coding region → impact on others gene activities

Genetic data from one genetic marker (SNP) for each gene

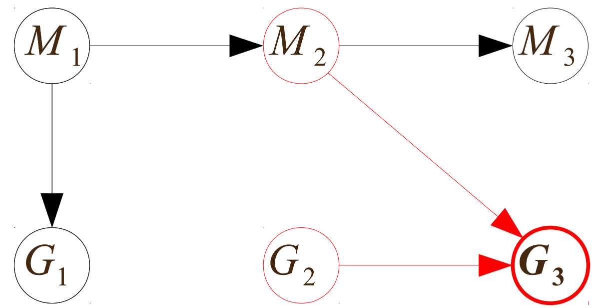
Discrete Bayesian network

Directed acyclic graph G composed of n variables $X_i = \{G_i, M_i\}$



Discrete Bayesian network

Directed acyclic graph G composed of n variables $X_i = \{G_i, M_i\}$



Genetic data

Gene expressions

Conditional distribution $P_G(\mathbf{G_3}/G_{2,} M_2)$

		G_3	$!G_3$
G_2	M_2	0.72	0.28
G_2	$!M_2$	0.59	0.41
$!G_2$	M_2	0.63	0.37
$!G_2$	$!M_2$	0.10	0.90

Graphic representation of a joint probability distribution

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

Learning strategy

We look for the graph $G_{score} = argmax_{G_i} P(G_i / D)$ with dataset D .

$$P(G_i / D) = \frac{P(D / G_i) P(G_i)}{P(D)}$$

$$\propto P(D / G_i) P(G_i)$$

- $P(D / G_i)$:marginal likelihood of *Gi*
- $P(G_i)$:prior probability of the graph G*i*
  $\longrightarrow$ assumed to be uniform

Objective function easy to evaluate and avoids over-fitting

- decomposable and penalized scores
  - BDe score *(**D.Heckerman** Machine learning 1995)*
  - *BIC score (**G.Schwartz** Annals of statistics 1978)*

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
   - hill climbing (with restarts)
   - tabu search
   - simulated annealing
   - MCMC
   - genetic algorithms
   - ...

Local Operators

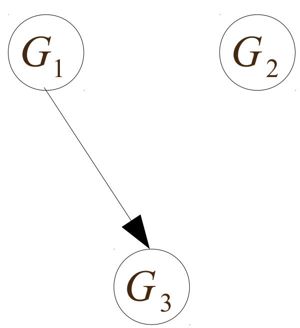
- addition
- deletion
- reversal (deletion + addition on the same pair)
- **swap (deletion + addition including an extra node)**

Local Operators

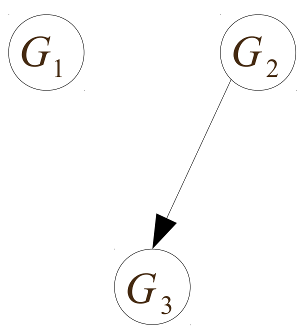
- addition
- deletion
- reversal (deletion + addition on the same pair)
- **swap (deletion + addition including an extra node)**

Example:

Current situation



Target situation

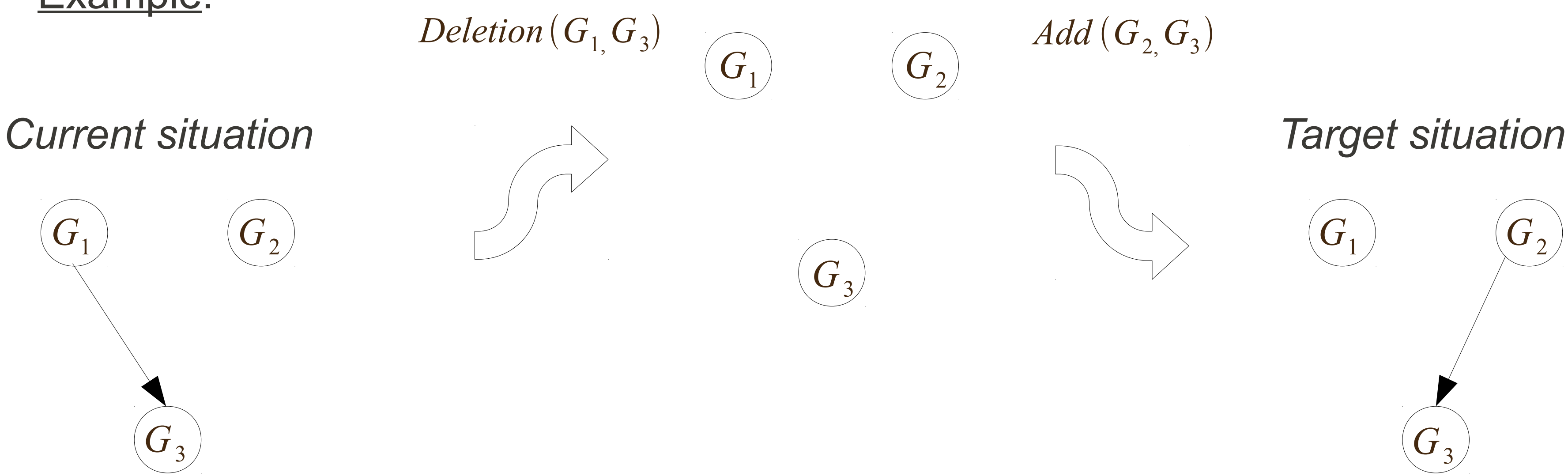


$$\Delta_{score} Add(G_{2,} G_3) > \Delta_{score} Add(G_{1,} G_3) > 0$$

Local Operators

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

Example:

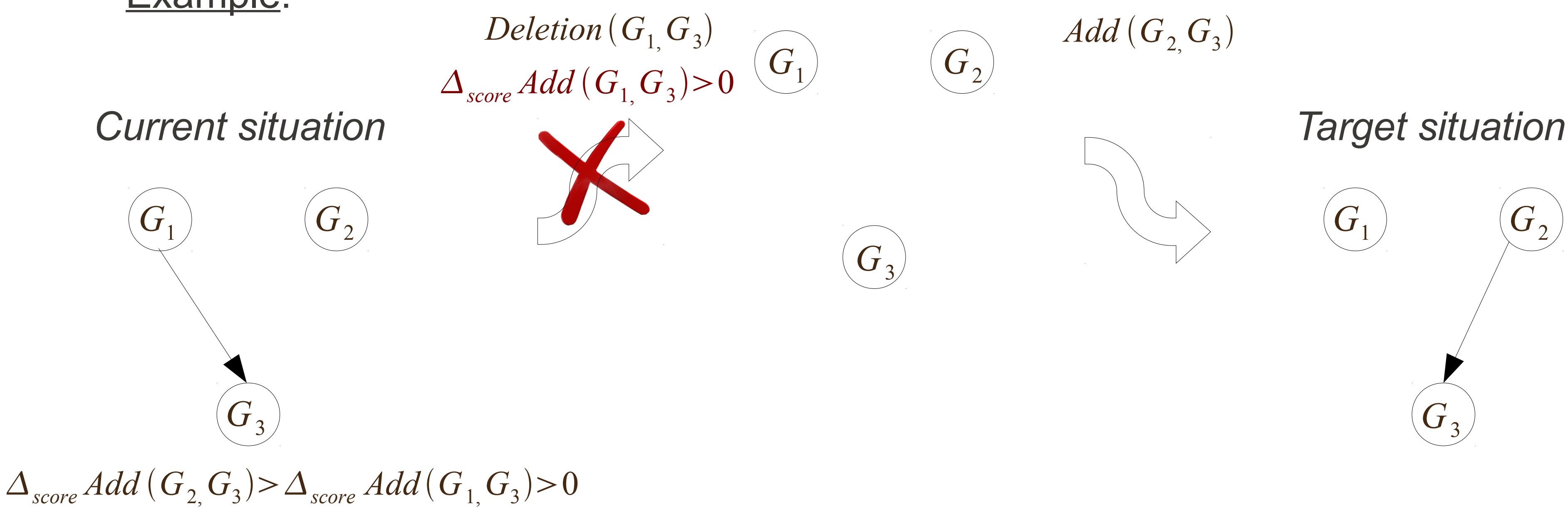


$$\Delta_{score} Add(G_2, G_3) > \Delta_{score} Add(G_1, G_3) > 0$$

Local Operators

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

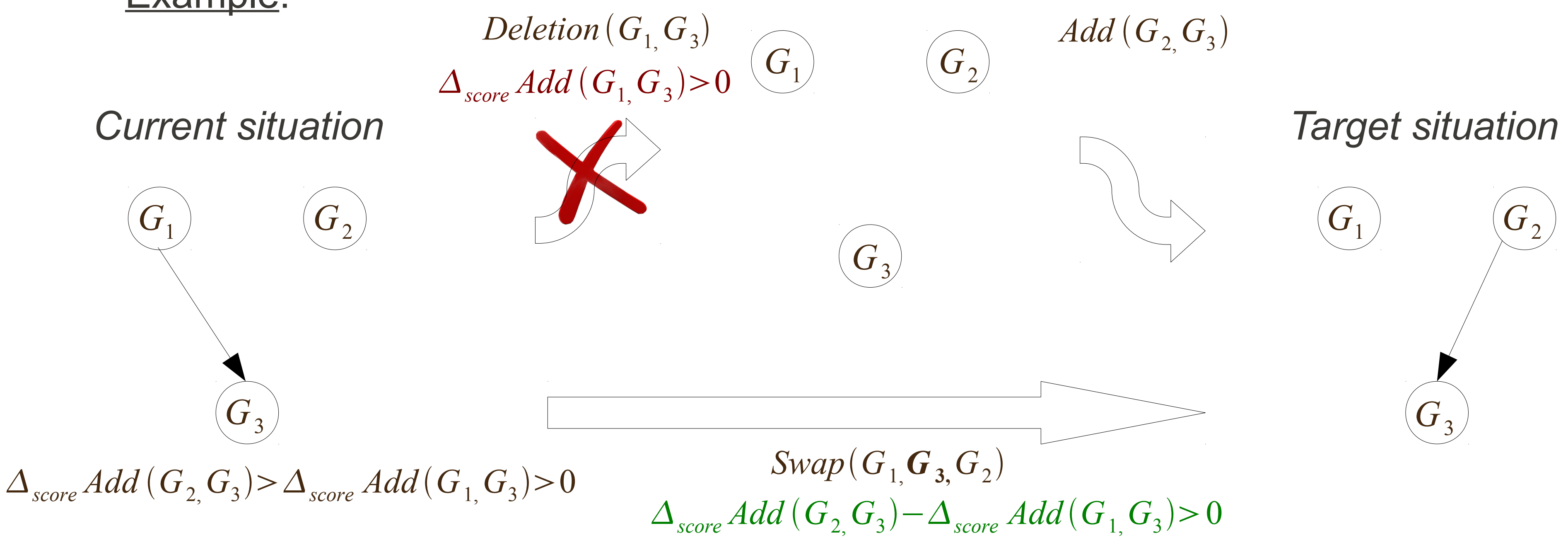
Example:



Local Operators

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

Example:



→ escape from some local maxima

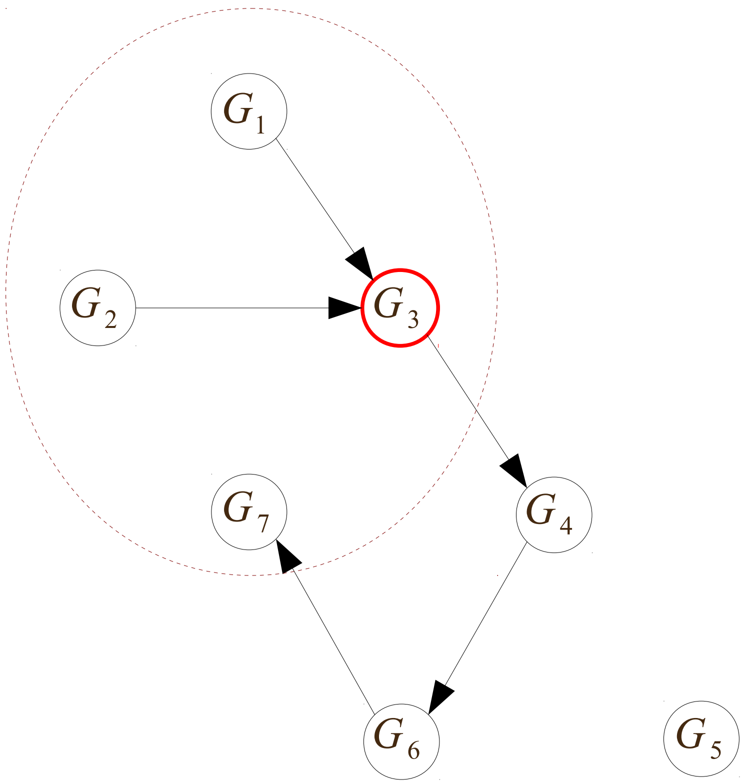
ISC Operator

(Iterative Swap Operator)

$Swap(G_2, \mathbf{G_3}, G_7)?$

Current situation

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



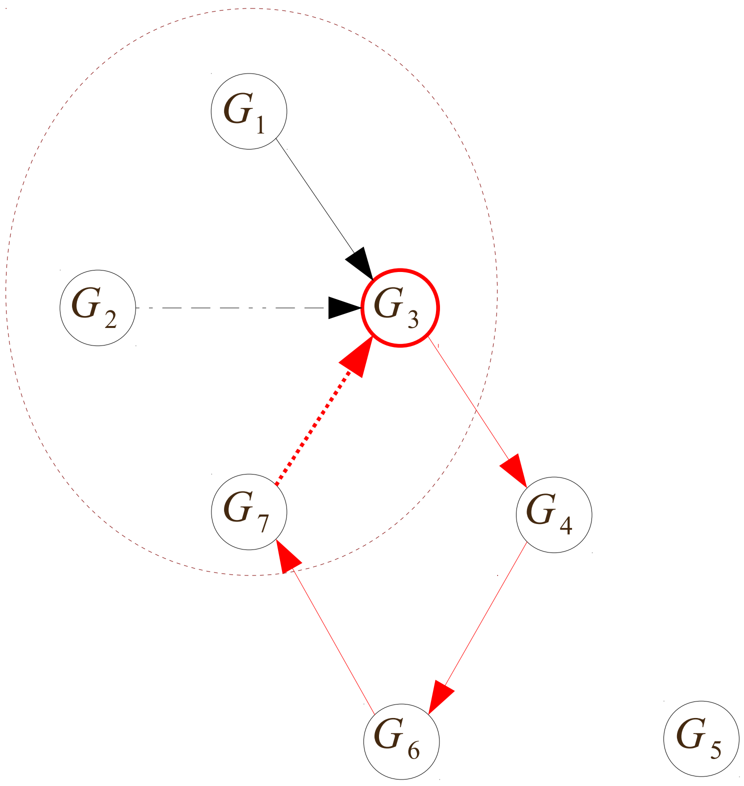
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



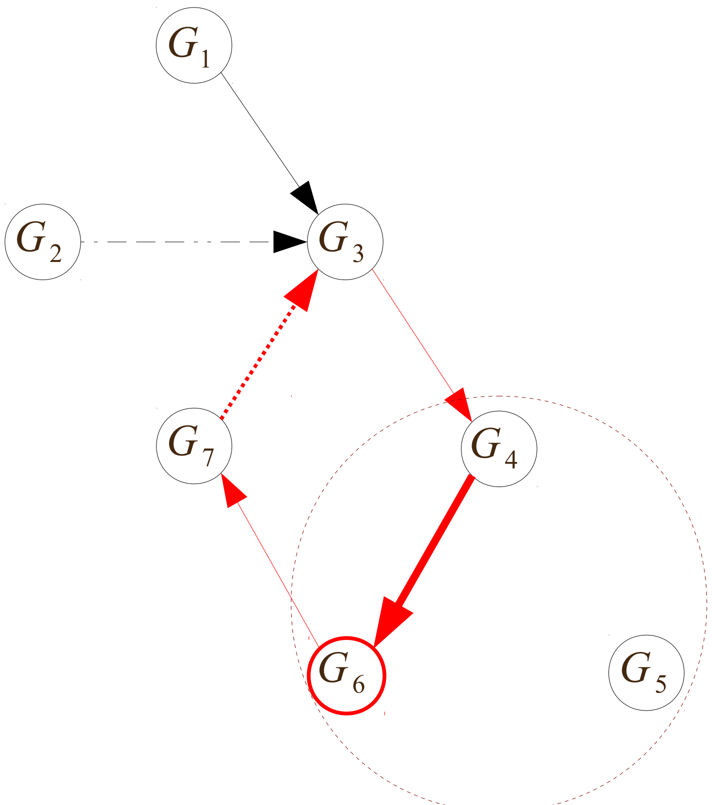
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



While there exist a cycle and ! STOP

Select the edge of the cycle minimizing $\Delta_{score} Add$

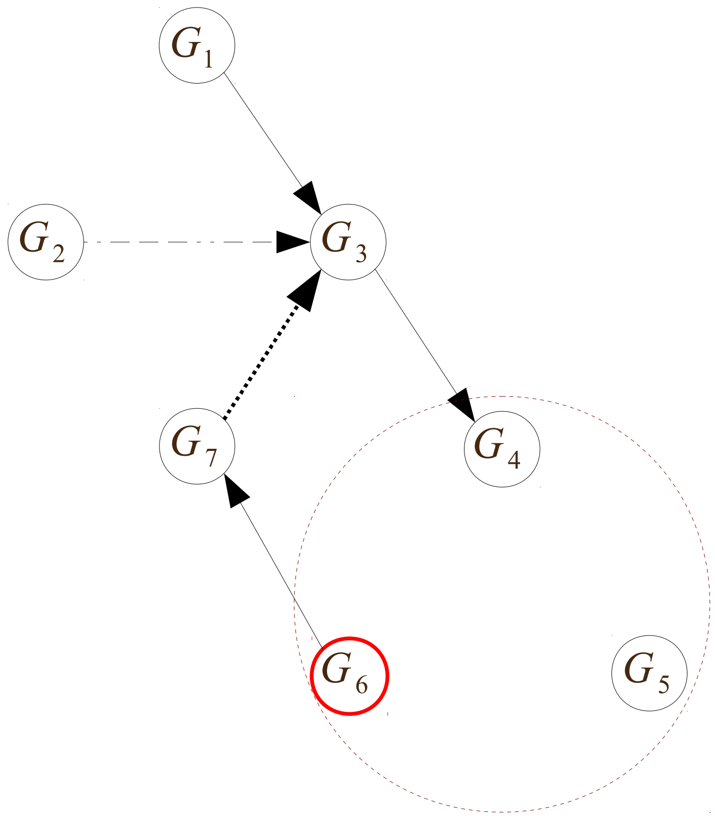
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



While there exist a cycle and ! STOP

Select the edge of the cycle minimizing $\Delta_{score} Add$

Try to delete it

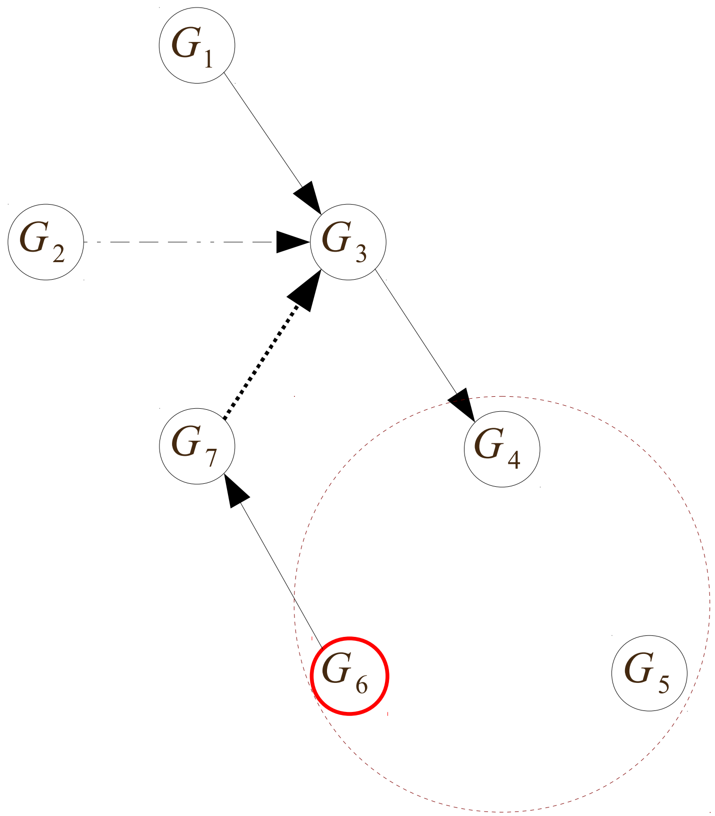
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



While there exist a cycle and ! STOP

- Select the edge of the cycle minimizing $\Delta_{score} Add$
- Try to delete it
- If $\Delta_{score} Swap(G_{2,} \mathbf{G_{3,}} G_7) + \Delta_{score} Deletion(G_{4,} G_6) \leq 0$
- Else
  - Record $Deletion(G_{4,} G_6)$

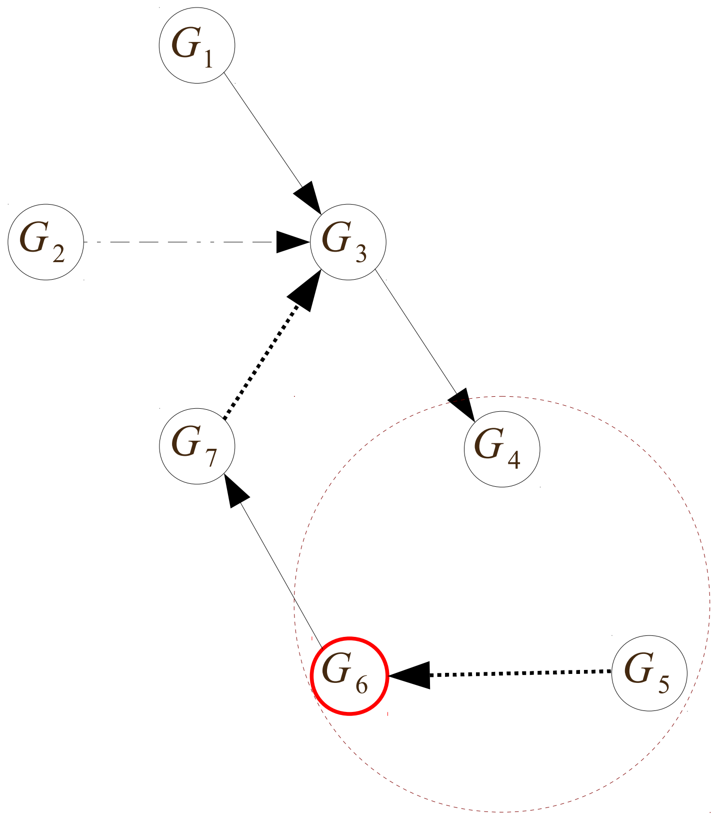
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



While there exist a cycle and ! STOP

- Select the edge of the cycle minimizing $\Delta_{score} Add$
- Try to delete it
- If $\Delta_{score} Swap(G_2, \mathbf{G_3}, G_7) + \Delta_{score} Deletion(G_4, G_6) \leq 0$
  - Try to swap this edge
- Else
  - Record $Deletion(G_4, G_6)$

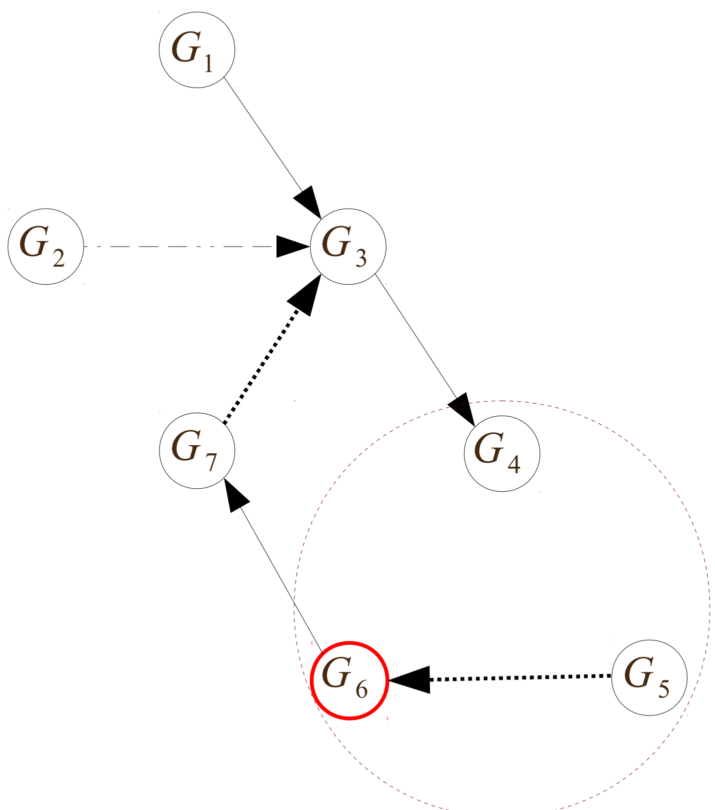
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



While there exist a cycle and ! STOP

- Select the edge of the cycle minimizing $\Delta_{score} Add$
- Try to delete it
- If $\Delta_{score} Swap(G_2, \mathbf{G_3}, G_7) + \Delta_{score} Deletion(G_4, G_6) \leq 0$
  - Try to swap this edge
  - If $\Delta_{score} Swap(G_2, \mathbf{G_3}, G_7) + \Delta_{score} Swap(G_4, \mathbf{G_6}, G_5) \leq 0$
    - STOP
  - Else
    - Record $Swap(G_4, \mathbf{G_6}, G_5)$
- Else
  - Record $Deletion(G_4, G_6)$

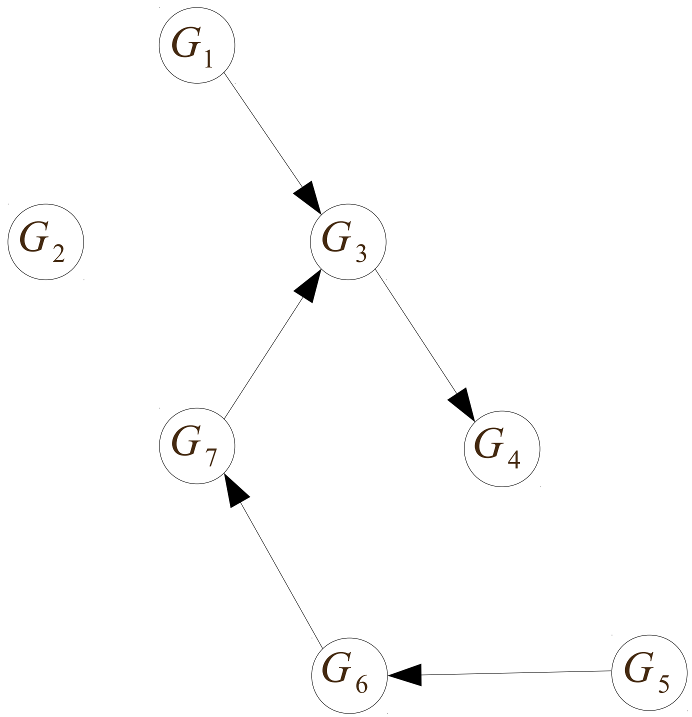
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



While there exist a cycle and ! STOP

- Select the edge of the cycle minimizing $\Delta_{score} Add$
- Try to delete it
- If $\Delta_{score} Swap(G_2, \mathbf{G_3}, G_7) + \Delta_{score} Deletion(G_4, G_6) \leq 0$
  - Try to swap this edge
  - If $\Delta_{score} Swap(G_2, \mathbf{G_3}, G_7) + \Delta_{score} Swap(G_4, \mathbf{G_6}, G_5) \leq 0$
    - STOP
  - Else
    - Record $Swap(G_4, \mathbf{G_6}, G_5)$
- Else
  - Record $Deletion(G_4, G_6)$

If ! STOP

- Validate all recorded moves

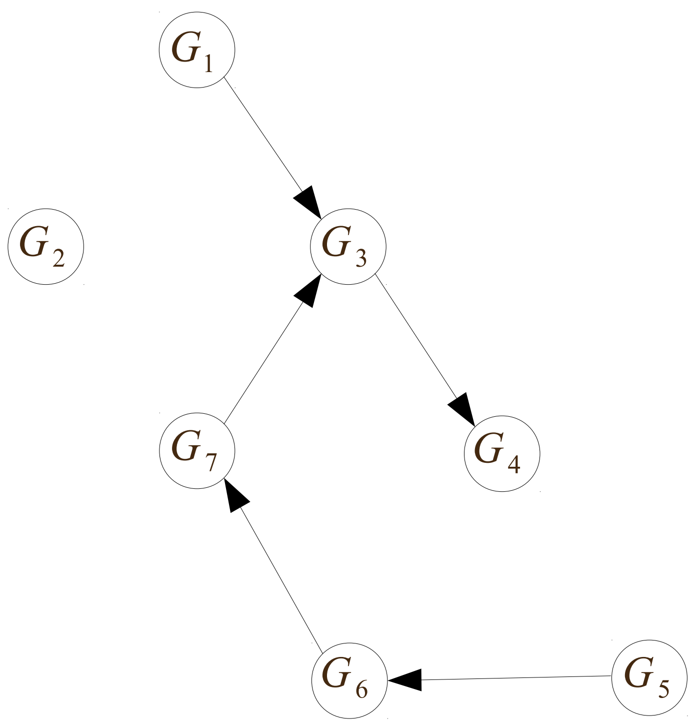
ISC Operator

(Iterative Swap Operator)

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

Current situation

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



~~While there exist a cycle and ! STOP~~ **nISC operator**

- Select the edge of the cycle minimizing $\Delta_{score} Add$
- Try to delete it
- If $\Delta_{score} Swap(G_2, \mathbf{G_3}, G_7) + \Delta_{score} Deletion(G_4, G_6) \leq 0$
  - Try to swap this edge
  - If $\Delta_{score} Swap(G_2, \mathbf{G_3}, G_7) + \Delta_{score} Swap(G_4, \mathbf{G_6}, G_5) \leq 0$
    - STOP
  - Else
    - Record $Swap(G_4, \mathbf{G_6}, G_5)$
- Else
  - Record $Deletion(G_4, G_6)$

If ! STOP

- Validate all recorded moves

Comet Language

Is a

High level programming language

(L.Michel and P.Van Hentenryck, 2002)

http://www.comet-online.org/

Comet Language

Is a

- High level programming language

To

- Model optimization problems
- Implement search procedures

(L.Michel and P.Van Hentenryck, 2002)

http://www.comet-online.org/

Comet Language

Is a

- High level programming language

To

- Model optimization problems
- Implement search procedures

In domains of

- Constraint programming
- Constraint-Based Local search

(L.Michel and P.Van Hentenryck, 2002)

http://www.comet-online.org/

Comet Language

Is a

- High level programming language

To

- Model optimization problems
- Implement search procedures

In domains of

- Constraint programming
- Constraint-Based Local search

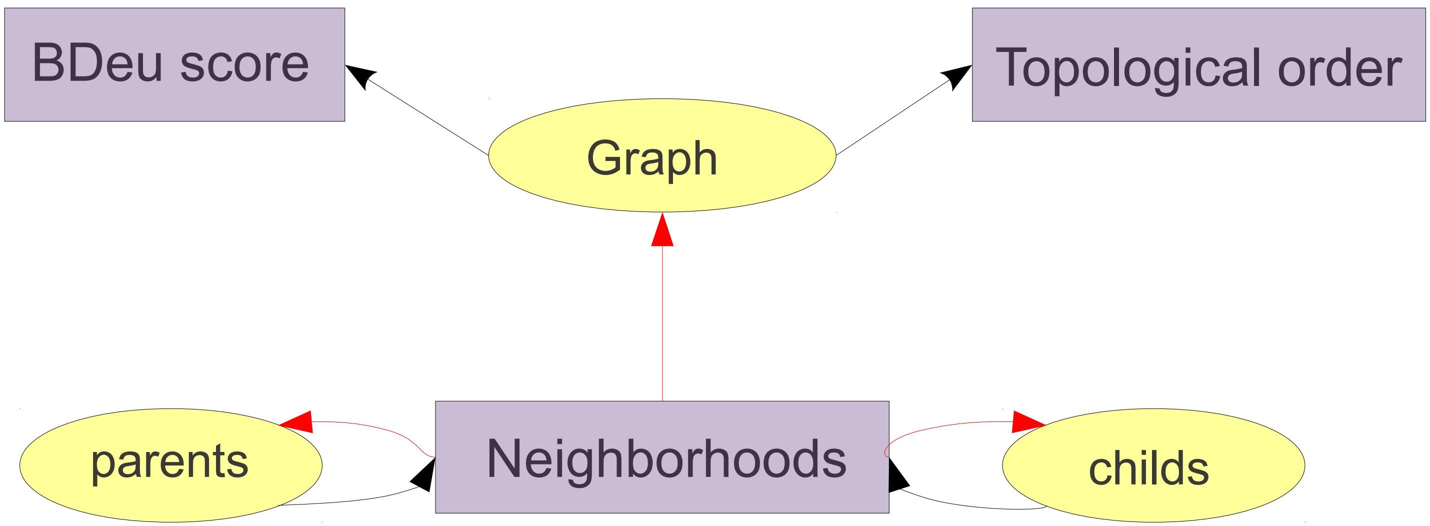
Offering easy implementation for

- Invariants
- Objective functions
- Constraints definition
- Parallel programming
- ...

(L.Michel and P.Van Hentenryck, 2002)

http://www.comet-online.org/

Hill-climbing implementation in Comet



Invariant

Incremental variable

→ update when is modified

→ modify

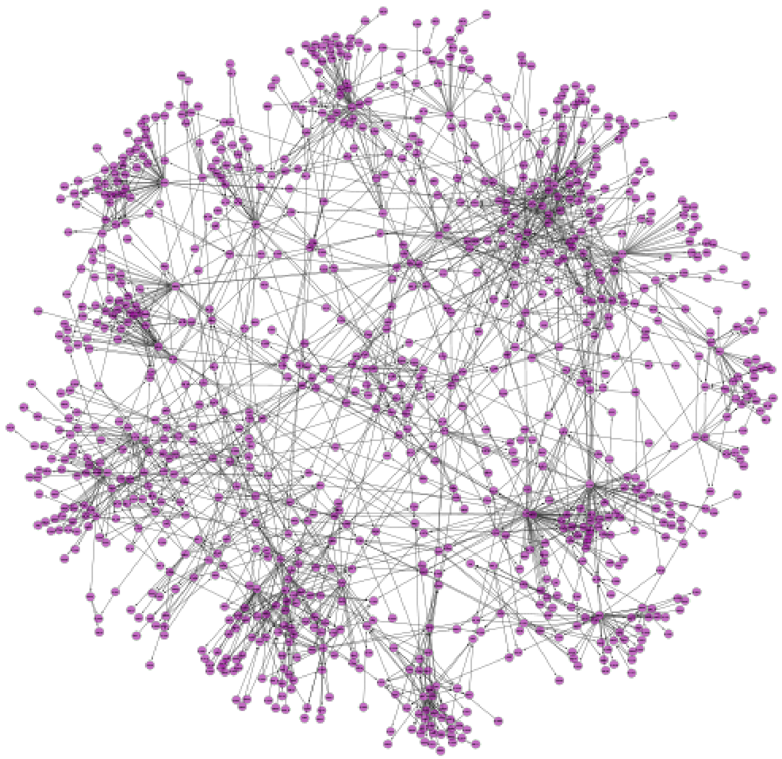
Experimentation

DREAM5 systems genetics challenge (November 2010, New York)

Objective: recover gene regulatory network from
- Gene expressions
- Genetic data

Our gold network
- 2000 nodes (1000 genes / 1000 genetic markers)
- 1983 edges

Simulated population of 300 individuals



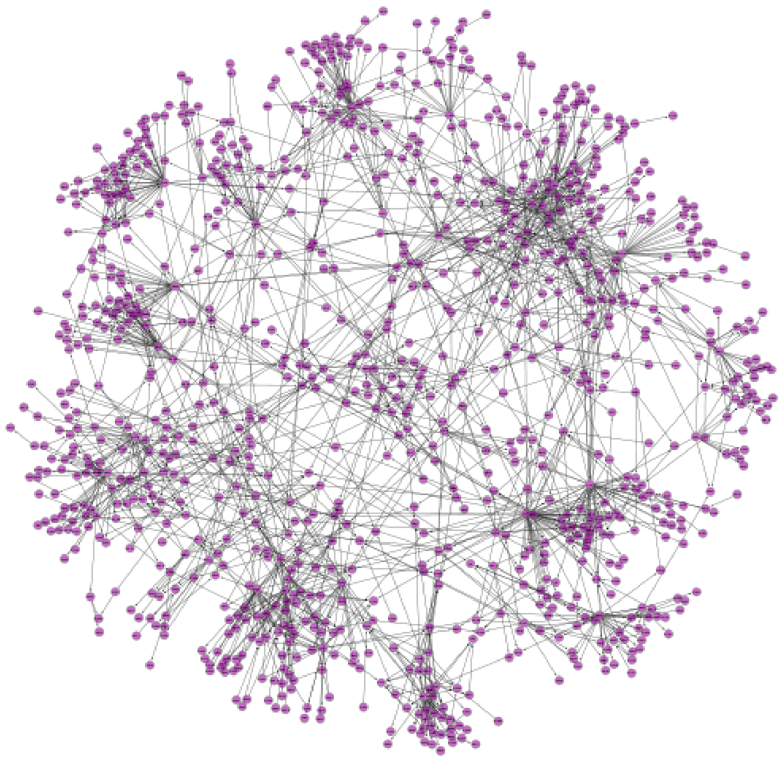
Gold standard network

Experimentation

DREAM5 systems genetics challenge (November 2010, New York)
Objective: recover gene regulatory network from
- Gene expressions
- Genetic data

Our gold network
- 2000 nodes (1000 genes / 1000 genetic markers)
- 1983 edges

Simulated population of 300 individuals



Gold standard network

- Discretization of data (max. 4 classes)
- Pre-filtering candidate parents under condition

$$\Delta Add(Parent, Target) > 0$$

- Limit number of parents : 6

Results (1/4)

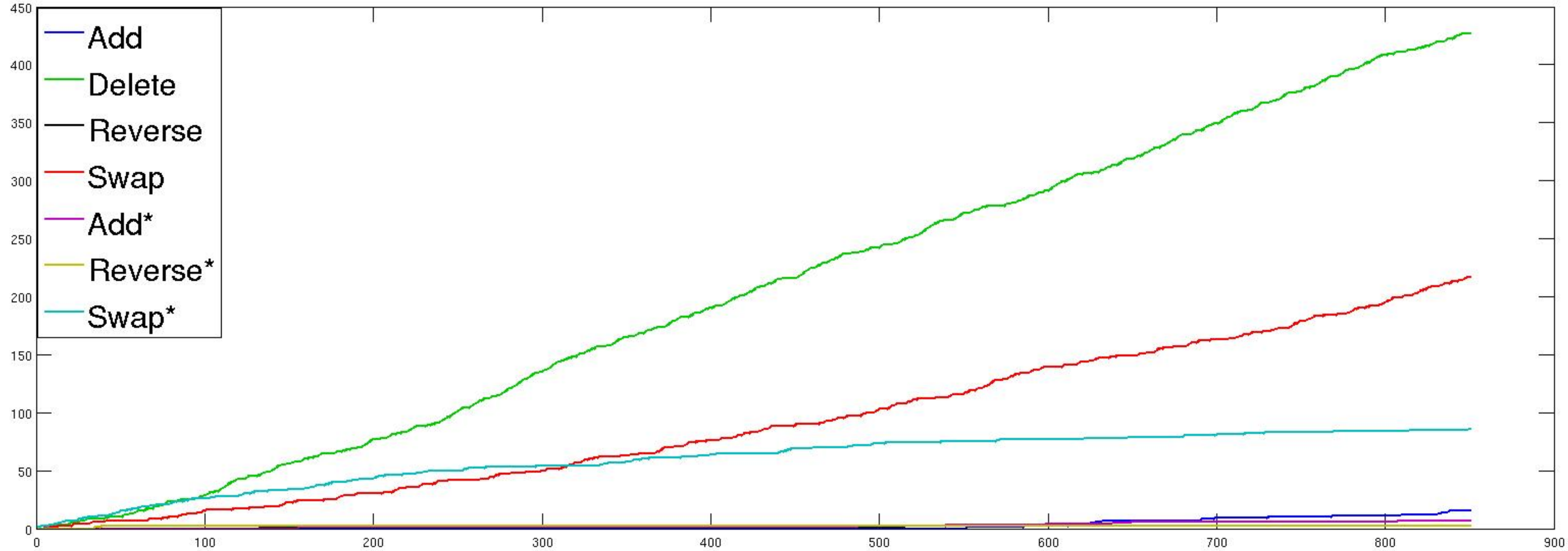
- **1000** runs of hill climbing algorithm
- Initialized with **random networks** (2 parents max)
- **5** operator configurations:
  - Addition + Deletion
  - Addition + Deletion + Reversal
  - Addition + Deletion + Swap
  - Addition + Deletion + Reversal + Swap
  - Addition² + Deletion + Reversal² + Swap² (²:nISC)

	A+D	A+D+R	A+D+S	A+D+R+S	A²+D+R²+S²
BDeu scores ➢ *mean* ➢ *deviation*	-359 580 169.3	-359 430 168.5	**-357 990** **92.9**	-357 850 91.0	**-357 460** **55.2**
Mean time (in seconds)	17.9	27.0	**27.6**	32.3	**149.2**

Results (2/4)

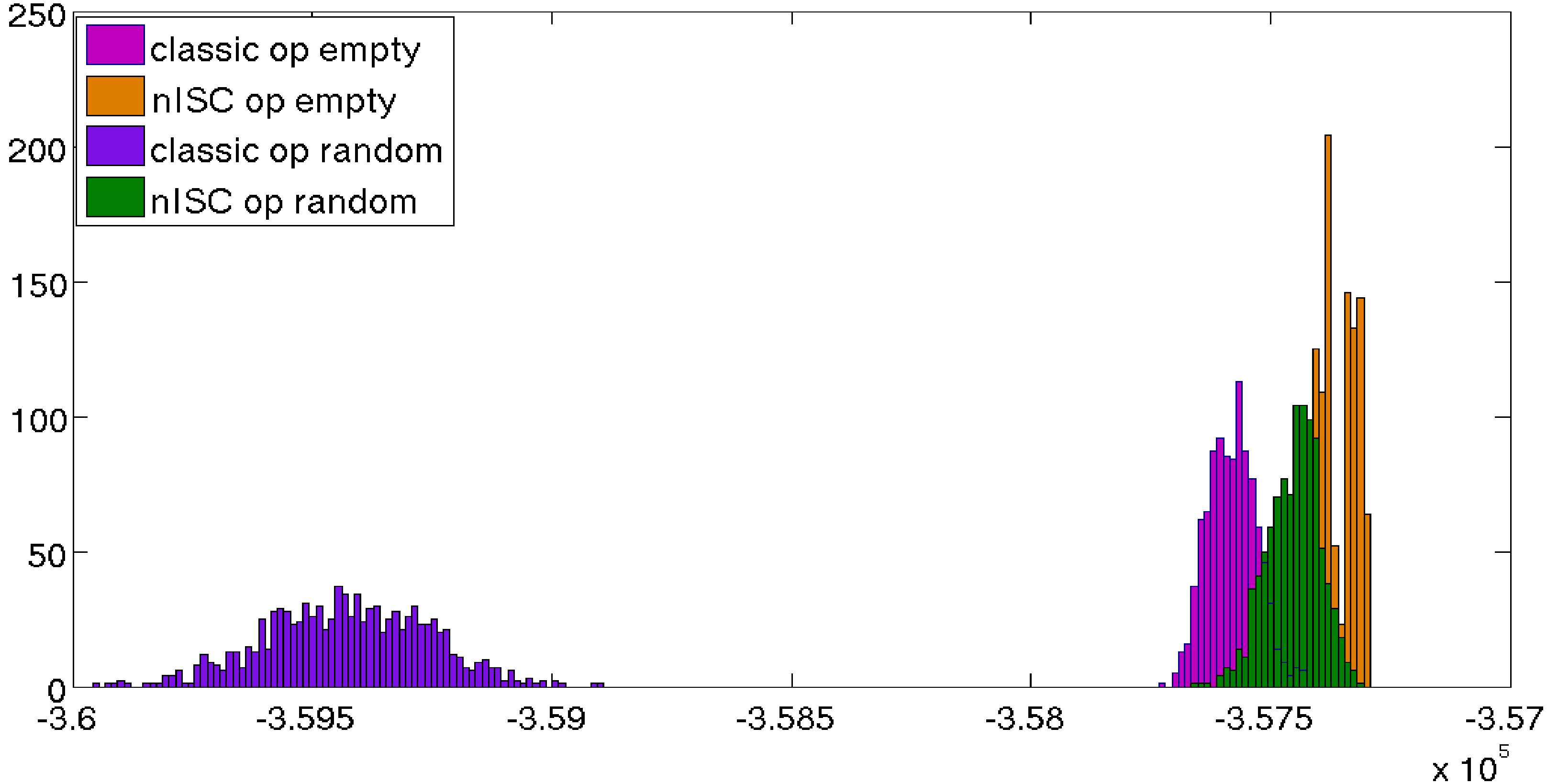
- **1** run of hill climbing algorithm
- Initialized with **random** networks (2 parents max)
- **1** operator configurations: ✗ Addition² + Deletion + Reversal² + Swap² (²:nISC)

Number of applied operators by type during the search



Results (3/4)

- **1000** runs of hill climbing algorithm
- **2** starting configurations:
  - empty network
  - random networks (2 parents max)
- **2** operator configurations:
  - Addition + Deletion + Reversal
  - Addition² + Deletion + Reversal² + Swap² (²:nISC)



Results (4/4)

- **1000** runs of hill climbing algorithm
- Initialized with **random networks** (2 parents max)
- **5** configurations:
  - Addition + Deletion + Reversal
  - Addition + Deletion + Swap
  - Addition² + Deletion + Reversal² + Swap² (²:**nISC**)
  - Addition* + Deletion + Reversal* + Swap* (*:**ISC**)
  - **Tabu search** with Addition + Deletion + Reversal (10 000 operations, tabuu list size :100)

	A+D+R	A+D+S	A²+D+R²+S²	A*+D+R*+S*	Tabu
BDeu scores - *mean* - *deviation*	-359 430 168.5	**-357 990** **92.9**	**-357 460** **55.2**	-357 450 54.5	-359 150 160.4
Mean time (in seconds)	27.0	**27.6**	**149.2**	373.1	291.5

Conclusion & Perspectives

We

- Propose a new Iterative Swap Operator breaking cycles
- Improve BDeu scores of learned networks with this operator
- Compare initial structure effect

TODO list:

- try other meta-heuristics
- tune Tabu parameters
- improve time efficiency of ISC operator

Question time !