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 (\textit{swap} and \textit{swap*})
- Experiments on standard BN benchmarks
- Experiments in genetical genomics (DREAM 2012)
- Conclusion & perspectives
Different levels of regulation

Transcription

Gène A

ARNm

Gène B

ARNm

Gène C

Traduction

Protéine A

(1)

Protéine B

(2)

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

*Bootstrap threshold* = 0.3

© URGV-Loudet
**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 $G$: $P_G(X_i|Pa_i) = \theta_i$

(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" |
|---|---|---|---|
|   |   | 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     |
Directed Acyclic Graph (DAG)

- Conditional probability distribution of $X_i$, given its parents $Pa_i$ in $G$: $P_G(X_i / Pa_i) = \theta_i$

(independence of these local probabilities)

Probability Distribution for the Alarm Node given the events of "Earthquakes" and "Burglaries"

<table>
<thead>
<tr>
<th>Event</th>
<th>Event</th>
<th>$P(A \mid E, B)$</th>
<th>$P(\neg A \mid E, B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>B</td>
<td>0.90</td>
<td>0.10</td>
</tr>
<tr>
<td>E</td>
<td>!B</td>
<td>0.20</td>
<td>0.80</td>
</tr>
<tr>
<td>!E</td>
<td>B</td>
<td>0.90</td>
<td>0.10</td>
</tr>
<tr>
<td>!E</td>
<td>!B</td>
<td>0.01</td>
<td>0.99</td>
</tr>
</tbody>
</table>
### Alarm dataset (37 variables)

<table>
<thead>
<tr>
<th>SAO2</th>
<th>FIO2</th>
<th>PRESS</th>
<th>EXPCO2</th>
<th>MINVOL</th>
<th>MIVOLS</th>
<th>HYPOV</th>
<th>LVFAI</th>
<th>ANAPH</th>
<th>INSUF</th>
<th>VENTMACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>NORMAL</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>NORMAL</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>NORMAL</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>NORMAL</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>NORMAL</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>NORMAL</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>NORMAL</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>NORMAL</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>NORMAL</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>NORMAL</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>NORMAL</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>NORMAL</td>
</tr>
<tr>
<td>HIGH</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>HIGH</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>HIGH</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>HIGH</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>HIGH</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>HIGH</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
<tr>
<td>LOW</td>
<td>NORMAL</td>
<td>HIGH</td>
<td>LOW</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>HIGH</td>
</tr>
</tbody>
</table>
Structure Learning

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 \( Pa_i \)

\[
\Delta_{\text{score}} \; OP_i = f(G') - f(G) = f_{X_i}(G') - f_{X_i}(G)
\]

(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

<table>
<thead>
<tr>
<th>1. Search space</th>
<th>2. Initial structure</th>
<th>3. Neighborhood operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Directed Acyclic Graph</td>
<td>➢ empty structure</td>
<td>➢ addition of an edge</td>
</tr>
<tr>
<td>➢ Partial DAG (PDAG)</td>
<td>➢ random structure</td>
<td>➢ deletion of an edge</td>
</tr>
<tr>
<td></td>
<td>➢ informed structure (MWST, expert...)</td>
<td>➢ reversal of an edge</td>
</tr>
<tr>
<td></td>
<td>➢ variable orders</td>
<td>➢ k look-ahead</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➢ optimal reinsertion</td>
</tr>
</tbody>
</table>

### 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 Markov-equivalent structures

➢ Markov-equivalent structures in Bayesian networks

\[ G_1 \quad G_2 \quad G_3 \]
SGS Algorithm

➢ Sohastic Greedy Search (SGS)

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

➢ Markov-equivalent structures in Bayesian networks

\[ G_1 \sim G_2 \]

\[ G_1 \xrightarrow{\sim} G_2 \]

\[ G_1 \sim G_3 \]

\[ G_1 \xrightarrow{\sim} G_2 \]
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 Markov-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)

Example:

\[ \Delta_{\text{score \ Add}} (G_2, G_3) > \Delta_{\text{score \ Add}} (G_1, G_3) > 0 \]
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)

Example:

Current situation

\[
\Delta_{\text{score}} \text{Add}(G_1, G_3) > 0
\]

Target situation

\[
\Delta_{\text{score}} \text{Add}(G_2, G_3) > \Delta_{\text{score}} \text{Add}(G_1, G_3) > 0
\]
Swap Operator

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

**Example:**

Current situation

\[
\Delta_{\text{score}} \text{Add}(G_1, G_3) > 0, \quad \Delta_{\text{score}} \text{Add}(G_2, G_3) \geq 0
\]

Target situation

\[
\Delta_{\text{score}} \text{Add}(G_1, G_3) \geq 0, \quad \Delta_{\text{score}} \text{Add}(G_2, G_3) > 0
\]

\[
\text{Swap}(G_1, G_3, G_2) \quad \Delta_{\text{score}} \text{Add}(G_2, G_3) - \Delta_{\text{score}} \text{Add}(G_1, G_3) > 0
\]

→ escape from some local maxima
Swap* Operator

\[ \text{Swap}(G_2, G_3, G_7)? \]

**Current situation**

\[ \Delta_{\text{score Add}} (G_7, G_3|G_1) > \Delta_{\text{score Add}} (G_2, G_3|G_1) > 0 \]
Swap^* Operator

\[
\text{Swap}(G_2, G_3, G_7) \rightarrow \text{Cycle}\{G_3, G_4, G_6, G_7\}
\]

Current situation
\[
\Delta_{\text{score Add}}(G_7, G_3|G_1) > \Delta_{\text{score Add}}(G_2, G_3|G_1) > 0
\]

Objective: delete the cycles
**Swap** Operator

Swap \((G_2, G_3, G_7)\)? \(\rightarrow\) 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** \(\Delta_{score\ Add}\)
Current situation

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

Objective: delete the cycles

Step 1. Try to delete the edge minimizing \( \Delta_{\text{score} \text{Add}} \)

Improving score? Yes
Swap$^*$ Operator

\[ \text{Swap}(G_2, G_3, G_7) ? \rightarrow \text{Cycle}(G_3, G_4, G_6, G_7) \]

**Current situation**
\[ \Delta_{\text{score Add}}(G_7, G_3|G_1) > \Delta_{\text{score Add}}(G_2, G_3|G_1) > 0 \]

Objective: delete the cycles

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

Improve score?
Yes $\rightarrow$ Acyclic?

- Yes $\rightarrow$ OK!
- No $\rightarrow$ Return to Step 1
\textbf{Swap$^*$ Operator}

\[ \text{Swap}(G_2, G_3, G_7)? \rightarrow \text{Cycle}\{G_3, G_4, G_6, G_7\} \]

\textbf{Current situation}

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

Objective: delete the cycles

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

Improving score?

Yes $\rightarrow$ Acyclic?

\[ \begin{align*}
\text{Yes} & \rightarrow \text{OK!} \\
\text{No} & \rightarrow \text{Return to Step 1}
\end{align*} \]

No $\rightarrow$ Continue to Step 2

Step 2. Try to swap this edge
**Swap Operator**

Current situation

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

Objective: delete the cycles

Step 1. Try to delete the edge minimizing \( \Delta_{\text{score Add}} \)

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

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 \( \Delta_{\text{score Add}} \)

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:

<table>
<thead>
<tr>
<th></th>
<th>Alarm</th>
<th>Insurance</th>
<th>Hailfinder</th>
<th>Pigs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>37</td>
<td>27</td>
<td>56</td>
<td>441</td>
</tr>
<tr>
<td>Edges</td>
<td>46</td>
<td>52</td>
<td>66</td>
<td>592</td>
</tr>
<tr>
<td>In-degree</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

➢ 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
  \[ \Delta \text{Add}(\text{Parent}, \text{Target}) > 0 \]
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 $\text{SGS}^3$ ($r=1$) with 500 samples

➢ $\text{SGS}^3$ Initialized with *empty* and *random network* (2 parents max)

---

**empty network**

![Graph](empty_network_graph.png)

**random network**

![Graph](random_network_graph.png)
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

<table>
<thead>
<tr>
<th>Wilcoxon test</th>
<th>Alarm 500</th>
<th>Alarm 5 000</th>
<th>Insurance 500</th>
<th>Insurance 5 000</th>
<th>Hailfinder 500</th>
<th>Hailfinder 5 000</th>
<th>Pigs 500</th>
<th>Pigs 5 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGS³ vs GES</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>SGS³ vs LAGD</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>~</td>
<td>+</td>
<td>n/a</td>
</tr>
<tr>
<td>LAGD vs GES</td>
<td>+</td>
<td>~</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>n/a</td>
</tr>
</tbody>
</table>
Comparison of **Hamming distances** for SGS³, LAGD and GES

Hamming distance = False Positive + False Negative

<table>
<thead>
<tr>
<th></th>
<th>Alarm</th>
<th>Insurance</th>
<th>Hailfinder</th>
<th>Pigs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>500</td>
<td>5 000</td>
<td>500</td>
<td>5 000</td>
</tr>
<tr>
<td>SGS³</td>
<td>11*</td>
<td>8</td>
<td><strong>24</strong>*</td>
<td><strong>10</strong>*</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>5 000</td>
<td>500</td>
<td>5 000</td>
</tr>
<tr>
<td>LAGD</td>
<td>15</td>
<td>10</td>
<td><strong>24</strong>*</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>5 000</td>
<td>500</td>
<td>5 000</td>
</tr>
<tr>
<td>GES</td>
<td>11*</td>
<td>6*</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>5 000</td>
<td>500</td>
<td>5 000</td>
</tr>
</tbody>
</table>

* best result
Comparison of **Hamming distances** for **SGS³**, **LAGD** and **GES**

Hamming distance = False Positive + False Negative

<table>
<thead>
<tr>
<th></th>
<th>Alarm</th>
<th>Insurance</th>
<th>Hailfinder</th>
<th>Pigs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>500</td>
<td>5 000</td>
<td>500</td>
<td>5 000</td>
</tr>
<tr>
<td>SGS³</td>
<td>11*</td>
<td>8</td>
<td>24*</td>
<td>10*</td>
</tr>
<tr>
<td>LAGD</td>
<td>15</td>
<td>10</td>
<td>24*</td>
<td>16</td>
</tr>
<tr>
<td>GES</td>
<td>11*</td>
<td>6*</td>
<td>25</td>
<td>15</td>
</tr>
</tbody>
</table>

* best result

Pigs network

![Diagram of the Pigs network](image)
Results (4/4)

- Comparison of Hamming distances for SGS, LAGD and GES

Hamming distance = False Positive + False Negative

<table>
<thead>
<tr>
<th></th>
<th>Alarm</th>
<th>Insurance</th>
<th>Hailfinder</th>
<th>Pigs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>500</td>
<td>5 000</td>
<td>500</td>
<td>5 000</td>
</tr>
<tr>
<td>SGS³</td>
<td>11*</td>
<td>8</td>
<td>24*</td>
<td>10*</td>
</tr>
<tr>
<td>LAGD</td>
<td>15</td>
<td>10</td>
<td>24*</td>
<td>16</td>
</tr>
<tr>
<td>GES</td>
<td>11*</td>
<td>6*</td>
<td>25</td>
<td>15</td>
</tr>
</tbody>
</table>

* best result

Pigs network

```
G_1 -- G_2 -- G_3 -- G_4
    |     |     |
    G_5 -- G_6 -- G_7
    |     |     |
    G_8 -- G_9
```

SGS

```
G_1 -- G_2 -- G_3 -- G_4
    |     |     |
    G_5 -- G_6 -- G_7
    |     |     |
    G_8 -- G_9
```
Comparison of **Hamming distances** for **SGS**, **LAGD** and **GES**

Hamming distance = False Positive + False Negative

<table>
<thead>
<tr>
<th></th>
<th>Alarm 500</th>
<th>Alarm 5 000</th>
<th>Insurance 500</th>
<th>Insurance 5 000</th>
<th>Hailfinder 500</th>
<th>Hailfinder 5 000</th>
<th>Pigs 500</th>
<th>Pigs 5 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGS³</td>
<td>11*</td>
<td>8</td>
<td>24*</td>
<td>10*</td>
<td>41</td>
<td>29*</td>
<td>32</td>
<td>41</td>
</tr>
<tr>
<td>LAGD</td>
<td>15</td>
<td>10</td>
<td>24*</td>
<td>16</td>
<td>47</td>
<td>39</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>GES</td>
<td>11*</td>
<td>6*</td>
<td>25</td>
<td>15</td>
<td>39*</td>
<td>33</td>
<td>9*</td>
<td>0*</td>
</tr>
</tbody>
</table>

* best result

Pigs network

![Diagram of Pigs network](image-url)
Results (4/4)

- Comparison of Hamming distances for SGS, LAGD and GES

Hamming distance = False Positive + False Negative

<table>
<thead>
<tr>
<th></th>
<th>Alarm 500</th>
<th>Alarm 5 000</th>
<th>Insurance 500</th>
<th>Insurance 5 000</th>
<th>Hailfinder 500</th>
<th>Hailfinder 5 000</th>
<th>Pigs 500</th>
<th>Pigs 5 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGS³</td>
<td>9*</td>
<td>4*</td>
<td>24*</td>
<td>9*</td>
<td>40</td>
<td>26*</td>
<td>1*</td>
<td>2</td>
</tr>
<tr>
<td>LAGD</td>
<td>15</td>
<td>10</td>
<td>24*</td>
<td>16</td>
<td>47</td>
<td>39</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>GES</td>
<td>11</td>
<td>6</td>
<td>25</td>
<td>15</td>
<td>39*</td>
<td>33</td>
<td>9</td>
<td>0*</td>
</tr>
</tbody>
</table>

* 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 \( (p=100, 1000, 5000, e \sim 6p) \) x 8 configurations

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Marker Distance</th>
<th>Biological Variance</th>
<th>Heritability</th>
<th>Population Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N(5,1)</td>
<td>N(1,0.1)</td>
<td>High</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>N(5,1)</td>
<td>N(1,0.1)</td>
<td>High</td>
<td>900</td>
</tr>
<tr>
<td>3</td>
<td>N(5,1)</td>
<td>N(1,0.25)</td>
<td>Low</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>N(5,1)</td>
<td>N(1,0.25)</td>
<td>Low</td>
<td>900</td>
</tr>
<tr>
<td>5</td>
<td>N(1,0.1)</td>
<td>N(1,0.1)</td>
<td>High</td>
<td>300</td>
</tr>
<tr>
<td>6</td>
<td>N(1,0.1)</td>
<td>N(1,0.1)</td>
<td>High</td>
<td>900</td>
</tr>
<tr>
<td>7</td>
<td>N(1,0.1)</td>
<td>N(1,0.25)</td>
<td>Low</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td>N(1,0.1)</td>
<td>N(1,0.25)</td>
<td>Low</td>
<td>900</td>
</tr>
</tbody>
</table>

SysGenSIM (de la Fuente et al, Bioinformatics 2011)
$SGS^3 \ (BDeu \ \alpha=1)$ with 10 restarts and 100 bootstraps

Network 1000-2

![Graph showing precision vs. recall with various configurations labeled from #33 to #40 and different colors representing each configuration.](image-url)
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

- 10 alpha x 100 bootstrap
- alpha=1
- 10 alpha (seuil bootstrap 0.8)
- 10 alpha (aucun bootstrap)

Precision vs. Recall graph with various line styles and colors representing the different configurations.
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 benches

TODO list:

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