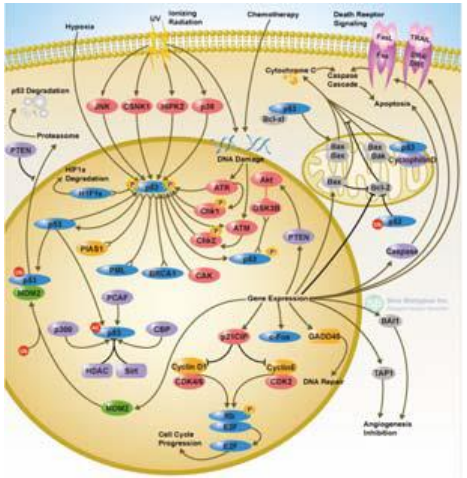
Gene Regulatory Network Inference using Bayesian Graphic Model with Gene Expression Data in Cancer Cells

Tanjin Xu, Jun He, Xingyi Li

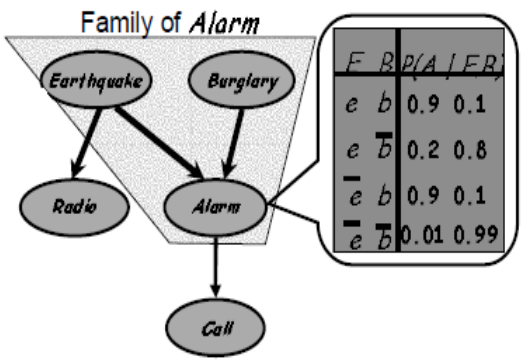
MAR 20, 2015



Outline

- Introduction
- Related work
- Methodology
- Result
- Discussion
- Conclusion

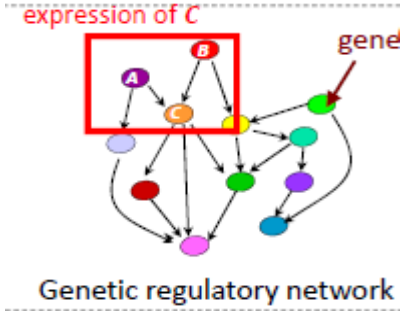
- Bayesian Network
  - Compact representation of probability distribution via conditional independence

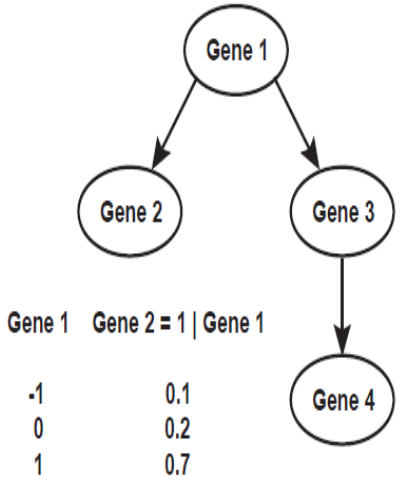


  - Directed Acyclic Graph (DAG)
    - Nodes – random variables
    - Edges – direct influence
  - Define a unique distribution in a factored from

$$P(B,E,A,C,R) = P(B)P(E)P(A \mid B,E)P(R \mid E)P(C \mid A)$$

- Gene Regulatory Network (GRN)

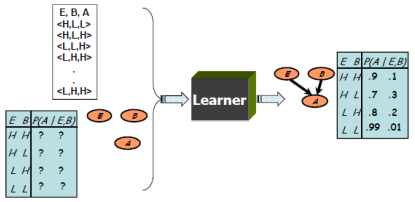




  - Nodes – genes
  - Edges – gene interactions (regulation relationship)

Oregon State University

Structure Learning



- Why learning GRN?
  - The genes' expression are all messed up in cancer and novel transcipts might be generated (new edges introduced)
  - The structure is undermined
  - Locate the significant genes (nodes) and interactions (edges) for target therapy

- Why Bayesian Network?
  - Conditional independencies & graphical language capture structure of many real-world distributions
  - Graph structure provides much insight into domain – "knowledge discovery"
  - Capable of combining domain knowledge with data
  - Dealing with missing data & hidden variables
    - *In this project, assume data is complete*

OSU
Oregon State
UNIVERSITY

Structure Learning (cont.)

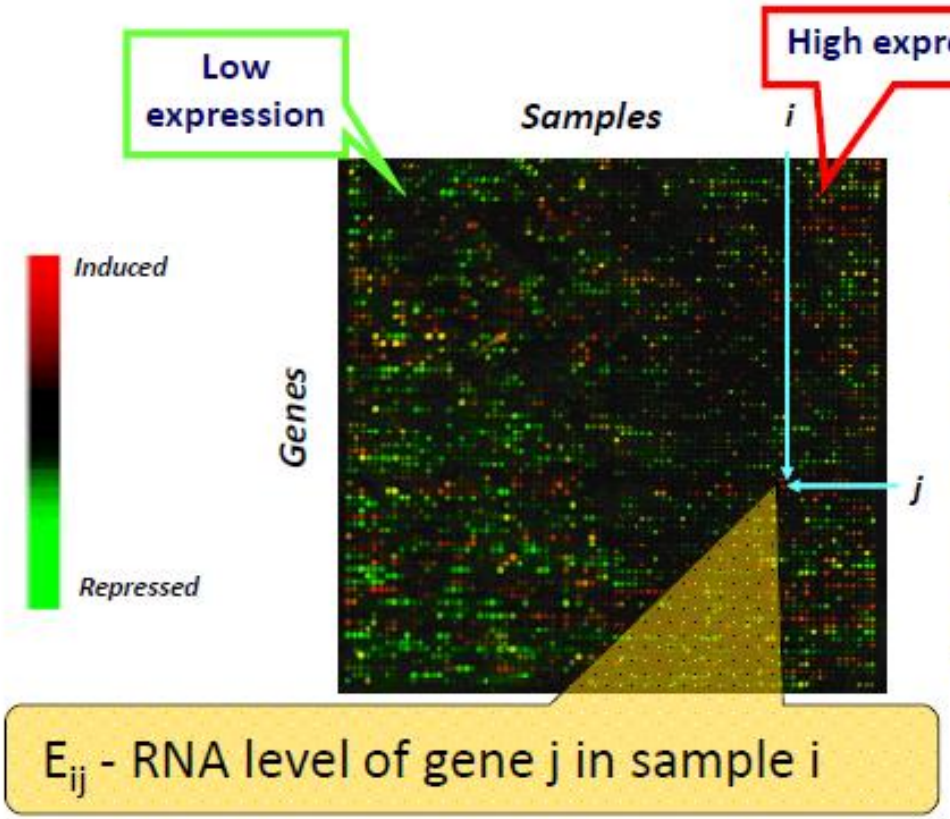
- Constraint Based
  - Test independencies & add edges according to the tests
  - Cons
    - Independence tests are less reliable on small samples
    - One incorrect independence test might propagate far

- Search and Score
  - Define a selection criterion that measures goodness of a model
  - Search in the space of all models (or orders)
  - Score functions: **BDe**, BDeu, BIC, etc.
  - Search algorithms: K2 (ordered), **Hill-climbing**, Simulated annealing…

- Mix models
  - Test for almost all independencies
  - Search and score

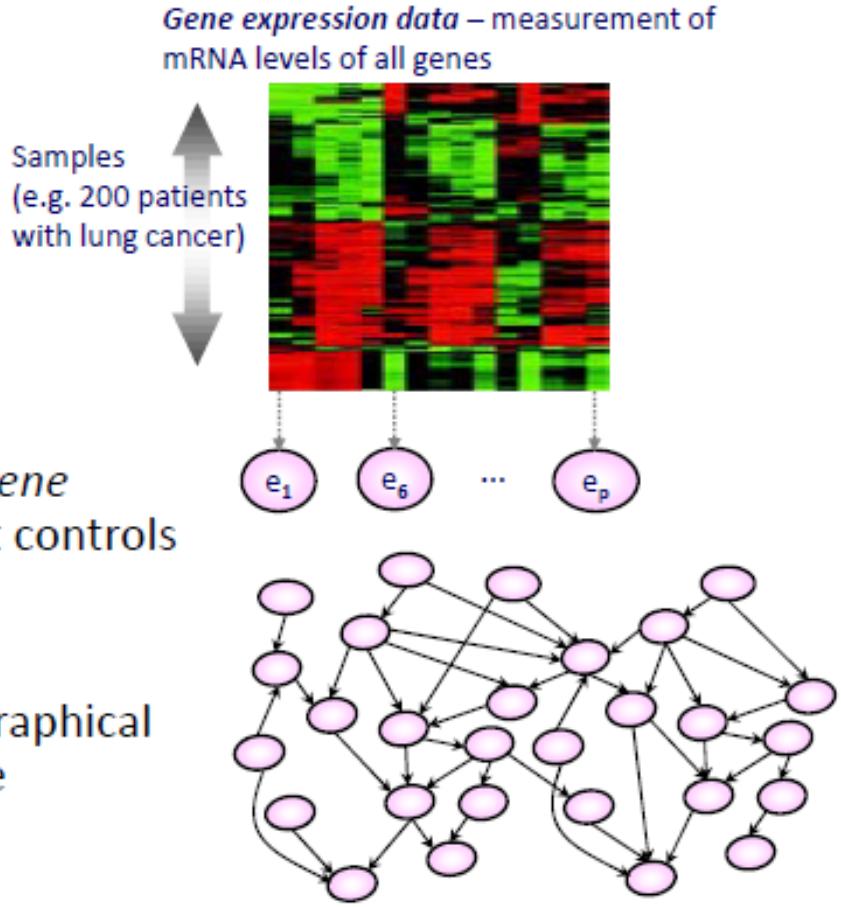
OSU
Oregon State
UNIVERSITY

Gene expression data



Learning gene regulatory networks

- **Input:**



- **Goal:** Reconstruct the *gene regulatory network* that controls gene expression

- **Method:** Probabilistic graphical models to represent the regulatory network

Related work

- BNs were first applied to gene expression studies in the analysis of the yeast cell cycle by *Friedman, et al.,* 2000
  - Applied both multinomial model and Gaussian model using heuristic search
  - *Only normal cells*
- Seeded Bayesian Networks by *Quackenbush, et al.,* 2008
  - Almost the same pipeline as Friedman
  - Combined different sources of gene interactions as priors (seeds)
  - Expression data from microarray in cancer celllines – *not real cancer cells*
  - *Only 40 genes are analyzed*
- Learning a Markov Logic Network for supervised gene regulatory network inference*, Brouard, et al.,* 2013
  - Build a binary classifier based on existing knowledge about known gene interactions
  - *Not appropriate for cancer gene expression analysis*

Outline

- Introduction
- Related work
- ***Methodology***
- Result
- Discussion
- Conclusion

Methodology

- Feature selection
  - Original gene expression data
    - Continuous data
    - 46585 (genes) x 432 (samples)
    - Possible DAG number: $n!2^{n(n-1)/2}$
  - Criteria
    - Genes with significant variance over all samples

$$\frac{\sigma}{\mu} > c, \sigma - \text{standar deviation}; \mu - \text{mean}$$

    - C = 1, $\sigma > 2$

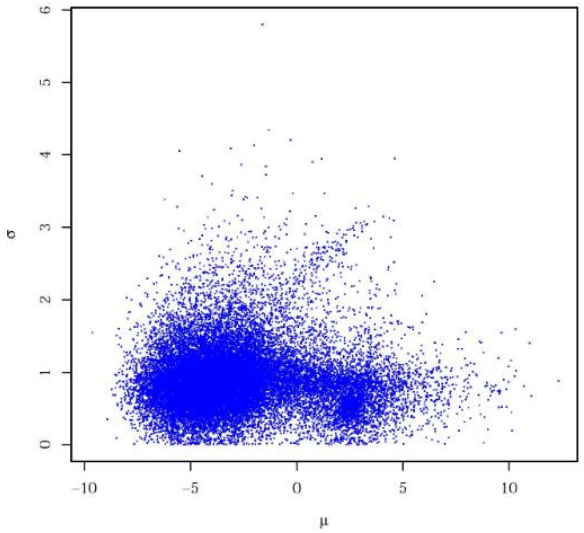
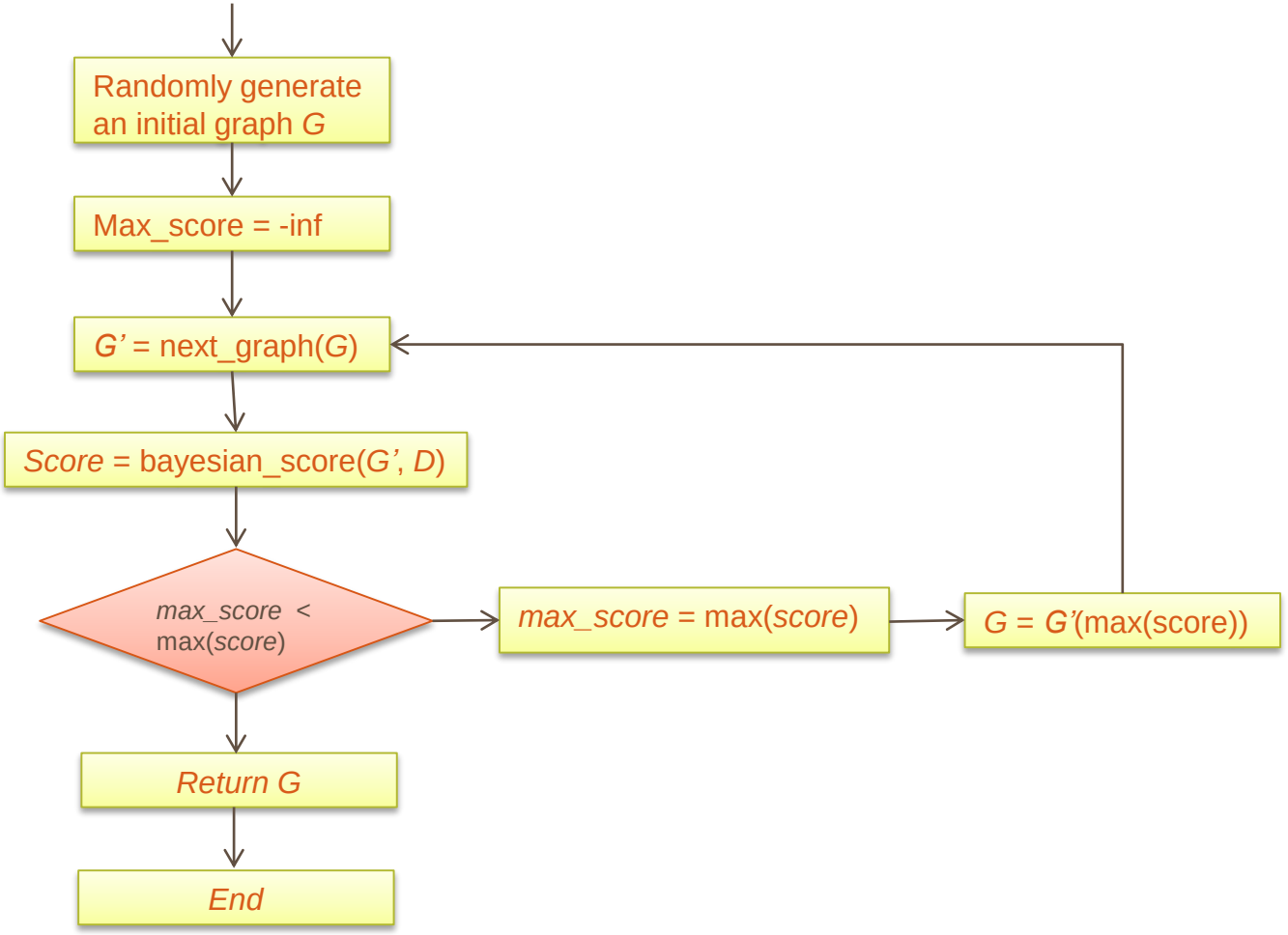


Figure 2: log(sd) vs. log(mean)

- Discretization
  - Normalized to [0,1] over all samples for each gene
  - Discretized to 0 (low-level) or 1 (high-level)

Oregon State University

Search Algorithms



Bayesian score & priors

- BDe score:

$$score_B(G,D) = \log P(D \mid G) + \log P(G)$$

- The posterior
  - Dirichlet prior

$$\Pr(D \mid G) = \int \Pr(D \mid G, \Theta) \Pr(\Theta \mid G) d\Theta$$

$$p(\Theta) \sim Dirichlet(\alpha_1, \alpha_2, ..., \alpha_k)$$

  - Closed form

$$\Pr(D \mid G) = \prod_{i=1}^{n} \prod_{u_i \in Val(Pa^G(X_i))} \frac{\Gamma(\alpha^G_{X_i|u_i})}{\Gamma(\alpha^G_{X_i|u_i} + M[u_i])} \prod_{x_i^j \in Val(X_i)} \left[\frac{\Gamma(\alpha^G_{X_i^j|u_i} + M[x_i^j, u_i])}{\Gamma(\alpha^G_{X_i^j|u_i})} \right]$$

$$\alpha^G_{x_i^j|u_i} = \frac{\alpha}{|x_i^j||u_i|}, \alpha = 2 \text{ (Friedman et al., 2008)}$$

- Structure prior

$$P(G) = e^{-|G|}, \text{ where } |G| \text{ represents \# of edges in G}$$

OSU Oregon State UNIVERSITY

How to calculate Gamma?

- Gamma(200) → INF
- Suppose each random variable X_i *is binary, for each* $u_i \in Val(Pa^G(X_i))$ *the inner product can be written as:*

$$\frac{\Gamma(\alpha_{x_i^0|u_i} + M[x_i^0, u_i])}{\Gamma(\alpha_{x_i^0|u_i})} \cdot \frac{\Gamma(\alpha_{x_i^1|u_i} + M[x_i^1, u_i])}{\Gamma(\alpha_{x_i^1|u_i})} \cdot \frac{\Gamma(\alpha_{x_i^0|u_i} + \alpha_{x_i^1|u_i})}{\Gamma(\alpha_{x_i^0|u_i} + \alpha_{x_i^1|u_i} + M[u_i])}$$

$$= \frac{\Gamma(\alpha_0 + M_0) \cdot \Gamma(\alpha_1 + M_1) \cdot \Gamma(\alpha_0 + \alpha_1)}{\Gamma(\alpha_0) \cdot \Gamma(\alpha_1) \cdot \Gamma(\alpha_0 + \alpha_1 + M_0 + M_1)}$$

$As\ \Gamma(x+1) = x\Gamma(x) = x(x-1)\Gamma(x-1) = \cdots$

Let $\alpha = \alpha_0 + \alpha_1, M = M_0 + M_1$, the equation above can be written as:

$$\frac{(\alpha_0 + M_0 - 1)(\alpha_0 + M_0 - 2)\cdots\alpha_0\Gamma(\alpha_0) \cdot (\alpha_1 + M_1 - 1)(\alpha_1 + M_1 - 2)\cdots\alpha_1\Gamma(\alpha_1)\Gamma(\alpha)}{\Gamma(\alpha_0) \cdot \Gamma(\alpha_1) \cdot (\alpha + M - 1)(\alpha + M - 2)\cdots\alpha\Gamma(\alpha)}$$

$$= \frac{(\alpha_0 + M_0 - 1)(\alpha_0 + M_0 - 2)\cdots\alpha_0 \cdot (\alpha_1 + M_1 - 1)(\alpha_1 + M_1 - 2)\cdots\alpha_1}{(\alpha + M - 1)(\alpha + M - 2)\cdots\alpha}$$

How to calculate Gamma? (cont.)

- Generally if X_i has k states, the formula can be written as:

$$Let\ \alpha=\alpha_1+\alpha_2+\cdots\alpha_k, m=m_1+m_2+\cdots+m_k$$

$$\Rightarrow \frac{(\alpha_1+m_1-1)(\alpha_1+m_1-2)\cdots\alpha_1(\alpha_2+m_2-1)(\alpha_2+m_2-2)\cdots\alpha_2\cdots(\alpha_k+m_k-1)((\alpha_k+m_k-2)\cdots\alpha_k}{(\alpha+m-1)(\alpha+m-2)\cdots\alpha}$$

with the log format:

$$\Rightarrow \sum_i \sum_{j=1}^{m_i} \log(\alpha_i+m_i-j)-\sum_{k=1}^{m}\log(\alpha+m-k)$$

- Then the closed form of the posterior probability:

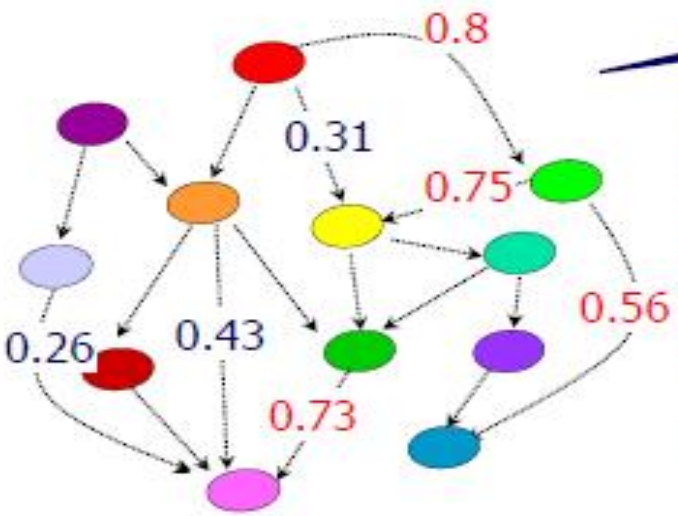
$$\sum_{i=1}^{n}\sum_{u_i\in val(pa(x_i))}((\sum_{j\in val(x_i)}\sum_{k=0}^{M[x_i^j,u_i]-1,}\log(\partial^g_{x_i^j|u_i}+k))-\sum_{k=0}^{M[u_i]-1}\log((\sum_{j\in val(x_i)}\partial^g_{x_i^j|u_i})+k))$$

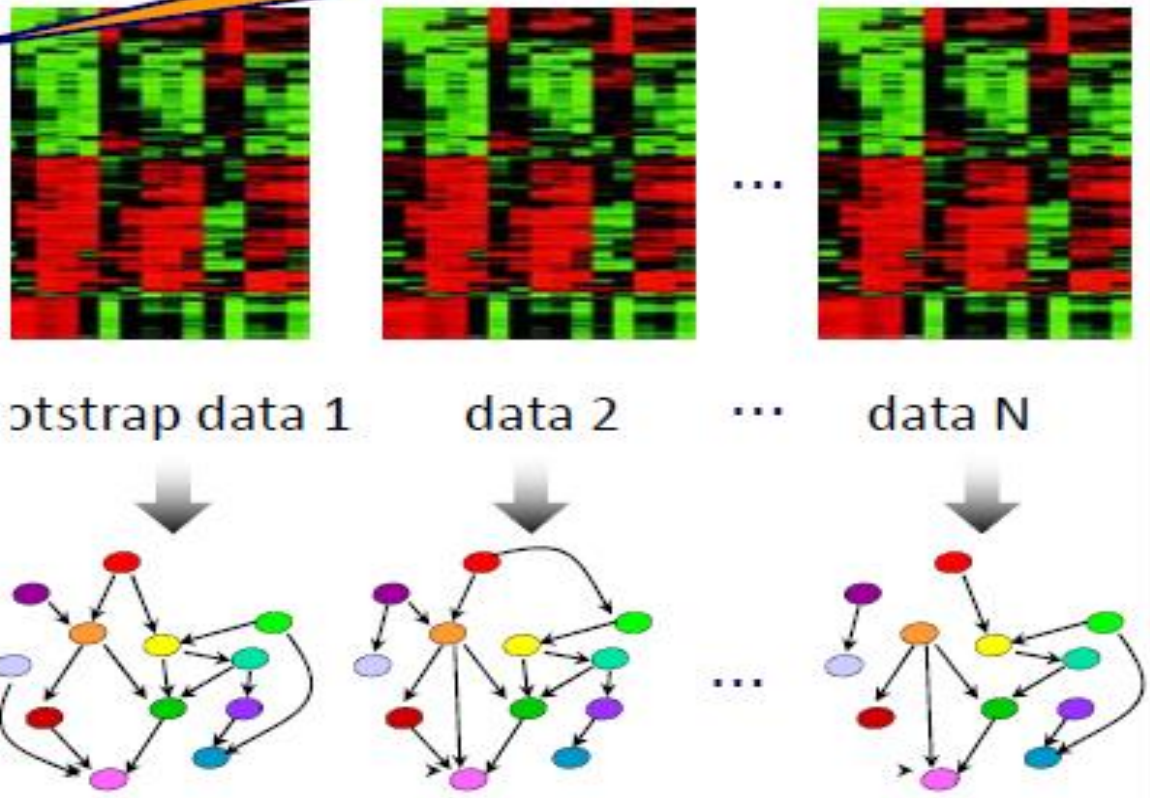
Bootstrapping

- Sampling with replacement

- Estimated confidence of each edge *i*

$$= \frac{\text{\# networks that contain the edge}}{\text{total \# networks (N)}}$$

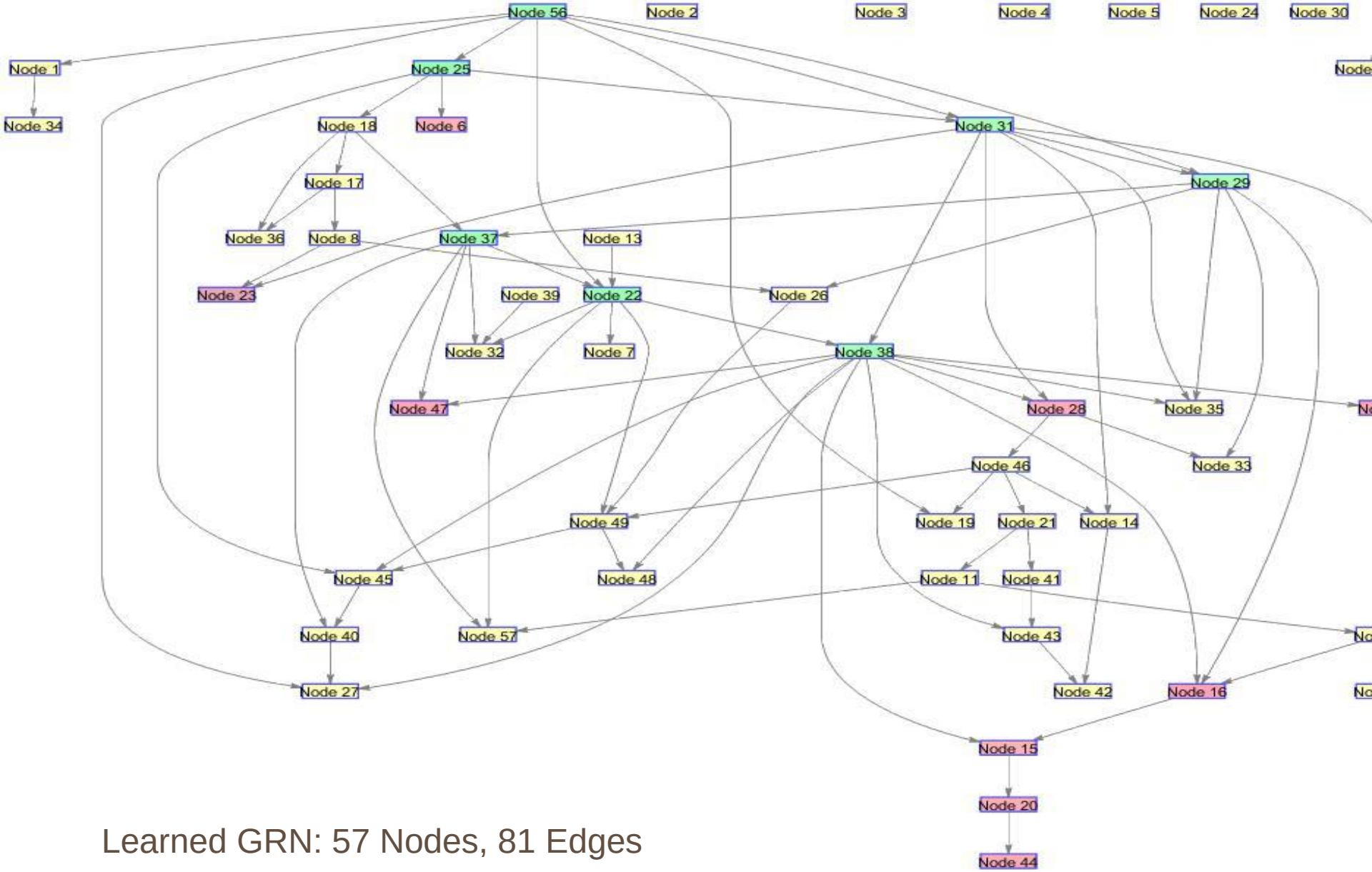




Inferring sub-networks from perturbed expression profiles, Pe'er et al. Bioinformatics 2001

Outline

- Introduction
- Related work
- Methodology
- *Result*
- Discussion
- Conclusion



Learned GRN: 57 Nodes, 81 Edges

Bootstrapping

- 74 iterations

Genes Name	Node #	Average out-degree
IGKV3-20	38	11.164
IGKV3D-7	31	8.6301
IGKV1OR10-1	37	7.2603
AC027612.6	22	6.6027
RP11-1166P10.8	25	6.0959
IGHV3OR16-9	56	5.9452
IGKV1OR9-2	29	5.0822
IGLV1-47	10	4.5753
IGKV1D-27	28	4.5068
IGHV4-4	15	4.3014

Gene pairs				Confidence level
SNORD3B-2	50	RP11-160E2.6	51	1
IGKV3D-7	31	IGLC6	23	0.93151
RP11-1166P10.8	25	IGKV4-1	6	0.89041
IGKV3-20	38	IGKV1D-42	9	0.87671
IGHV4-61	20	IGKV1-27	44	0.87671
IGKV3-20	38	IGKV1D-27	28	0.86301
IGHV4-4	15	IGHV4-61	20	0.86301
IGKV3-20	38	IGHV4OR15-8	47	0.83562
IGKV3-20	38	IGHV2-5	16	0.83562
IGKV3-20	38	IGHV4-4	15	0.83562

Discussions

- Genes with top-3 out-degrees
  - IGKV3-20, IGKV3D-7,IGKV10R10-1
  - These IgG related genes can be expressed in bladder cancer and drive cancer progression (Liang et al., 2013)
- Some genes relations are very robust
  - SNORD3B-2 → RP11-160E2.6 (1)
  - IGKV3D-7 → IGLC6 (0.93151)
  - Non-coding genes also have important role in cancer -- Verified by Nallar et al., 2013
- GRN is a sparse network (57 nodes, 81 edges)
  - No more than a few dozen genes directly affect its transcription

Discussions (cont.)

- TO-DOs in future
  - Apparently some genes were strongly enriched
    - Need better feature selection strategy
  - Search optimization
    - Combine priors from existing literature
    - Global optimum structure
  - Better validation
    - Compare with existing literature about gene interactions in cancer
    - Integrate with “driver” gene detection
  - A different model (Gaussian instead of multinomial?)

Conclusions

- Learned a Bayesian network structure with cancer gene expression data
- Enhanced approach for Bayesian score computation
- Verified the importance of the hub genes and the robust gene connections in the learned network through existing literature

Thank you!