



# A Novel Algorithm for Scalable and Accurate Bayesian Network Learning

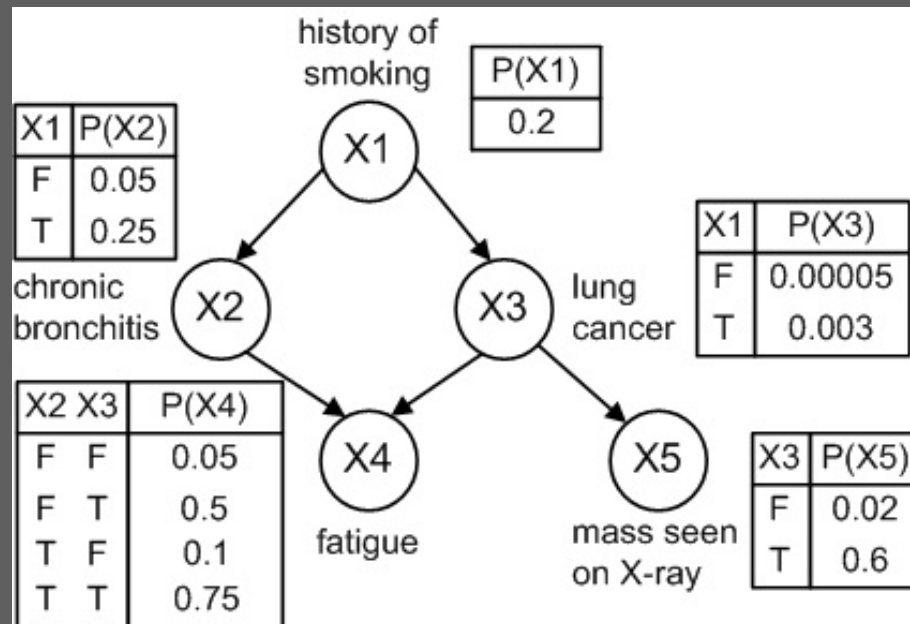
Laura E. Brown M.S., Ioannis Tsamardinos Ph.D.,  
Constantin Aliferis M.D. Ph.D

Department of Biomedical Informatics  
Vanderbilt University

MEDINFO 2004 Sept 9, 2004



# What are Bayesian Networks?



- A Bayesian Network (BN),  $B = (G, J)$ 
  - $G$ , a directed acyclic graph (DAG)
  - $J$ , set of parameters to quantify the probability distribution

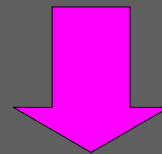
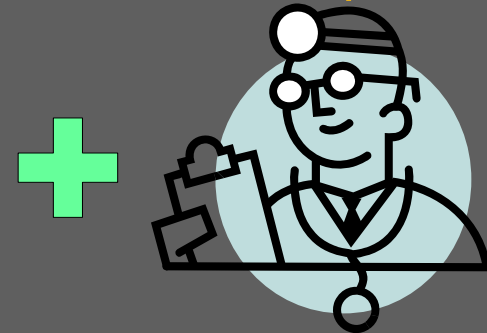
Figure from Glymour, C. and Cooper, G.F. Computation, Causation, and Discovery, 2000.

# Learning Bayesian Networks

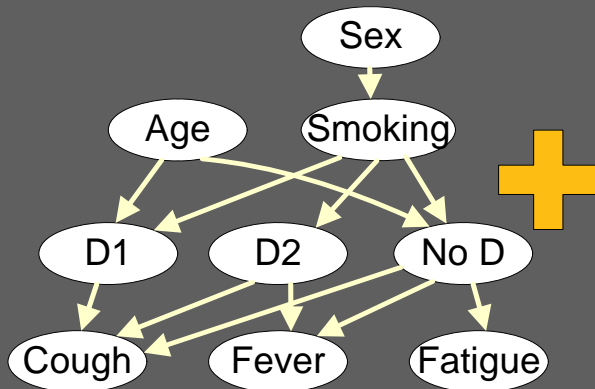
## Given Data

| Patient | Age | Sex | Smoking | Fever | Cough | Fatigue | D1 | D2 | No D |
|---------|-----|-----|---------|-------|-------|---------|----|----|------|
| 1       | 50  | M   | Y       | Y     | N     | Y       | Y  | N  | N    |
| 2       | 72  | F   | N       | N     | Y     | N       | N  | Y  | N    |
| ...     |     |     |         |       |       |         |    |    |      |
| 1000    | 24  | F   | Y       | N     | N     | N       | N  | N  | Y    |

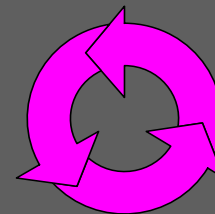
## Domain Expert



Construct Network  
by Hand



$P(\text{Sex}=\text{M}) = 0.5$   
 $P(\text{Sex}=\text{F}) = 0.5$   
 $P(\text{Smoking}=\text{Y} \mid \text{Sex}=\text{M}) = 0.3$   
 $P(\text{Smoking}=\text{Y} \mid \text{Sex}=\text{F}) = 0.4$   
 ...  
 $P(\text{Fatigue}=\text{Y} \mid \text{No D}=\text{Y}) = 0.2$   
 $P(\text{Fatigue}=\text{Y} \mid \text{No D}=\text{N}) = 0.25$



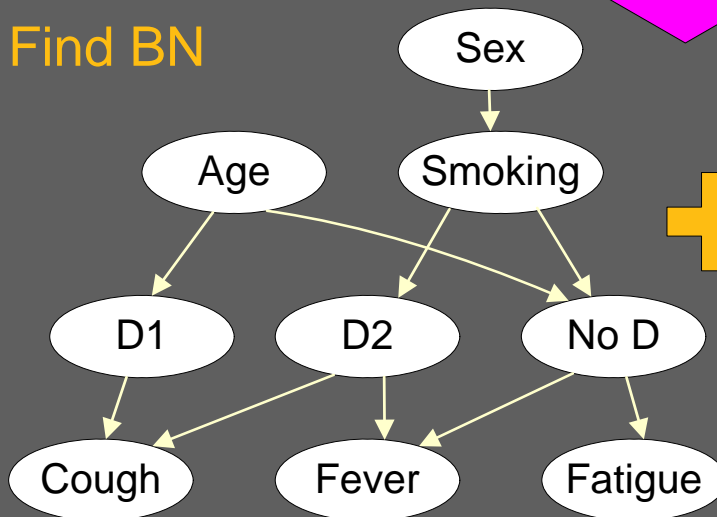
Assess  
Network &  
Modify  
Network

# Bayesian Network Learning Algorithms

## Given Data

| Patient | Age | Sex | Smoking | Fever | Cough | Fatigue | D1 | D2 | No D |
|---------|-----|-----|---------|-------|-------|---------|----|----|------|
| 1       | 50  | M   | Y       | Y     | N     | Y       | Y  | N  | N    |
| 2       | 72  | F   | N       | N     | Y     | N       | N  | Y  | N    |
| ...     |     |     |         |       |       |         |    |    |      |
| 1000    | 24  | F   | Y       | N     | N     | N       | N  | N  | Y    |

## Find BN



## Apply BN Learning Algorithm

$$P(\text{Sex}=\text{M}) = 0.5$$

$$P(\text{Sex}=\text{F}) = 0.5$$

$$P(\text{Smoking}=\text{Y} \mid \text{Sex}=\text{M}) = 0.3$$

$$P(\text{Smoking}=\text{Y} \mid \text{Sex}=\text{F}) = 0.4$$

...

$$P(\text{Fatigue}=\text{Y} \mid \text{No D}=\text{Y}) = 0.1$$

$$P(\text{Fatigue}=\text{Y} \mid \text{No D}=\text{N}) = 0.3$$

DAG + Probabilities to quantify distribution

# Bayesian Network Applications

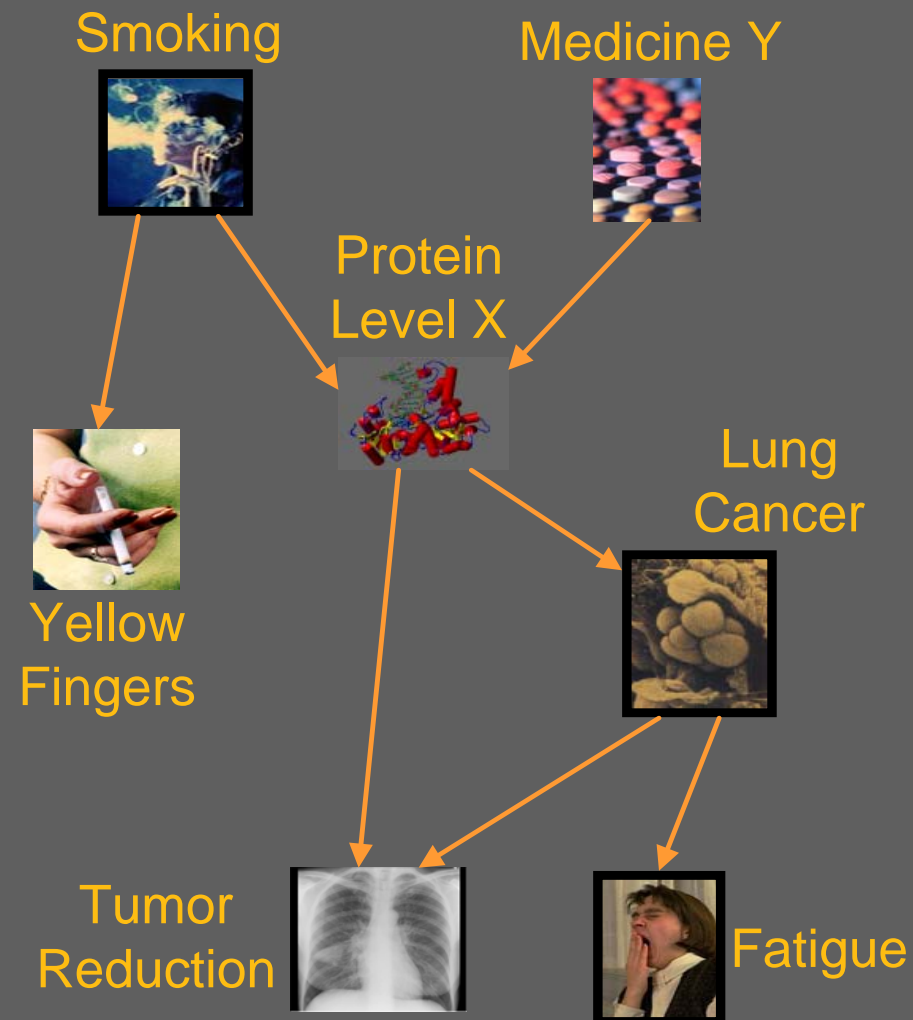
---

- BNs used for:
  - Probabilistic Inference
    - Decision Support Systems
  - Inferring causal relations
  - Variable selection
- BNs extensions
  - Decision theory

# BNs in Biomedicine: Probabilistic Inference

## Decision Support Systems

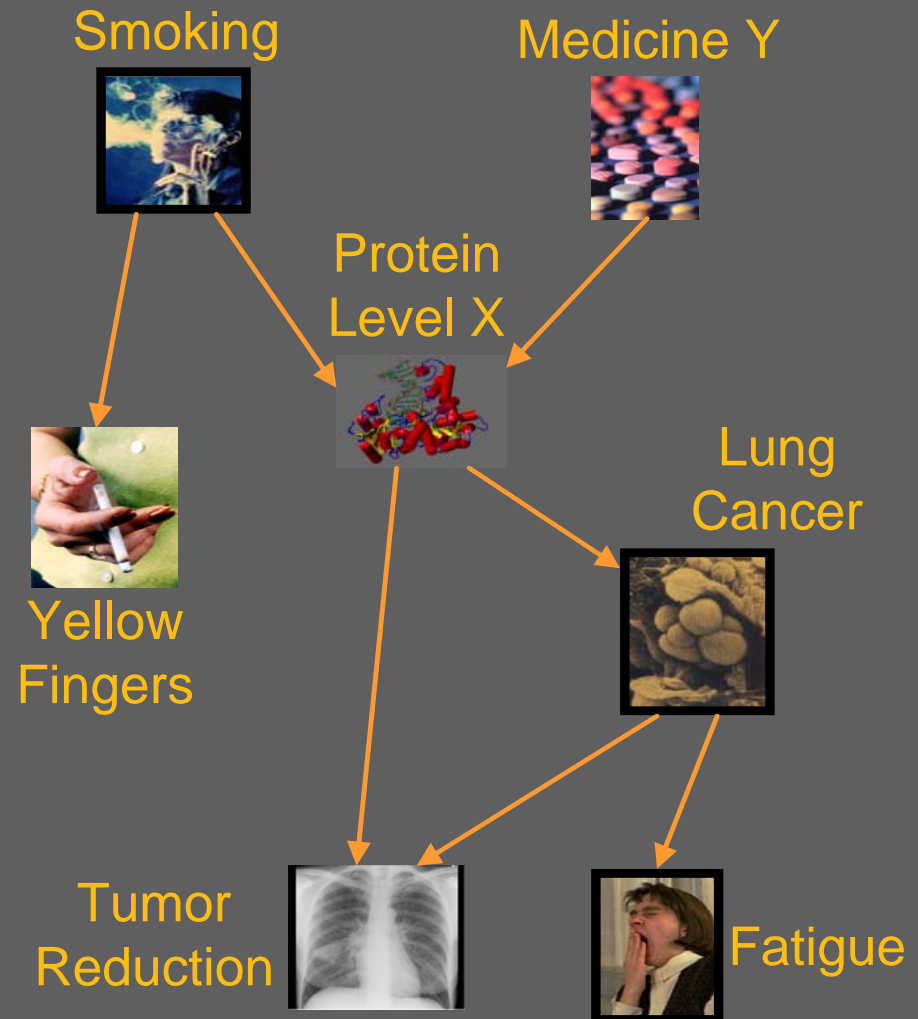
- Medical Disease Diagnosis
  - $P(\text{Disease} \mid \text{Findings})$
- Prognostic Reasoning
  - $P(\text{Outcome} \mid \text{Treatment and Disease})$



# BNs in Biomedicine: Probabilistic Inference

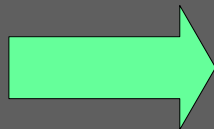
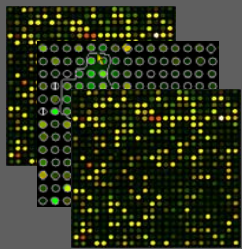
- Arbitrary Probabilities can be calculated

- $P(\text{Yellow Fingers} \mid \text{Lung Cancer})$
- $P(\text{Lung Cancer} \mid \text{Yellow Fingers, Medicine Y})$



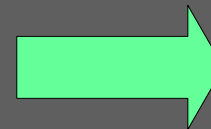
# BNs in Biomedicine: Variable Interaction Discovery

- Bioinformatics tool for gene regulatory pathways

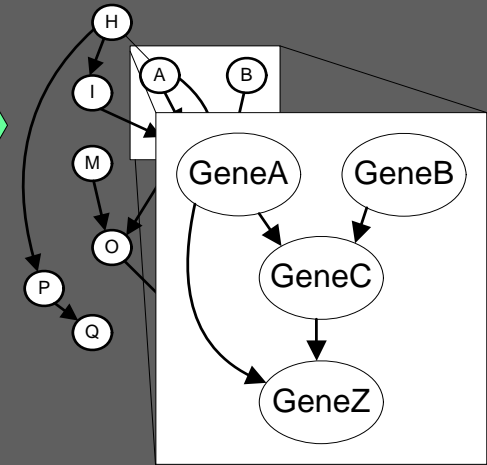


Transform  
Microarray  
Results to  
Data Set

| GeneA | GeneB | ... | GeneZ |
|-------|-------|-----|-------|
| 0     | 0     |     | 2     |
| 1     | 2     |     | 1     |
| 2     | 1     |     | 2     |
| 0     | 1     |     | 0     |
| ...   |       |     |       |
| 0     | 1     |     | 0     |



Apply BN  
Learning  
Algorithm





# BNs in Biomedicine: Data

---

- High dimensional data sets
  - Clinical
  - Biomedical
- Creation of BNs by hand is not feasible
- BN Learning Algorithms must scale

# The Problem

---

- Design of novel algorithm for Learning the Structure of a Bayesian Network (BN)
- Aims for improvements in:
  - Computational Efficiency
  - Scalability
  - Quality

# Bayesian Network Learning

- *NP-hard* Problem<sup>1</sup>
- Classical Approaches to problem
  - Constraint-based approach
    - PC<sup>2</sup>, TPDA<sup>3</sup>
  - Search-and-score approach
    - Greedy Search (GS), Sparse Candidate (SC)<sup>4</sup>

1 Chickering, DM and Geiger, D and Heckerman, D. Learning Bayesian Networks is NP-Hard. Tech Report 1994, Chickering, DM and Meek, C and Heckerman, D. Large-Sample Learning of Bayesian Networks is NP-hard. *UAI* 2003.

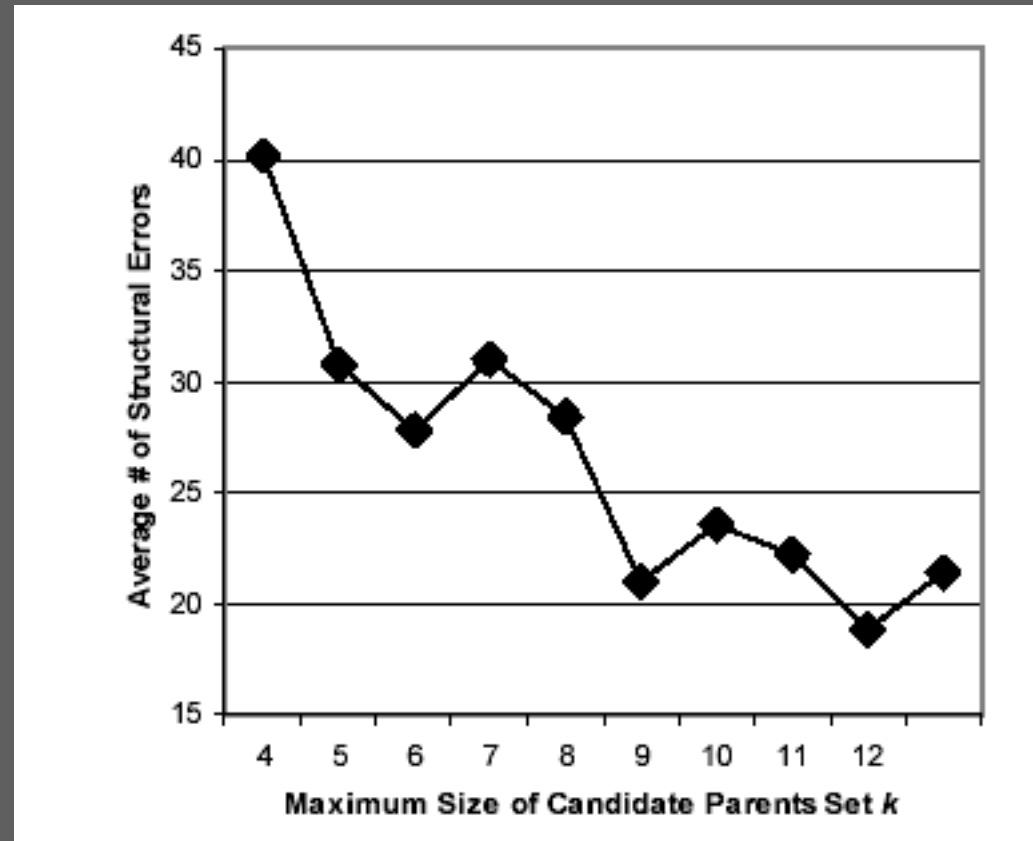
2 Spirtes P, Glymour C, Scheines R. *Causation, Prediction, and Search*. 2000.

3 Jie et al. Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence* 2002.

4 Friedman N, Nachman I, Pe'er D. Learning Bayesian Network Structure from Massive Datasets: The “Sparse Candidate” Algorithm. *UAI* 1999.

# Development of Novel Algorithm

- SC employs a constrained search
  - Method of choosing candidate sets flawed
    - User defined parameter,  $k$ , maximum size of candidate sets
    - Assumes uniform sparseness



# Development of Novel Algorithm

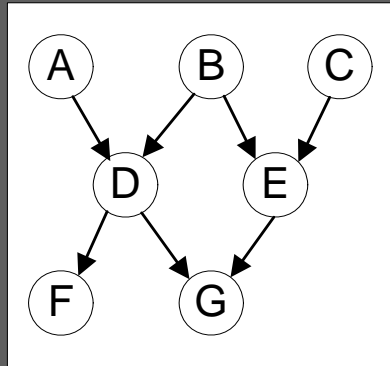
---

- New algorithm Max-Min Hill Climbing (MMHC)
  - Incorporates SC idea of constraining the search
  - Uses local causal discovery algorithm, Max-Min Parents and Children (MMPC), to find the candidate sets
    - Two phase algorithm, to find *ParentsChildren(target)*

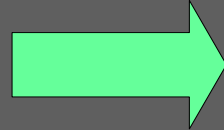
# Max-Min Hill Climbing (MMHC)

- Combines constraint-based and search-and-score methods
1. For each variable  $X$ ,  
    set Candidate Set( $X$ ) = MMPC( $X$ , Data)
  2. Starting from empty BN  
    Perform Greedy Hill Climbing Search  
    Use operators Add\_Edge, Delete\_Edge, and Reverse\_Edge.  
    Restricted Add\_Edge to only add  $X \rightarrow Y$   
        if  $X \in$  Candidate Set( $Y$ )

# Experimental Evaluation



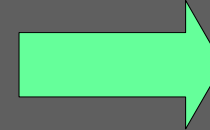
Gold Standard Network



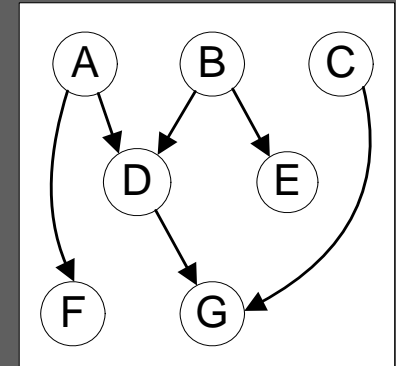
Sample Network to Produce Data Sets

| A   | B | C | D | E | F | G |
|-----|---|---|---|---|---|---|
| 0   | 1 | 0 | 0 | 3 | 1 | 2 |
| 1   | 0 | 0 | 2 | 0 | 1 | 1 |
| 0   | 1 | 0 | 1 | 2 | 0 | 2 |
| 0   | 1 | 0 | 1 | 1 | 0 | 0 |
| ... |   |   |   |   |   |   |
| 0   | 1 | 0 | 1 | 1 | 0 | 0 |

Data Set



Apply Algorithms to find Learned Networks



Learned Network

## ■ Metrics

- Running Time
- Structural Hamming Distance – compares gold-standard and learned network

# Experimental Evaluation

---

- Algorithms: MMHC, SC (k=5, 10), PC, TPDA, and Greedy Search
- Networks: Alarm, Alarm3, Alarm5, Munin1, Child, Gene, Alarm10, Hailfinder, Insurance, Barley, Mildew
- Sample Sizes: 500, 1000, 5000
- Data Sets: 5 data sets sampled from each network for each sample size



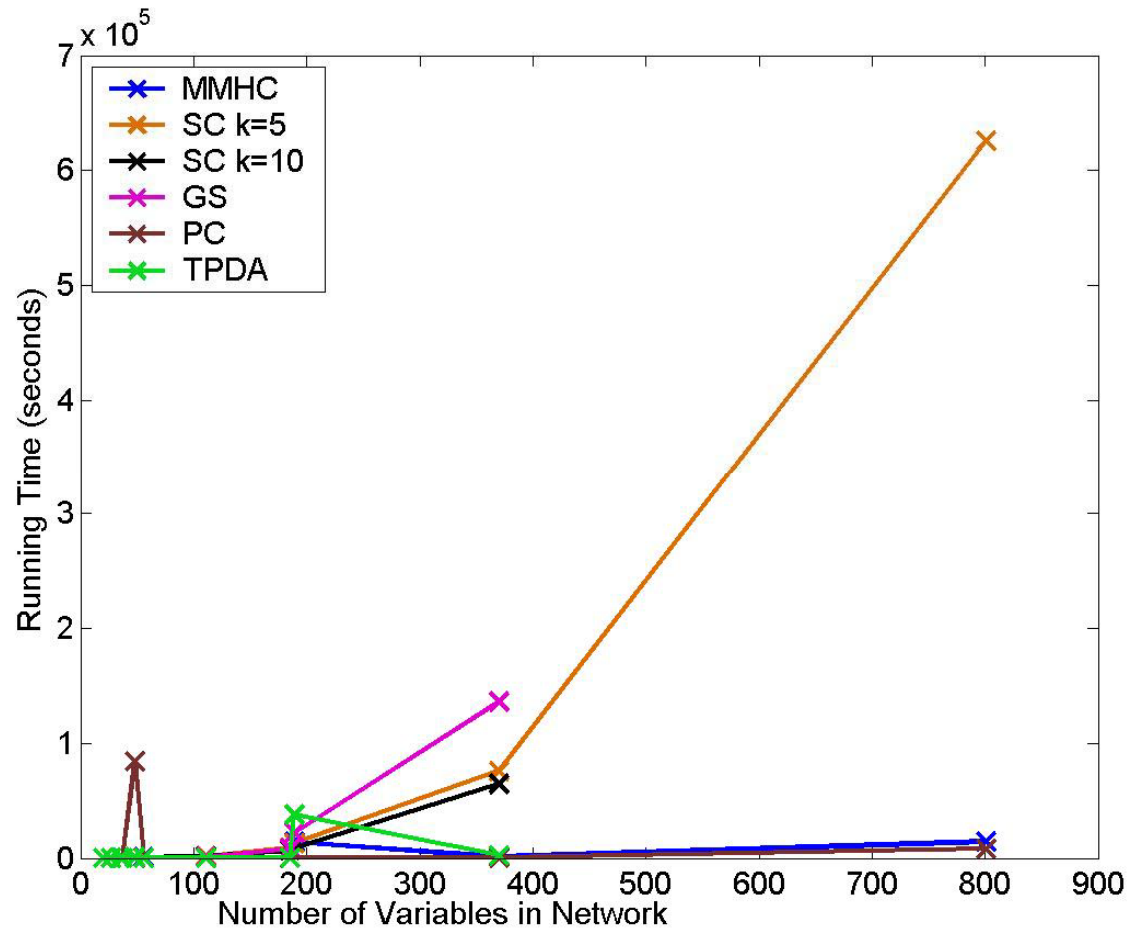
# Results: Computational Efficiency

- MMHC on Child = 14 seconds, on Gene = 17K seconds
- SC on Child = 9 seconds, on Gene = +600K seconds

## Average Normalized Running Time

| Algorithms | Sample Size |       |       | Average Across<br>Sample Sizes |
|------------|-------------|-------|-------|--------------------------------|
|            | 500         | 1000  | 5000  |                                |
| MMHC       | 1.00        | 1.00  | 1.00  | 1.00                           |
| SC k=5     | 9.50        | 10.80 | 17.67 | 12.66                          |
| SC k=10    | 7.83        | 13.81 | 18.99 | 13.54                          |
| GS         | 8.69        | 17.62 | 19.23 | 15.18                          |
| PC         | 43.55       | 19.36 | 98.63 | 53.85                          |
| TPDA       | 53.06       | 12.35 | 2.25  | 22.55                          |

# Results: Scalability



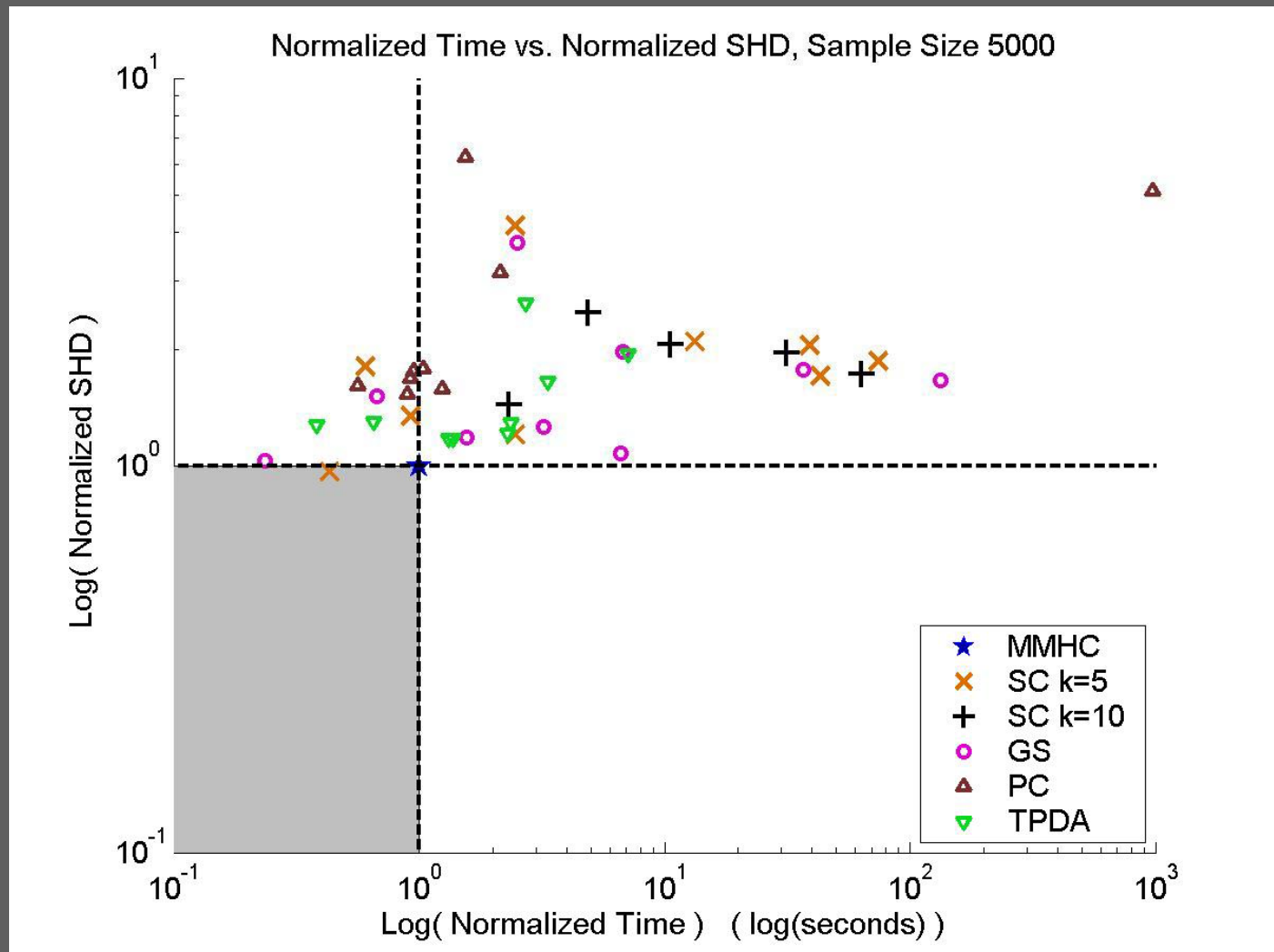
# Results: Quality

- Measure of Quality: Structural Hamming Distance

## Average Normalized Structural Hamming Distance

| Algorithms | Sample Size |      |      | Average Across<br>Sample Size |
|------------|-------------|------|------|-------------------------------|
|            | 500         | 1000 | 5000 |                               |
| MMHC       | 1.00        | 1.00 | 1.00 | 1.00                          |
| SC k=5     | 1.26        | 1.41 | 1.91 | 1.53                          |
| SC k=10    | 1.54        | 1.61 | 1.93 | 1.70                          |
| GS         | 1.09        | 1.18 | 1.69 | 1.32                          |
| PC         | 3.68        | 2.94 | 2.72 | 3.12                          |
| TPDA       | 2.91        | 2.52 | 1.51 | 2.31                          |

# Results: Time vs. Quality



# Evaluation Conclusions

---

- Computational Efficiency: MMHC outperforms on average
- Scalability: MMHC is able to scale as well or better
- Learned Network Quality: MMHC outperforms on average
- Time and Quality: MMHC is faster or finds high quality results

# Future Work

---

- Compare to other algorithms (GES, Optimal Reinsertion, BENEDICT)
- Apply to real-world data with expert analysis of results
- Creation of a software tool to make this algorithm available for researchers.

# Acknowledgements

---

- NLM & NIH for funding
- Discovery Systems Laboratory Members
  - Alexander Statnikov
- Web – Site for more information
- <http://discover1.mc.vanderbilt.edu/discover/public/>
- Questions???