Learning and representing molecular networks from data (Part 2)

Sushmita Roy

sroy@biostat.wisc.edu

Computational Network Biology

Biostatistics & Medical Informatics 826

https://compnetbiocourse.discovery.wisc.edu

Sep 20th 2018

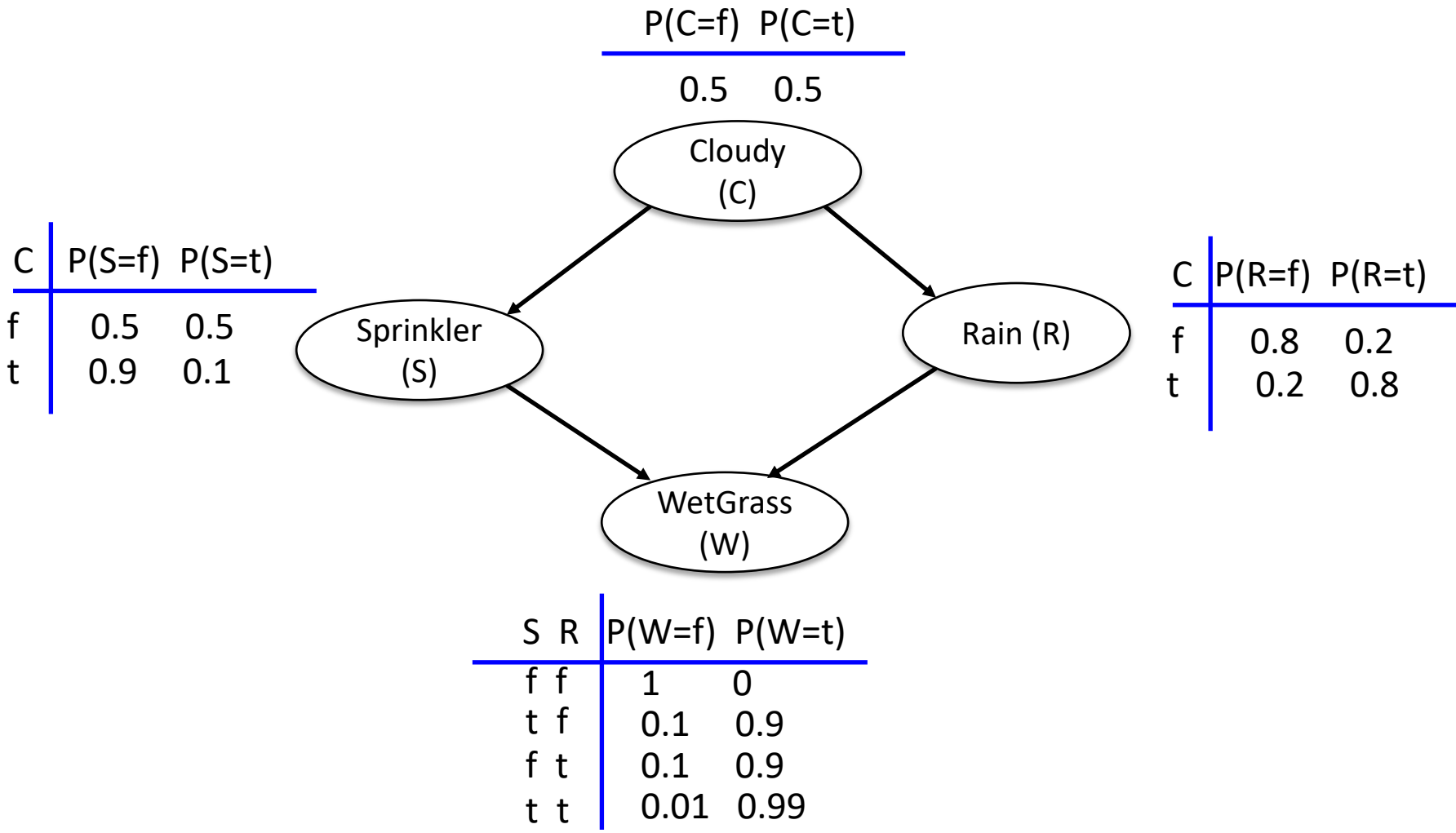
Goals for today

- Bayesian networks
- Learning Bayesian networks gene expression data
  - Sparse candidate (per-gene)
  - Module networks (per-module)

RECAP

- Expression-based network inference aims to infer regulatory networks from expression data
- Per-gene and per-module based methods
- Probabilistic graphical models are powerful representations of regulatory networks
  - Different PGMs encode different types of statistical dependencies
- Bayesian networks: DAG, CPD, Joint probability distribution

An example Bayesian network



Adapted from Kevin Murphy: Intro to Graphical models and Bayes networks:
http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html

Compute probabilities using a Bayesian network

What is $P(C = f, R = t, S = f, W = t)$

Bayes net allows us to write

$$P(W|S,R)P(S|C)P(R|C)P(C)$$

Plugging in the assignments for the variables:

$$P(W = t|S = f, R = t)P(S = f|C = f)$$
$$P(R = t|C = f)P(C = f)$$

Looking up in the CPD

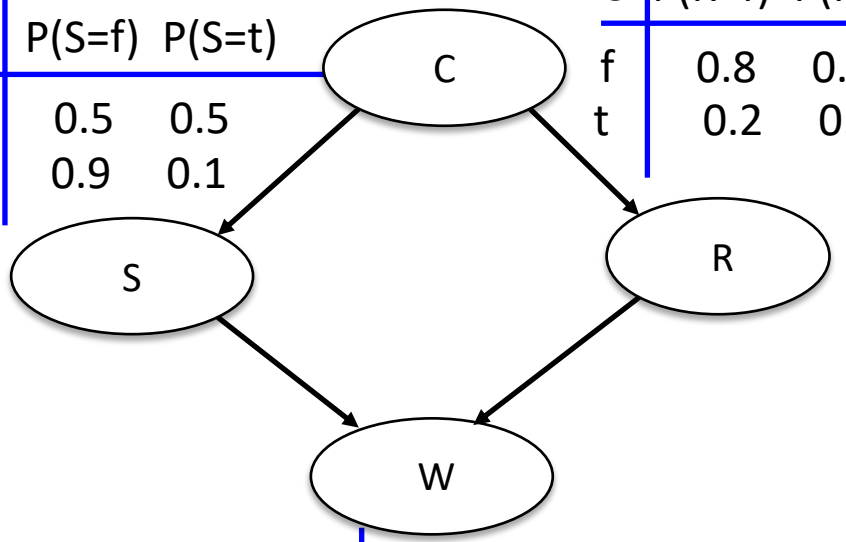
0.9*0.5*0.2*0.5

=0.045

P(C=f)	P(C=t)
0.5	0.5

C	P(S=f)	P(S=t)
f	0.5	0.5
t	0.9	0.1

C	P(R=f)	P(R=t)
f	0.8	0.2
t	0.2	0.8



S	R	P(W=f)	P(W=t)
f	f	1	0
t	f	0.1	0.9
f	t	0.1	0.9
t	t	0.01	0.99

Learning problems in Bayesian networks

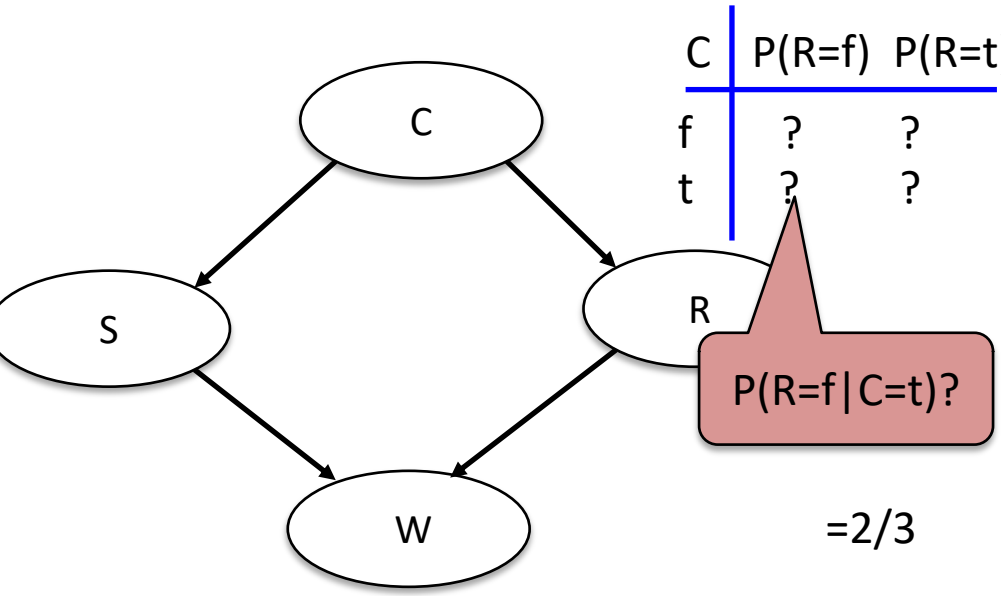
- Given a Bayesian network $B=\{G, \Theta\}$
- Parameter learning
  - Known graph structure G
  - Given a set of joint assignments of the random variables, estimate Θ, the parameters of the CPDs
- Structure learning
  - Given a set of joint assignments of the random variables, estimate the graph structure, G and parameters Θ
- Structure learning subsumes parameter learning

Bayes rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Estimating CPD from data

Supposed we had the following structure



And these observations for each variable

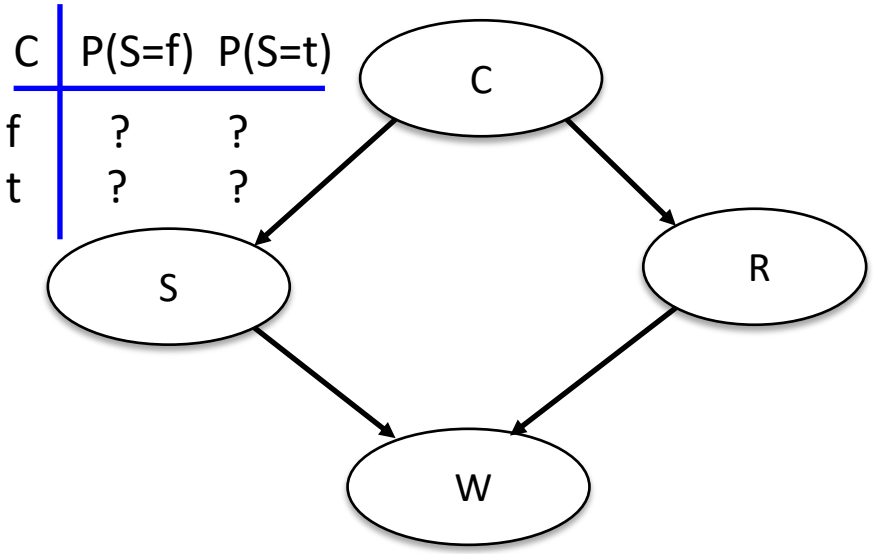
C	S	R	W
t	f	t	t
t	t	f	t
t	t	f	t
f	f	t	t
f	f	t	f
f	t	f	f
f	f	t	f

Parameters estimated in this way would be called the Maximum Likelihood (ML) parameters.

But we could put priors on the parameters and estimate a more robust set of parameters.

Estimating CPD from data

Supposed we had the following structure



And these observations for each variable

C	S	R	W
t	f	t	t
t	t	f	t
t	t	f	t
f	f	t	t
f	f	t	f
f	t	f	f
f	f	t	f

S	R	P(W=f)	P(W=t)
f	f	?	?
t	f	?	?
f	t	?	?
t	t	?	?

P(W=t|S=t,R=f)?

=2/3

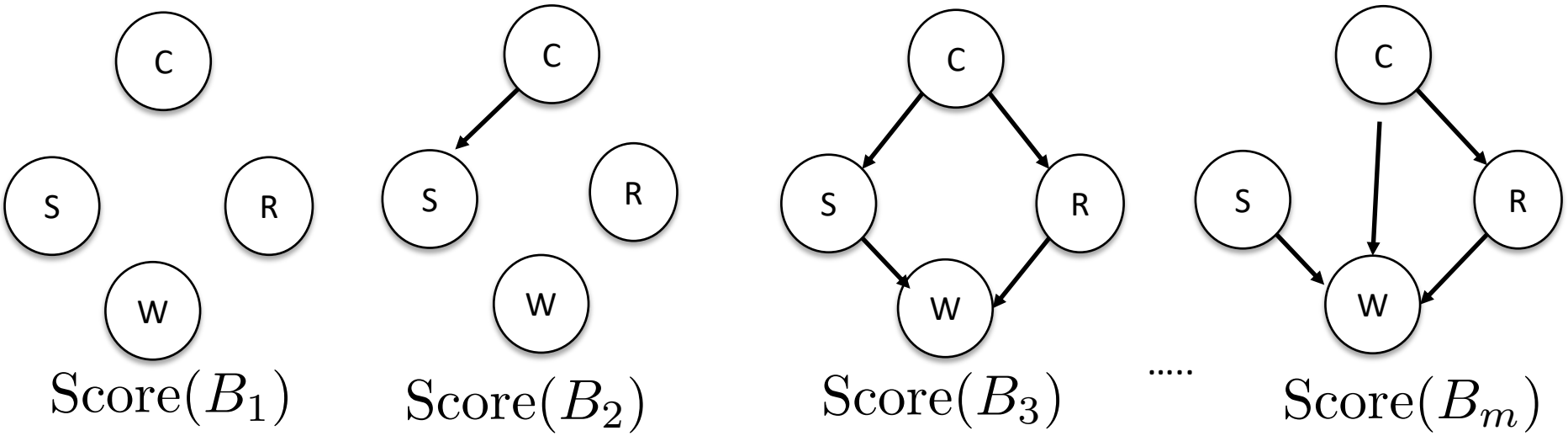
Structure learning

- Given a candidate graph how “good” is it?
  - Define a score of a graph
- What are possible candidate graphs?
  - Search over the space of possible graphs

Structure learning using score-based search

$B = \{G, \Theta\}$ A Bayesian network

$\text{Score}(B)$ Describes how well B describes the data



$\text{Score}(B_1)$ $\text{Score}(B_2)$ $\text{Score}(B_3)$ $\text{Score}(B_m)$

Scoring a Bayesian network

- Maximum likelihood score

$$\text{Score}_{ML}(\mathbf{G} : \mathbf{D}) = \log P(\mathbf{D}|\mathbf{G}, \boldsymbol{\Theta}_{\mathbf{ML}})$$

- Bayesian score

$$\text{Score}_{Bayes}(\mathbf{G} : \mathbf{D}) = \log P(\mathbf{G}|\mathbf{D})$$

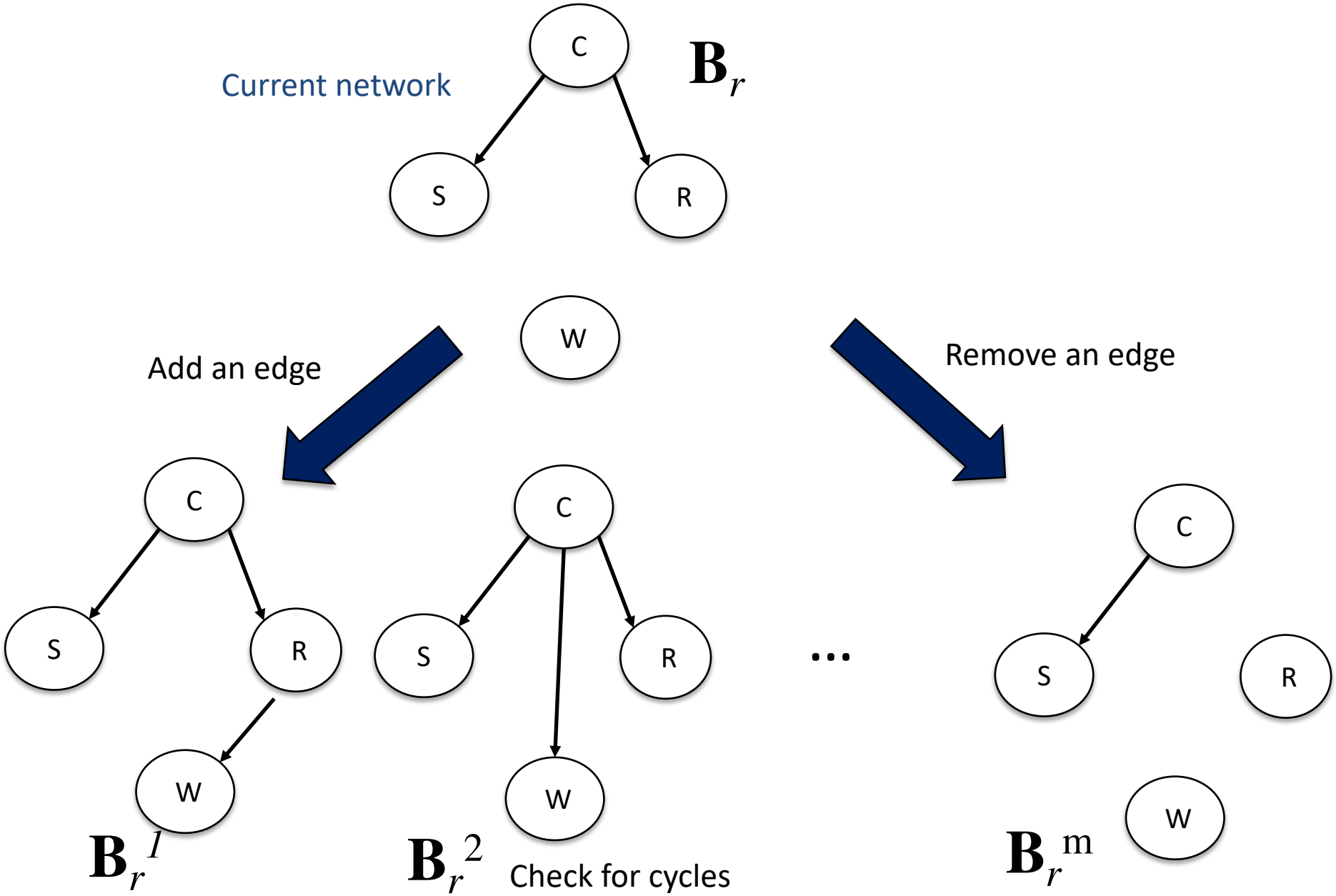
$$= \log \frac{P(\mathbf{D}|\mathbf{G})P(\mathbf{G})}{P(\mathbf{D})}$$

We typically ignore the denominator as it is the same for all models

Greedy hill climbing to search Bayesian network space

- Input: Data $\mathbf{D}$, An initial Bayesian network, $\mathbf{B}_0=\{\mathbf{G}_0, \mathbf{\Theta}_0\}$
- Output: $\mathbf{B}_{\text{best}}$
- Loop for $r=1, 2..$ until convergence:
  - $\{\mathbf{B}_r^1, .., \mathbf{B}_r^m\}$ = *Neighbors*($\mathbf{B}_r$) by making local changes to $\mathbf{B}_r$
  - $\mathbf{B}_{r+1}$: $arg\ max_j(\text{Score}(\mathbf{B}_r^j))$
- Termination:
  - $\mathbf{B}_{\text{best}}= \mathbf{B}_r$

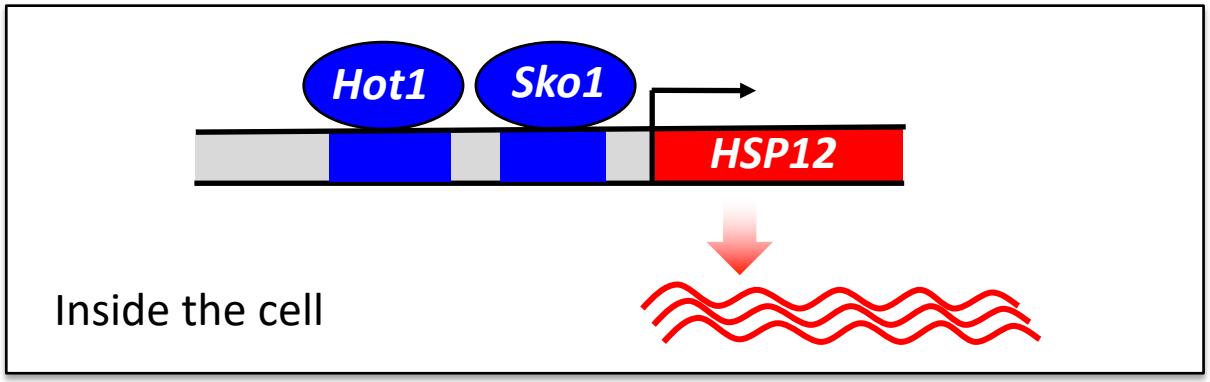
Local changes to $\mathbf{B}_r$

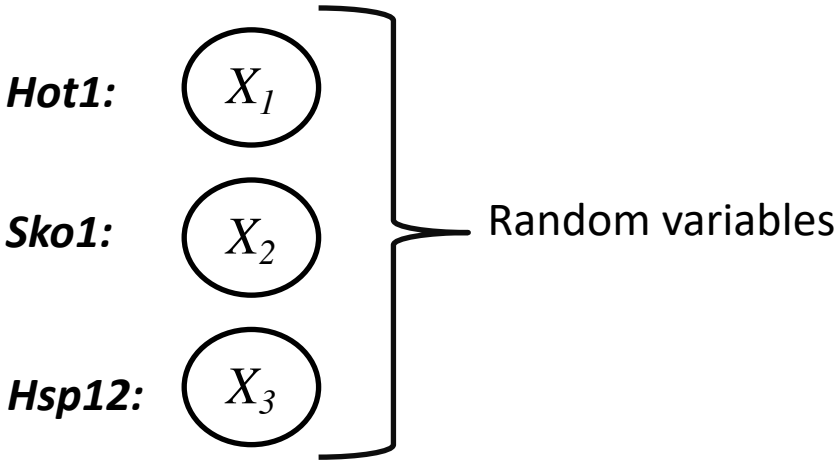


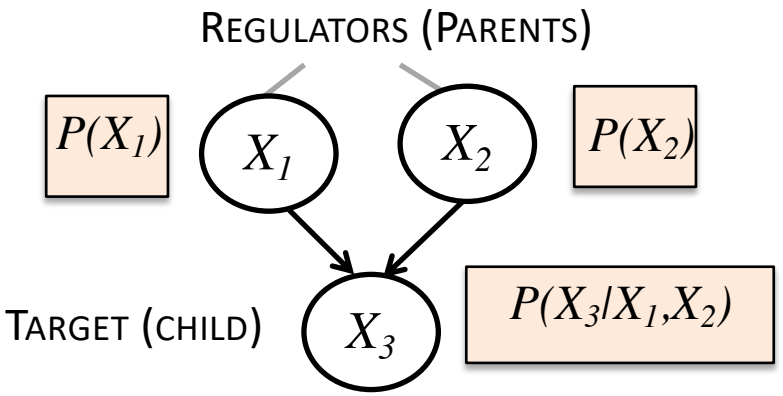
Goals for today

- Bayesian networks
- Learning Bayesian networks gene expression data
  - Sparse candidate (per-gene)
  - Module networks (per-module)

Bayesian network representation of a regulatory network

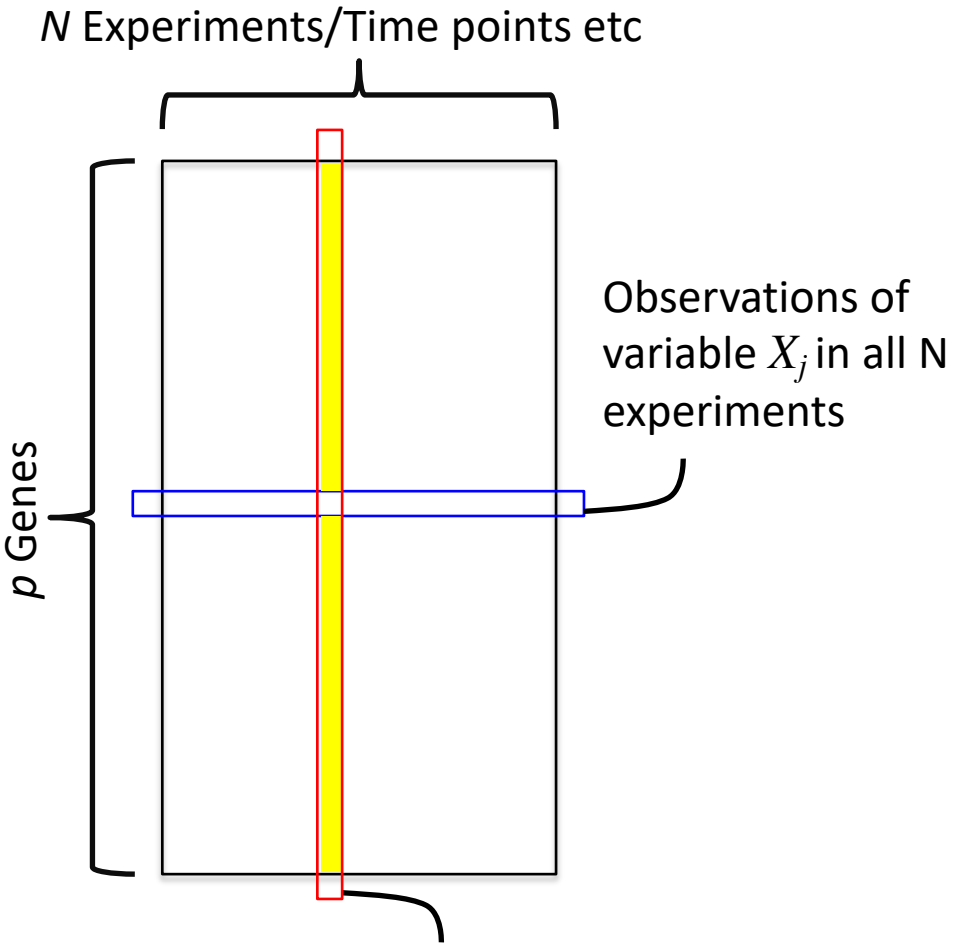






Bayesian network

Expression data matrix

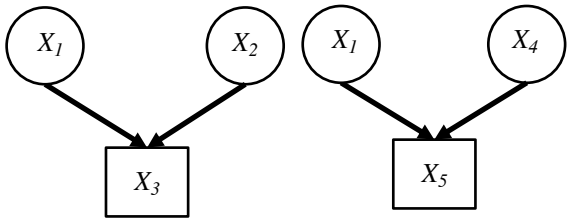


Challenges with applying Bayesian network to genome-scale data

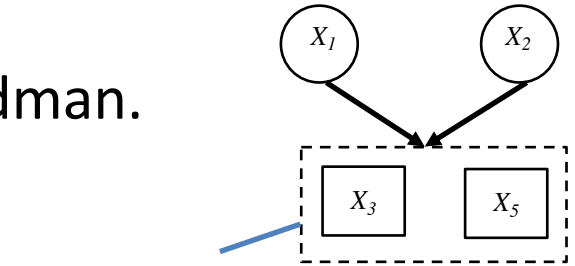
- Number of variables, p is in thousands
- Number of samples, N is in hundreds

Bayesian network-based methods to handle genome-scale networks

- Sparse candidate algorithm
  - Friedman, Nachman, Pe'er. 1999
  - Friedman, Linial, Nachman, Pe'er. 2000.
- Module networks
  - Segal, Pe'er, Regev, Koller, Friedman. 2005



Per-gene



Module (Cluster)

Per-module

The Sparse candidate algorithm for structure learning in Bayesian networks

- A fast Bayesian network learning algorithm
- Key idea: Identify k "promising" candidate parents for each X_i
  - $k<<p$, p: number of random variables
  - Candidates define a "skeleton graph" $\mathbf{H}$
- Restrict graph structure to select parents from $\mathbf{H}$
- Early choices in $\mathbf{H}$ might exclude other good parents
  - Resolve using an iterative algorithm

Sparse candidate algorithm

- Input:
  - A data set $\mathbf{D}$
  - An initial Bayes net $\mathbf{B}_0$
  - A parameter k: max number of parents per variable
- Output:
  - Final $\mathbf{B}_r$
- Loop for $r=1,2..$ until convergence
  - *Restrict*
    - Based on $\mathbf{D}$ and $\mathbf{B}_{r-1}$ select candidate parents C_i^r for X_i
    - This defines a skeleton directed network $\mathbf{H}_r$
  - *Maximize*
    - Find network $\mathbf{B}_r$ that maximizes the score $\mathrm{Score}(\mathbf{B}_r)$ among networks satisfying $$Pa^r(X_i) \subseteq C_i^r$$
- Termination: Return $\mathbf{B}_r$

Information theory for measuring dependence

- $I(X;Y)$ is the mutual information between two variables
  - Knowing X, how much information do we have for Y
- $P(Z)$ is the probability distribution of Z

$$I(X;Y) = \Sigma_{x,y \in X,Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

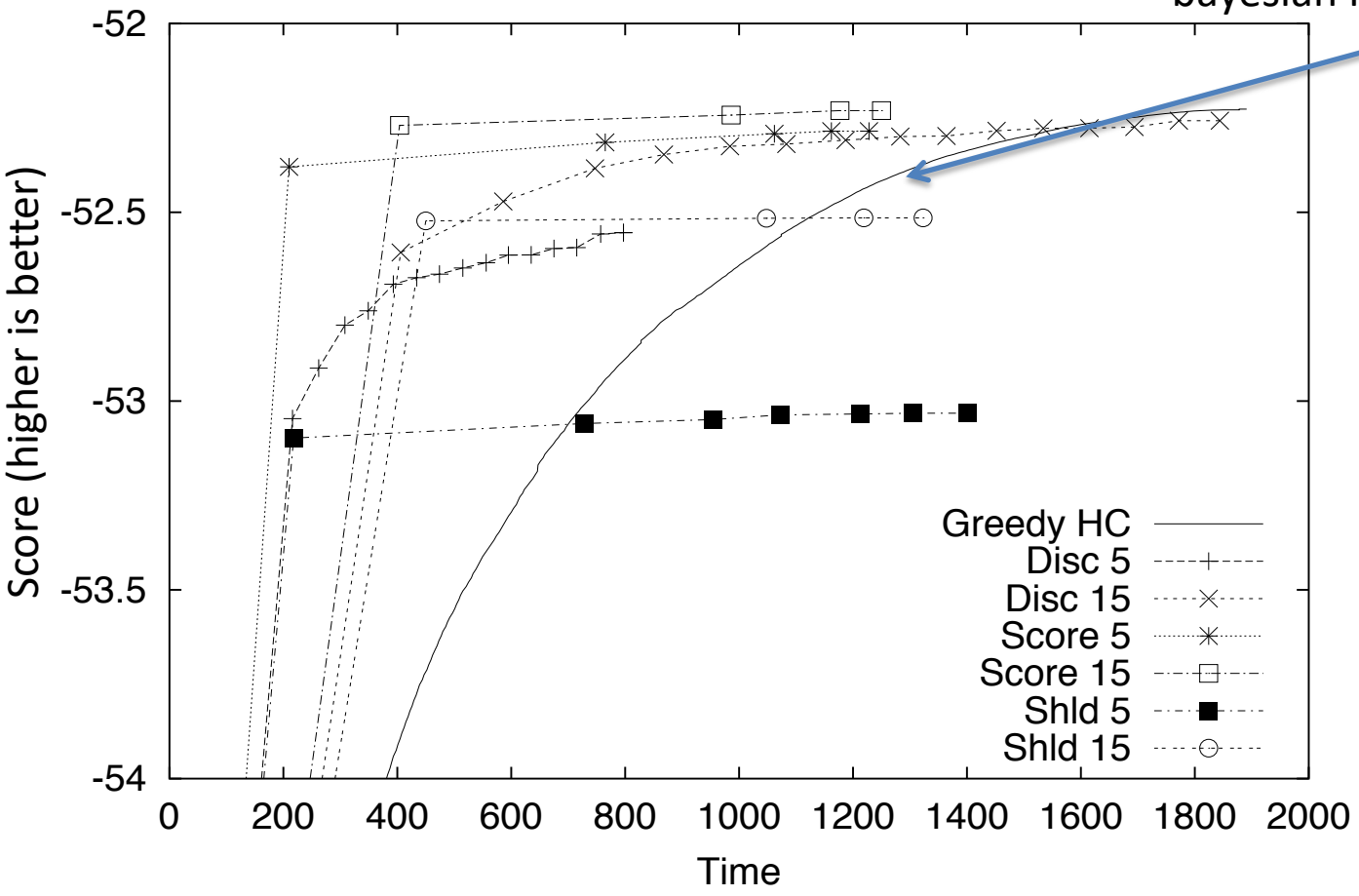
- Measures the difference between the two distributions: joint and product of marginals

Selecting candidate parents in the Restrict Step

- Mutual information is used only in the first step
- Disc: A good parent for X_i is one with strong statistical dependence with X_i
  - This is called a "Disc"repancy heuristic because it measures the discrepancy between $P'(X,Y)$ as described by the Bayesian network and $P(X,Y)$ estimated by the data.
- Shield: A good parent for X_i is one that captures most of the information of X_i
  - How much information do we gain if we add X_j to $Pa(X_i)$
- Score: A good parent for X_i has the highest score improvement when added to $Pa(X_i)$

Sparse candidate learns good networks faster than hill-climbing

Greedy hill climbing takes much longer to reach a high scoring bayesian network

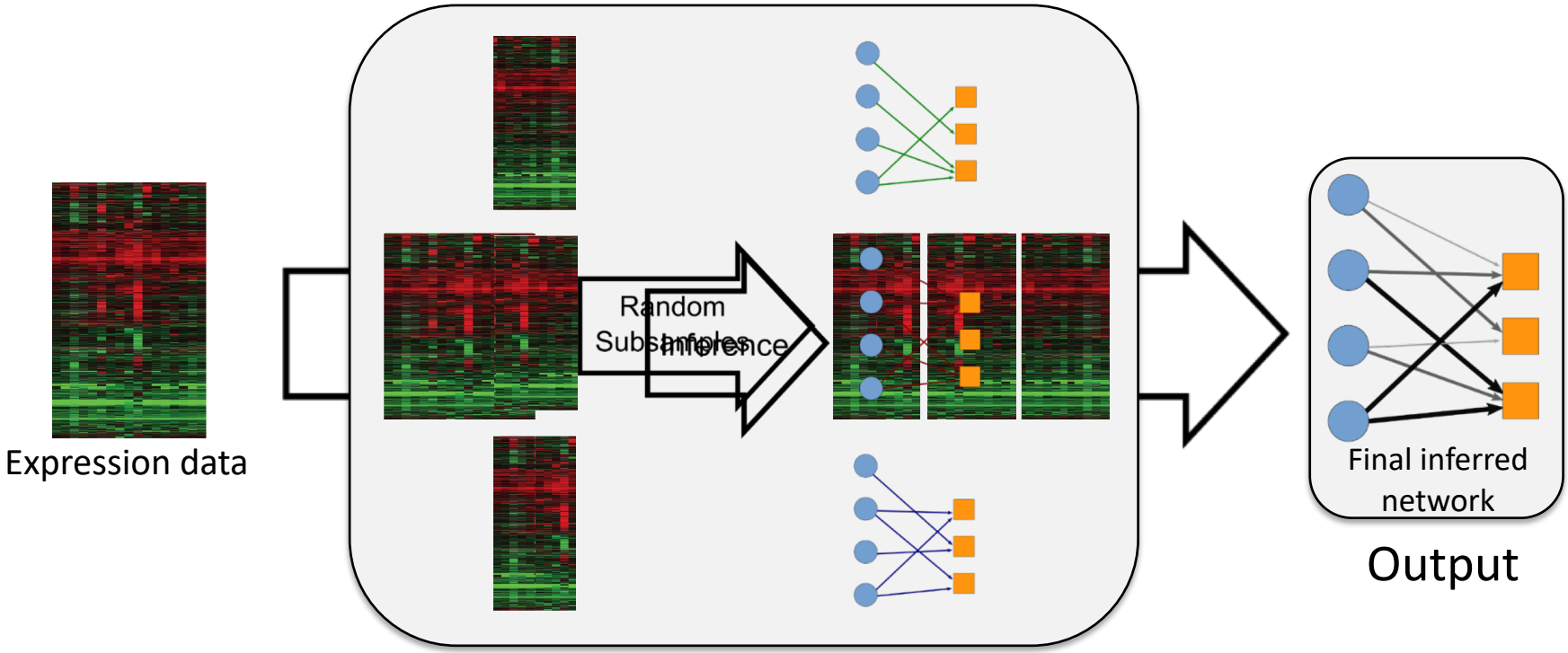


Size of Bayesian network: 100 variables

Some comments about choosing candidates

- How to select k in the sparse candidate algorithm?
- Should k be the same for all X_i ?
- Estimate an undirected *dependency network*
  - Learn a Bayesian network constrained on the dependency network structure
- Regularized regression approaches can be used to estimate the structure of an undirected graph
  - Schmidt, Niculescu-Mizil, Murphy 2007

Typically we will not learn one network



Bootstrap or stability selection to get edge confidence

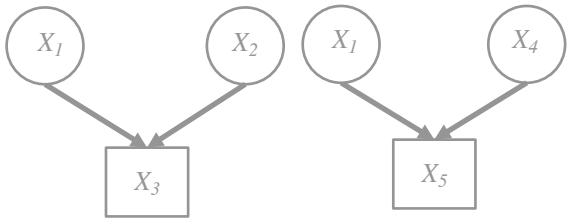
Slide credit: Alireza Fotuhi Siahpirani

Goals for today

- Bayesian networks
- Learning Bayesian networks gene expression data
  - Sparse candidate (per-gene)
  - Module networks (per-module)

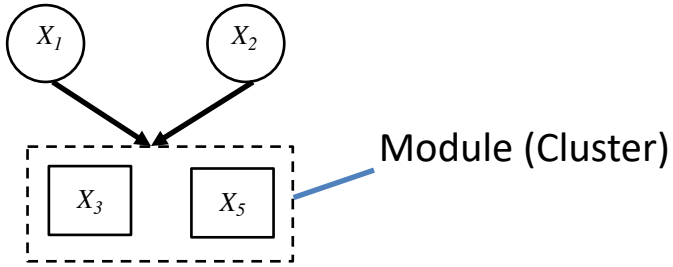
Bayesian network-based methods to handle genome-scale networks

- Sparse candidate algorithm
  - Friedman, Nachman, Pe'er. 1999
  - Friedman, Linial, Nachman, Pe'er. 2000.



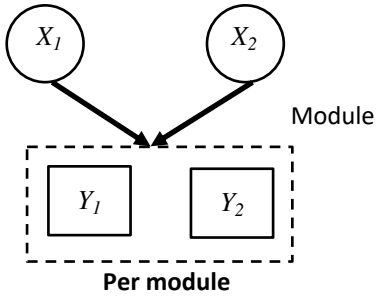
Per-gene

- Module networks
  - Segal, Pe'er, Regev, Koller, Friedman. 2005



Per-module

Per-module methods



- Find regulators for an entire module
  - Assume genes in the same module have the same regulators
- Module Networks (Segal et al. 2005)
- Stochastic LeMoNe (Joshi et al. 2008)

Module Networks

- Motivation:
  - Most complex systems have too many variables
  - Not enough data to robustly learn networks
  - Large networks are hard to interpret
- Key idea: Group similarly behaving variables into "modules" and learn the same parents and parameters for each module
- Relevance to gene regulatory networks
  - Genes that are co-expressed are likely regulated in similar ways

Segal et al 2005, JMLR

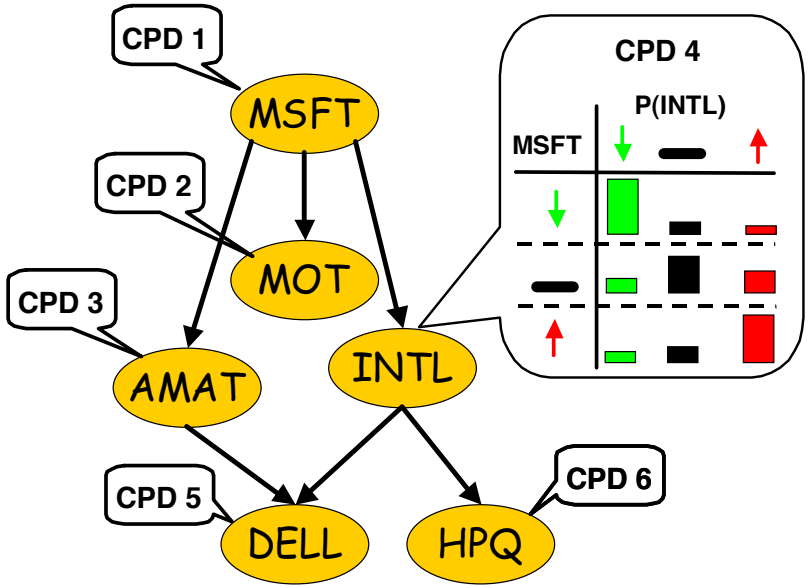
Definition of a module

- Statistical definition (specific to module networks by Segal 2005)
  - A set of random variables that share a statistical model
- Biological definition of a module
  - Set of genes that are co-expressed and co-regulated

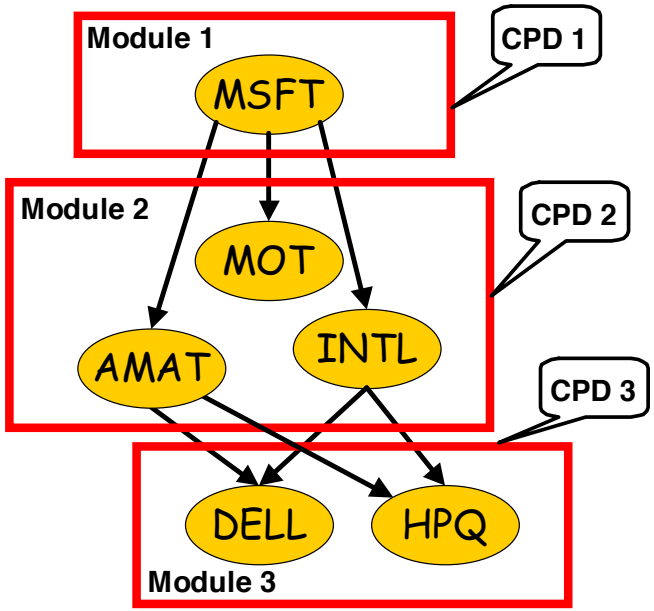
Bayesian network vs Module network

- Bayesian network
  - Different CPD per random variable
  - Learning only requires to search for parents
- Module network
  - CPD per module
    - Same CPD for all random variables in the same module
  - Learning requires parent search and module membership assignment

Bayesian network vs Module network



(a) Bayesian network



(b) Module network

Each variable takes three values: UP, DOWN, SAME

Some notation for a Module Network

- N random variables $\mathbf{X} = \{X_1, \cdots, X_N\}$
- Set of module variables $M_1 .. M_K$
- Module assignments A that specifies the module (1-to-K) for each X_i
- CPD per module $P(M_j|Pa_{Mj})$, Pa_{Mj} are parents of module M_j
  - Each variable X_i in M_j has the same conditional distribution

Learning a Module Network

- Given training dataset $\mathbf{D} = \{\mathbf{x}^1, \cdots, \mathbf{x}^m\}$, fixed number of modules, K
- Learn
  - Module assignments A of each variable to a module
  - The parents of each module to give structure S

Score of a module network

- Module network makes use of a Bayesian score

$$P(\mathcal{S}, \mathcal{A} \mid \mathcal{D}) \propto P(\mathcal{A}) P(\mathcal{S} \mid \mathcal{A}) P(\mathcal{D} \mid \mathcal{S}, \mathcal{A})$$

Priors

Data likelihood

$$\text{score}(\mathcal{S}, \mathcal{A} : \mathcal{D}) =$$

$$\log P(\mathcal{A}) + \log P(\mathcal{S} \mid \mathcal{A}) + \log P(\mathcal{D} \mid \mathcal{S}, \mathcal{A}).$$

Marginal likelihood

Priors

Score of a module network continued

Integrate parameters out

$$\log P(\mathcal{D}|\mathbf{S},\mathbf{A}) = \log \int P(\mathcal{D}|\mathbf{S},\mathbf{A},\theta)P(\theta|\mathbf{S},\mathbf{A})d\theta$$

Decomposes over each module

$$\log \prod_{j=1}^{k} \int L_j(\mathbf{U},\mathbf{X},\theta_{\mathbf{M}_j|\mathbf{U}} : \mathcal{D})P(\theta_{\mathbf{M}_j}|\mathbf{U})d\theta_{\mathbf{M}_j|\mathbf{U}}$$

Decomposes over each module

$$\sum_{j=1}^{K} \log \int L_j(\mathbf{U},\mathbf{X},\theta_{\mathbf{M}_j|\mathbf{U}} : \mathcal{D})P(\theta_{\mathbf{M}_j}|\mathbf{U})d\theta_{\mathbf{M}_j|\mathbf{U}}$$

$\mathbf{U}$: Set of parents defined by $\mathbf{S}$

$\mathbf{X}$: Set of variables.

For computing each L_j term we would need only the variables and parents associated with module j

Defining the likelihood

$$\mathbf{X}^j = \{X_i \in \mathbf{X} | A(X_i) = j\}$$

Likelihood of module j $L_j(\mathbf{Pa}_{M_j}, \mathbf{X}^j, \theta_j : \mathcal{D})$

$$L_j = \prod_{m=1}^{|\mathcal{D}|} \prod_{X_i \in \mathbf{X}^j} P(x_i[m] | \mathbf{pa}_{M_j}[m], \theta_j)$$

K: number of modules, $\boldsymbol{X}^j$: j^{th} module $\boldsymbol{Pa}_{Mj}$ Parents of module M_j

Module network learning algorithm

Input:

D // Data set

K // Number of modules

Output:

M // A module network

Learn-Module-Network

$\mathcal{A}_0$ = cluster $\mathcal{X}$ into K modules

$\mathcal{S}_0$ = empty structure

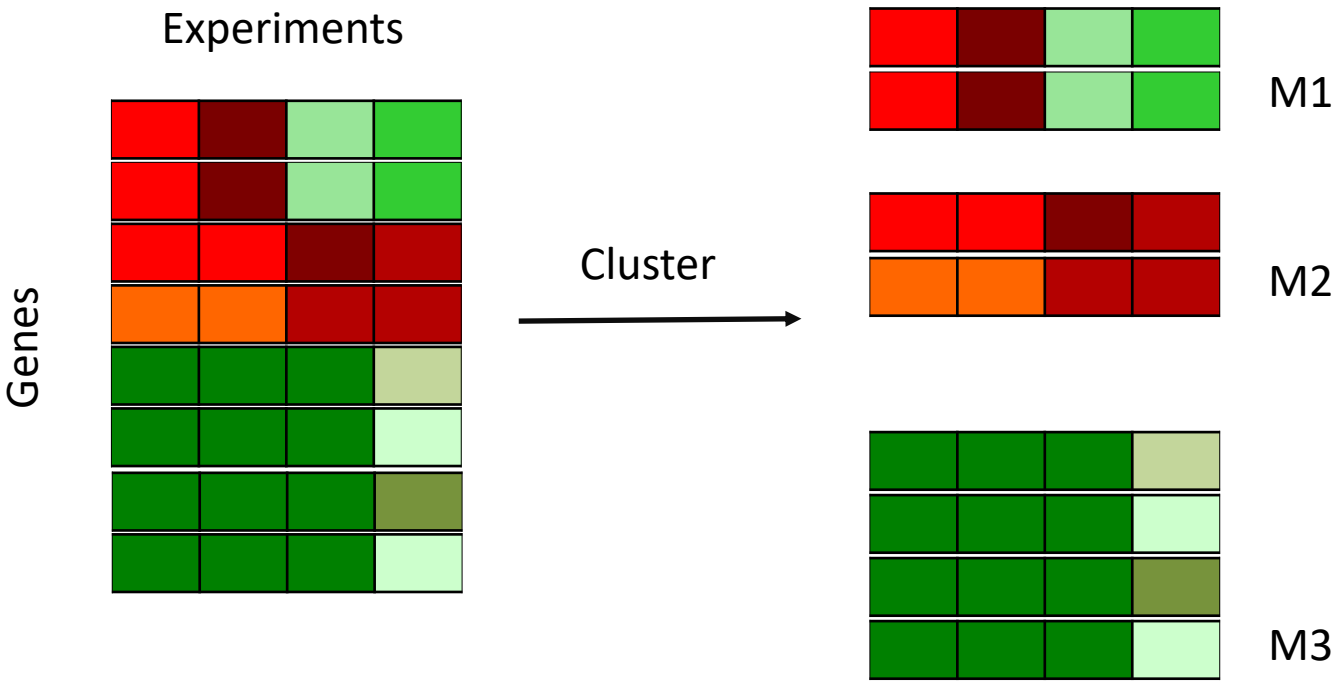
Loop $t = 1, 2, \ldots$ until convergence

$\mathcal{S}_t$ = Greedy-Structure-Search($\mathcal{A}_{t-1}, \mathcal{S}_{t-1}$)

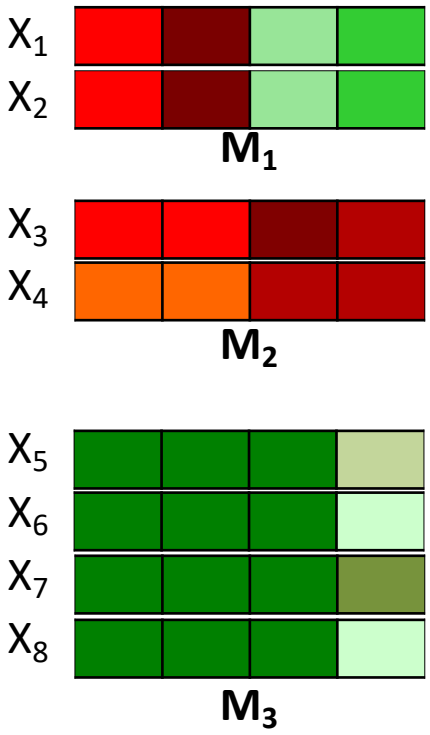
$\mathcal{A}_t$ = Sequential-Update($\mathcal{A}_{t-1}, \mathcal{S}_t$);

Return M = $(\mathcal{A}_t, \mathcal{S}_t)$

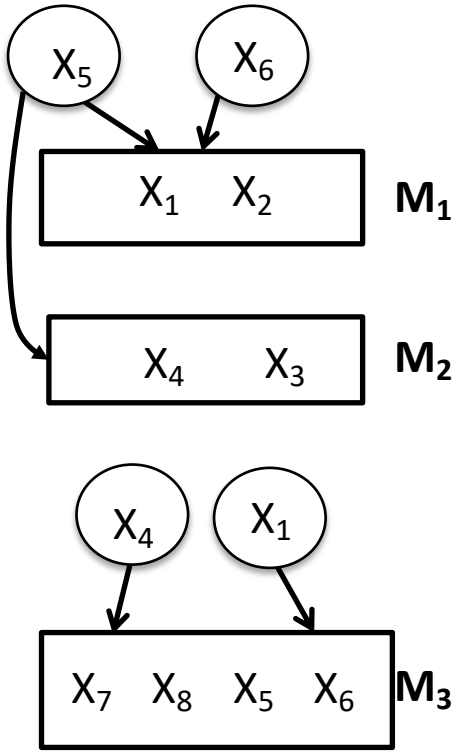
Initial modules identified by expression clustering



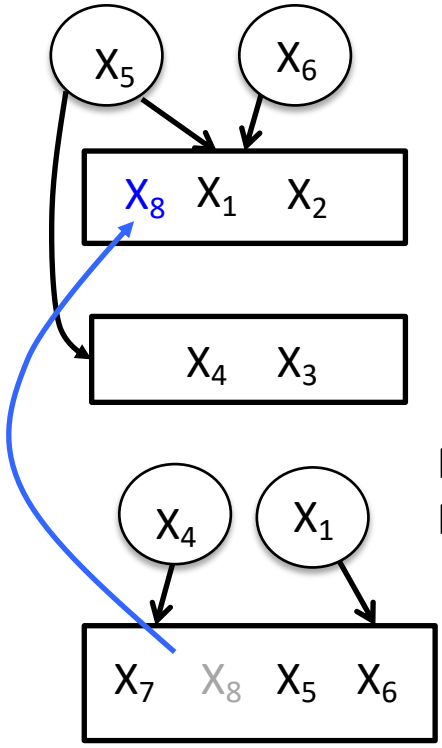
Iterations in learning Module Networks



Learn regulators/CPD per module



Revisit the modules



Module M_1 and M_3 get updated

Module re-assignment

- Must preserve the acyclic graph structure
- Must improve score
- Module re-assignment happens using a *sequential update* procedure:
  - Update only one variable at a time
  - The change in score of moving a variable from one module to another while keeping the other variables fixed

Module re-assignment via sequential update

Input:

D // Data set

$\mathcal{A}_0$ // Initial assignment function

$\mathcal{S}$ // Given dependency structure

Output:

$\mathcal{A}$ // improved assignment function

Sequential-Update

$\mathcal{A} = \mathcal{A}_0$

Loop

For $i = 1$ to n

For $j = 1$ to K

$\mathcal{A}' = \mathcal{A}$ except that $\mathcal{A}'(X_i) = j$

If $\langle \mathcal{G}_{\mathcal{M}}, \mathcal{A}' \rangle$ is cyclic, **continue**

If $\text{score}(\mathcal{S}, \mathcal{A}' : \mathcal{D}) > \text{score}(\mathcal{S}, \mathcal{A} : \mathcal{D})$

$\mathcal{A} = \mathcal{A}'$

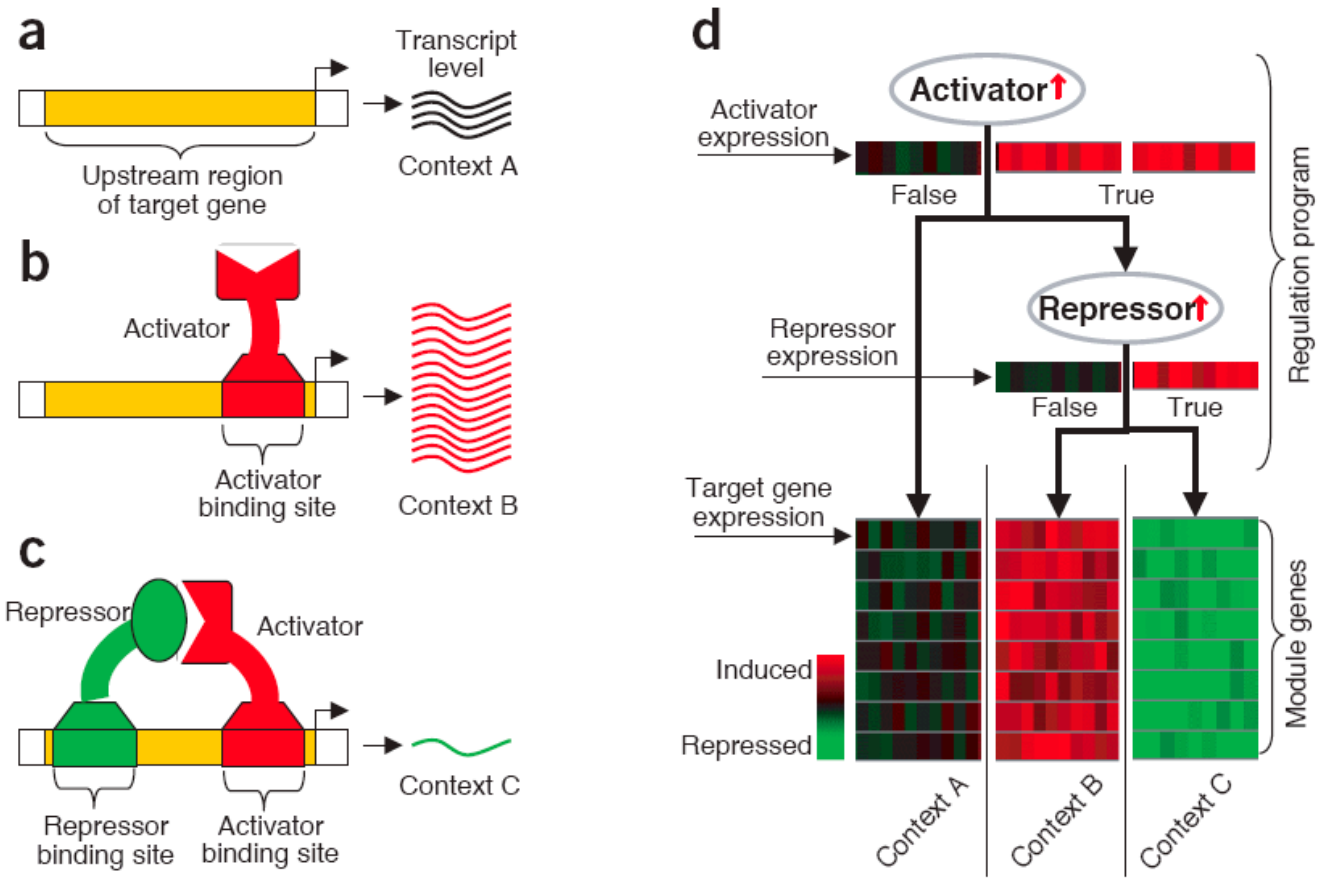
Until no reassignments to any of $X_1, \ldots X_n$

Return $\mathcal{A}$

Modeling questions in Module Networks

- How to score and learn module networks?
- How to model the CPD between parent and children?
  - Regression Tree
    - Applicable to continuous variables
    - Captures non-linear dependencies
    - Captures context-specific dependencies

Modeling the relationship between regulators and targets



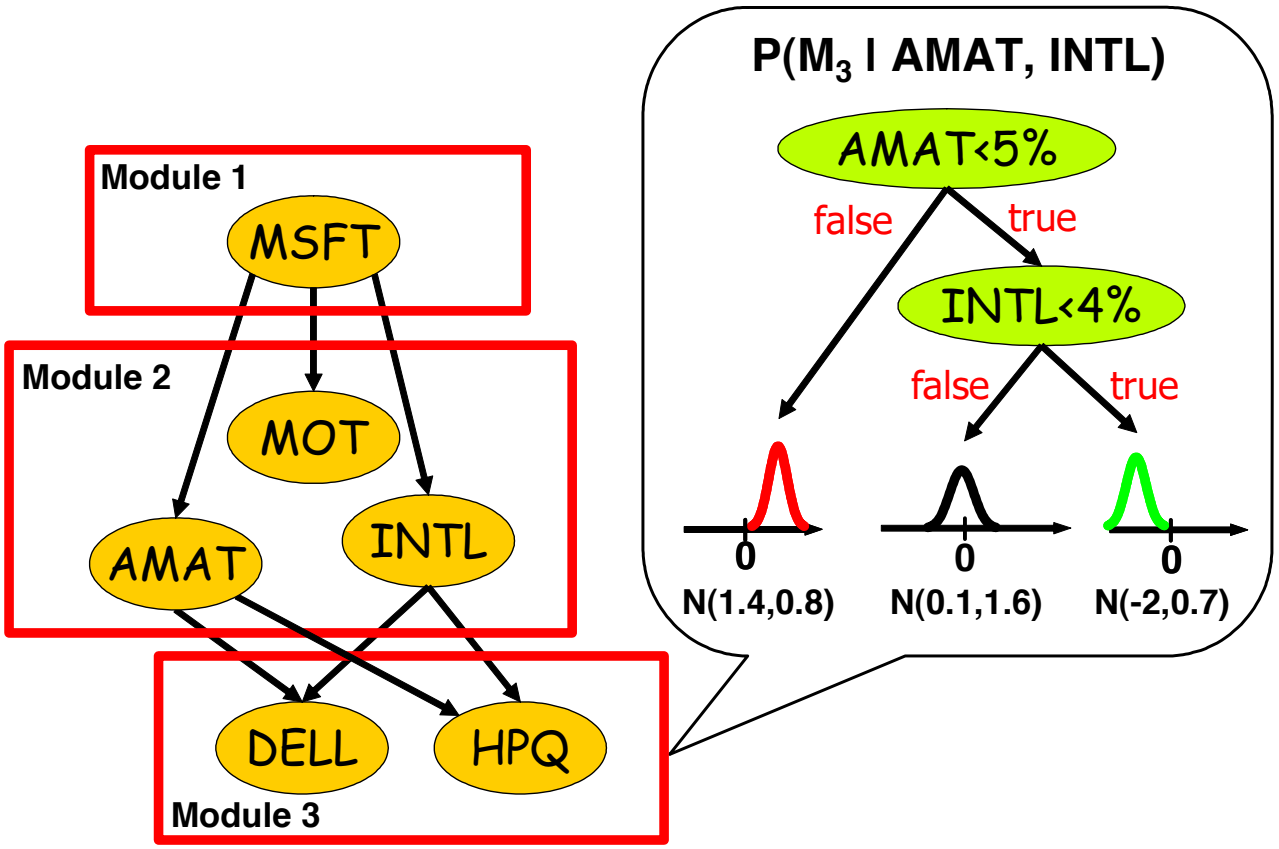
- suppose we have a set of (8) genes that all have in their upstream regions the same activator/repressor binding sites

Segal et al., Nature Genetics 2003

A regression tree

- A rooted binary tree T
- Each node in the tree is either an interior node or a leaf node
- Interior nodes are labeled with a binary test $X_i < u$, u is a real number observed in the data
- Leaf nodes are associated with univariate distributions of the child

An example regression tree for a Module network

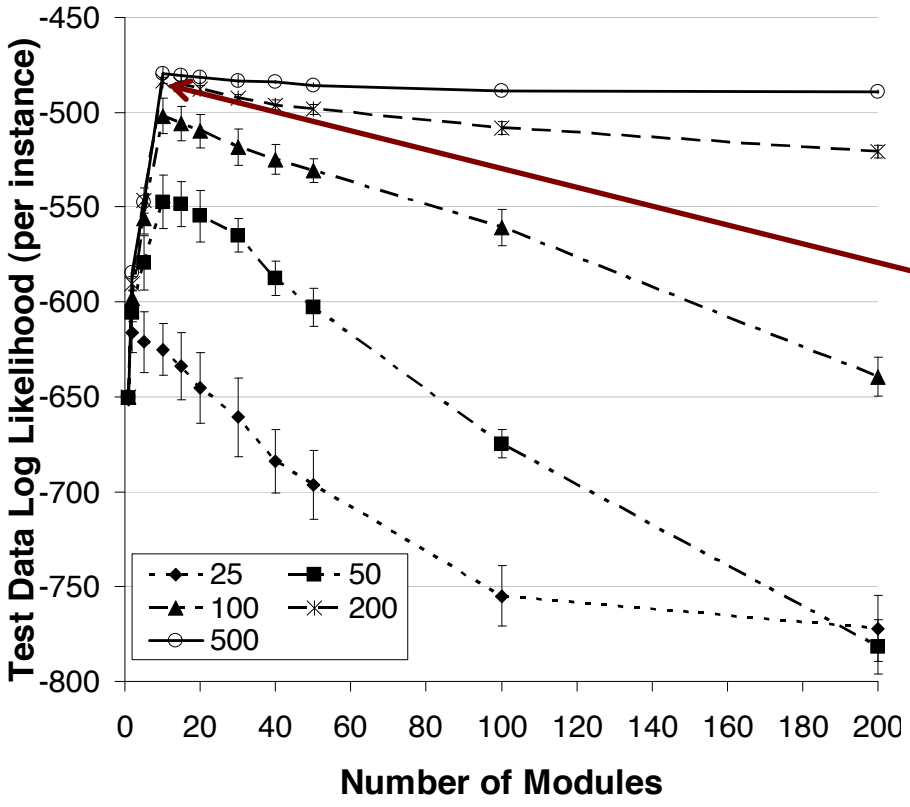


Module 3 values are modeled using Gaussians at each leaf node

Assessing the value of using Module Networks

- Using simulated data
  - Generate data from a known module network
  - Known module network was in turn learned from real data
    - 10 modules, 500 variables
  - Evaluate using
    - Test data likelihood
    - Recovery of true parent-child relationships are recovered in learned module network
- Using gene expression data
  - External validation of modules (Gene ontology, motif enrichment)
  - Cross-check with literature

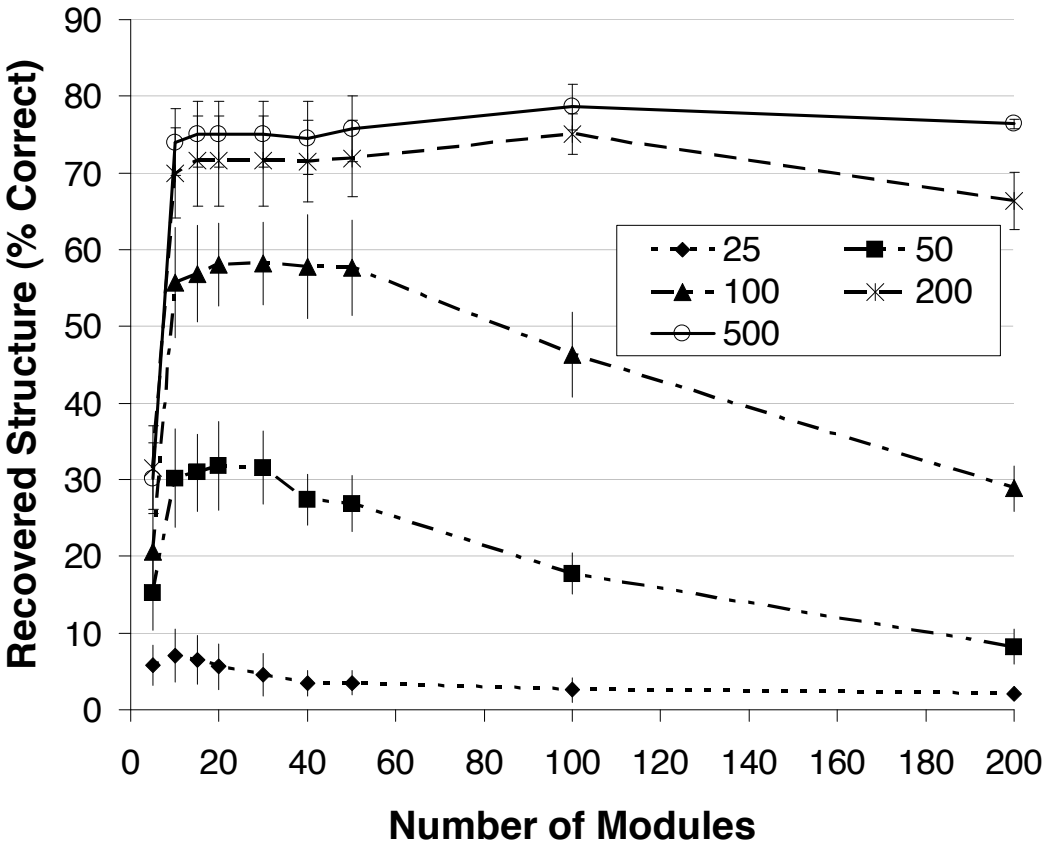
Test data likelihood



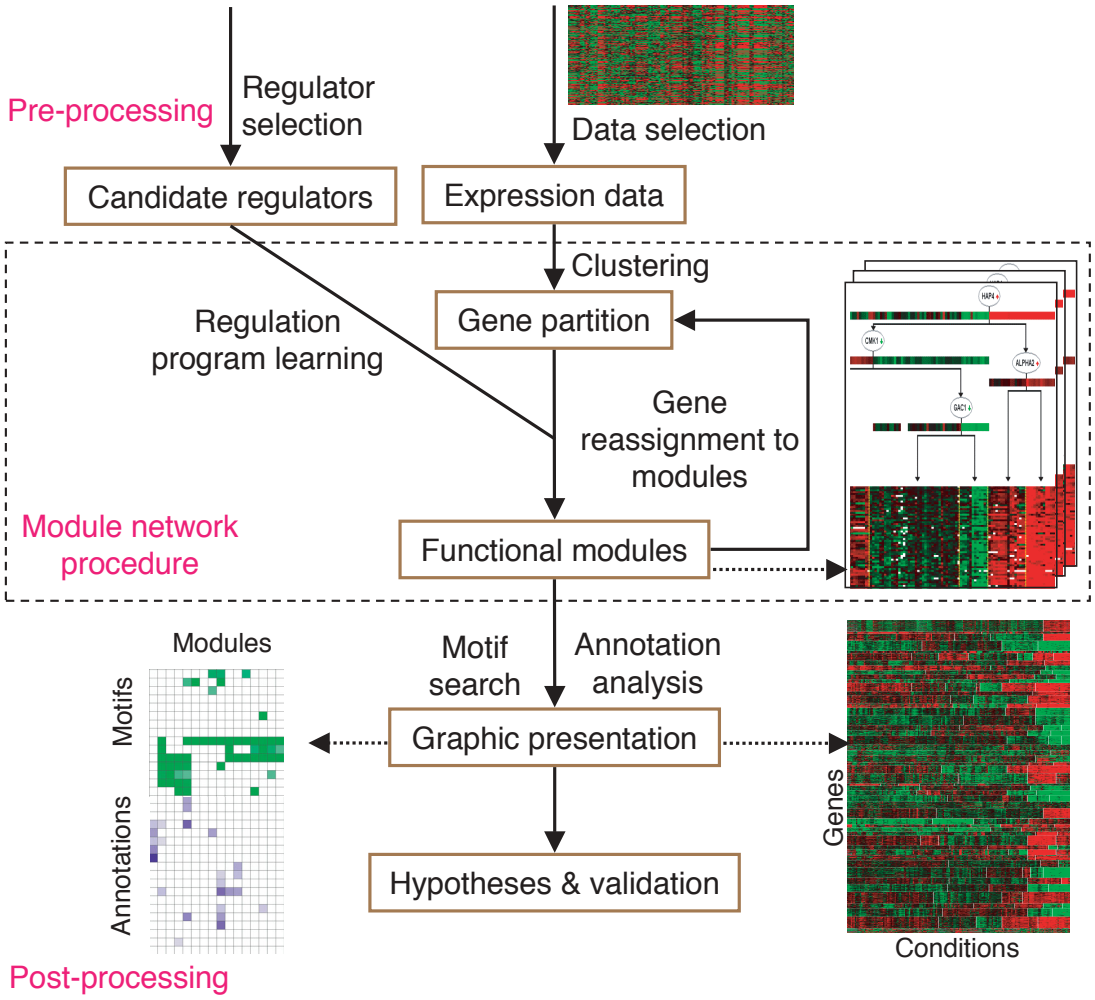
10 Modules is the best for almost all training data set sizes

Each line type represents size of training data

Recovery of graph structure



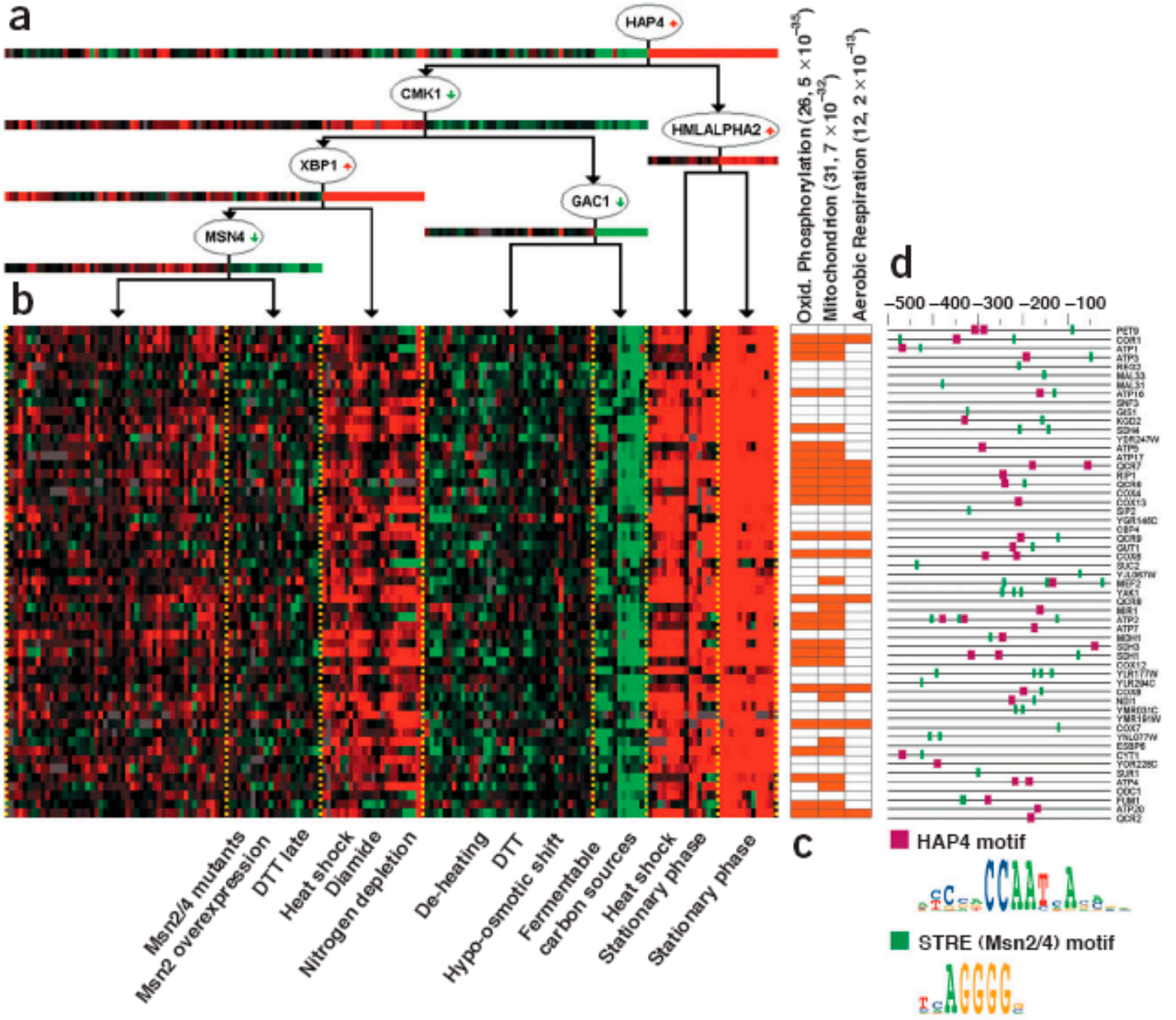
Application of Module networks to yeast expression data

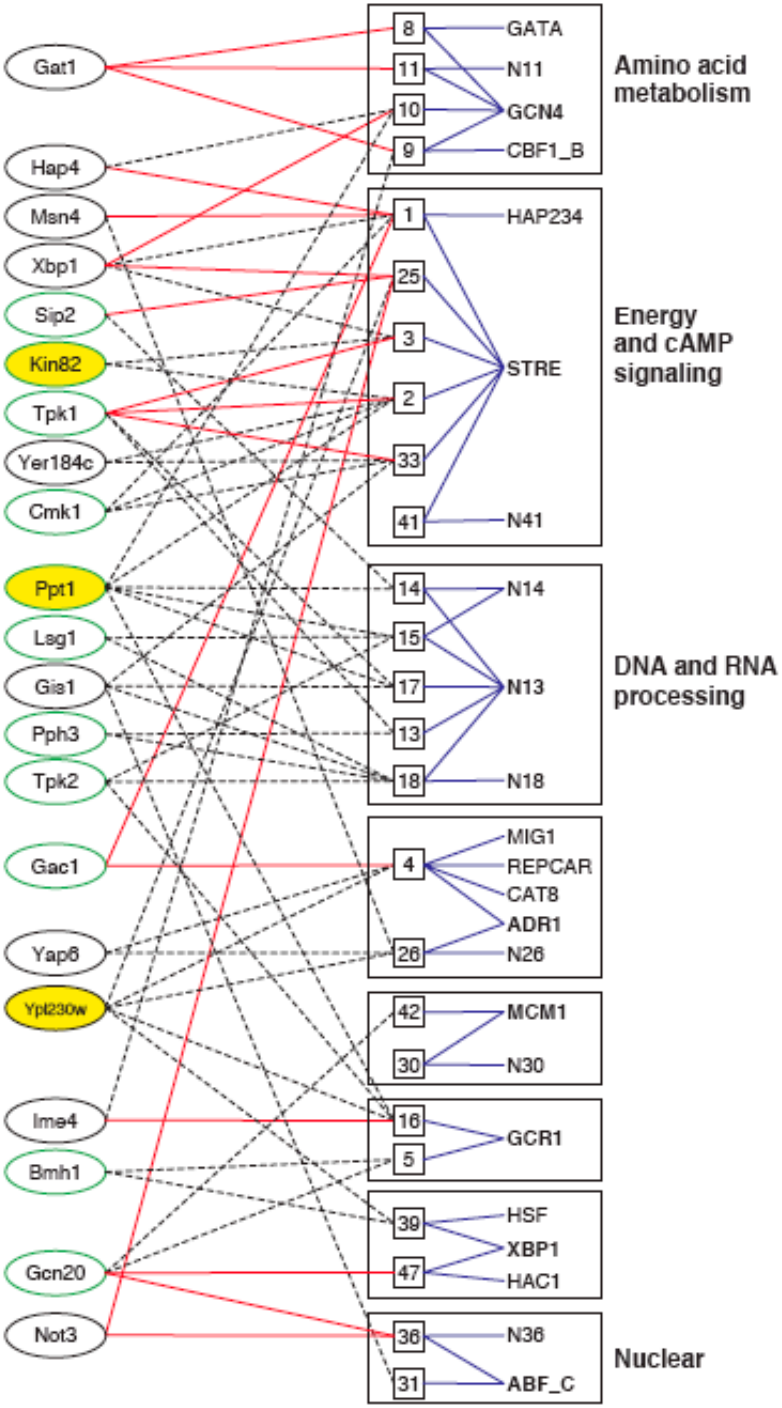


Segal et al, Regev, Pe'er, Gasch, Nature Genetics 2003

The Respiration and Carbon Module

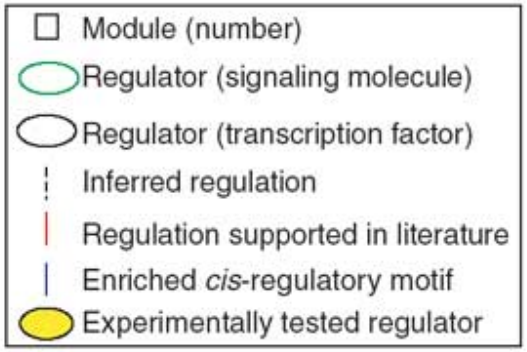
Regression tree representing rules of regulation





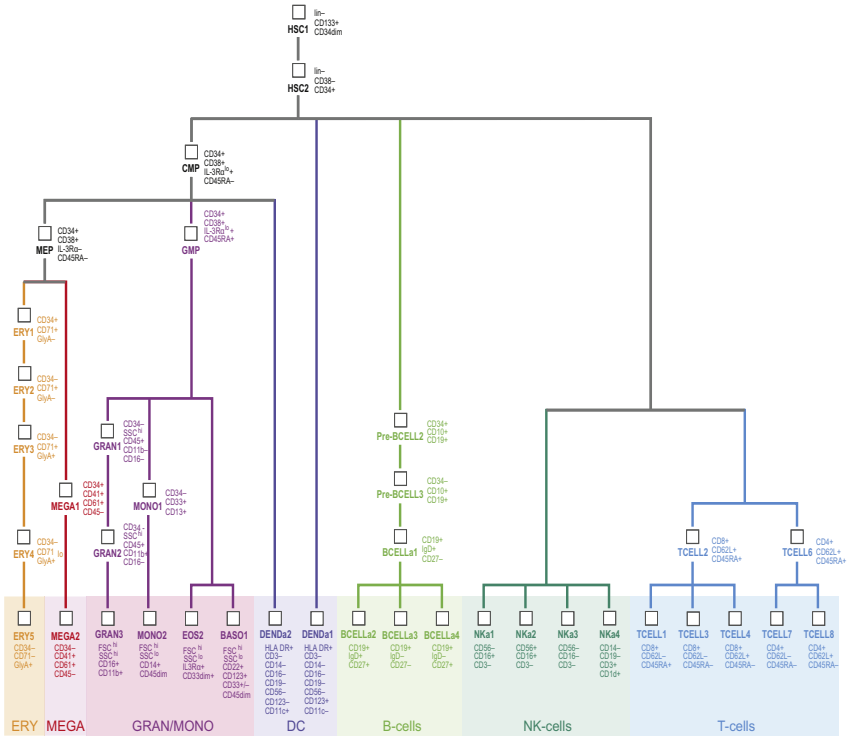
Global View of Modules

- modules for common processes often share common
  - regulators
  - binding site motifs



Application of Module networks to mammalian data

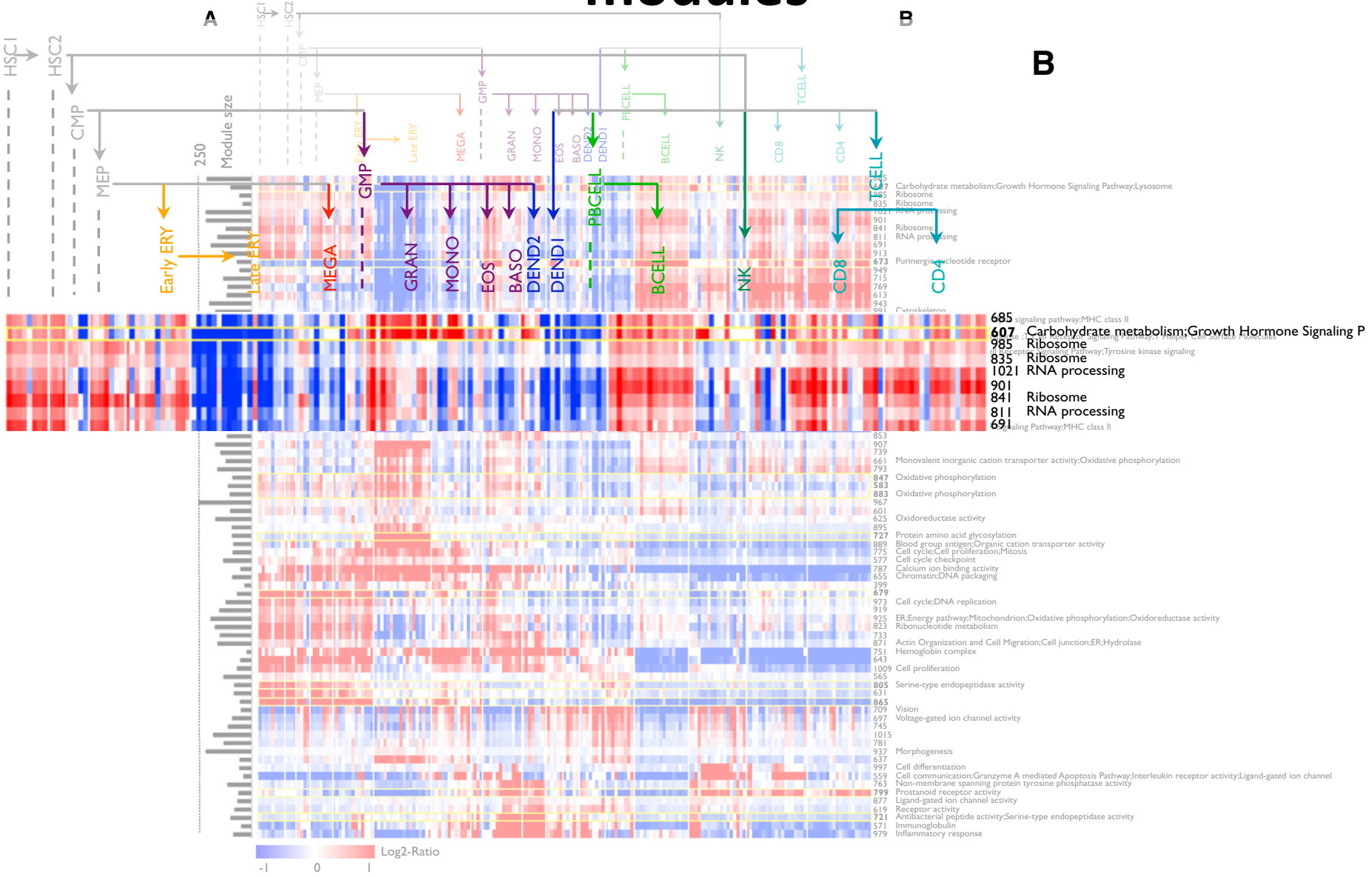
- Module networks have been applied to mammalian systems as well
- We will look at a case-study in the human blood cell lineage
- Dataset
  - Genome-wide expression levels in 38 hematopoietic cell types (211 samples)
  - 523 candidate regulators (Transcription factors)



Human hematopoetic lineage

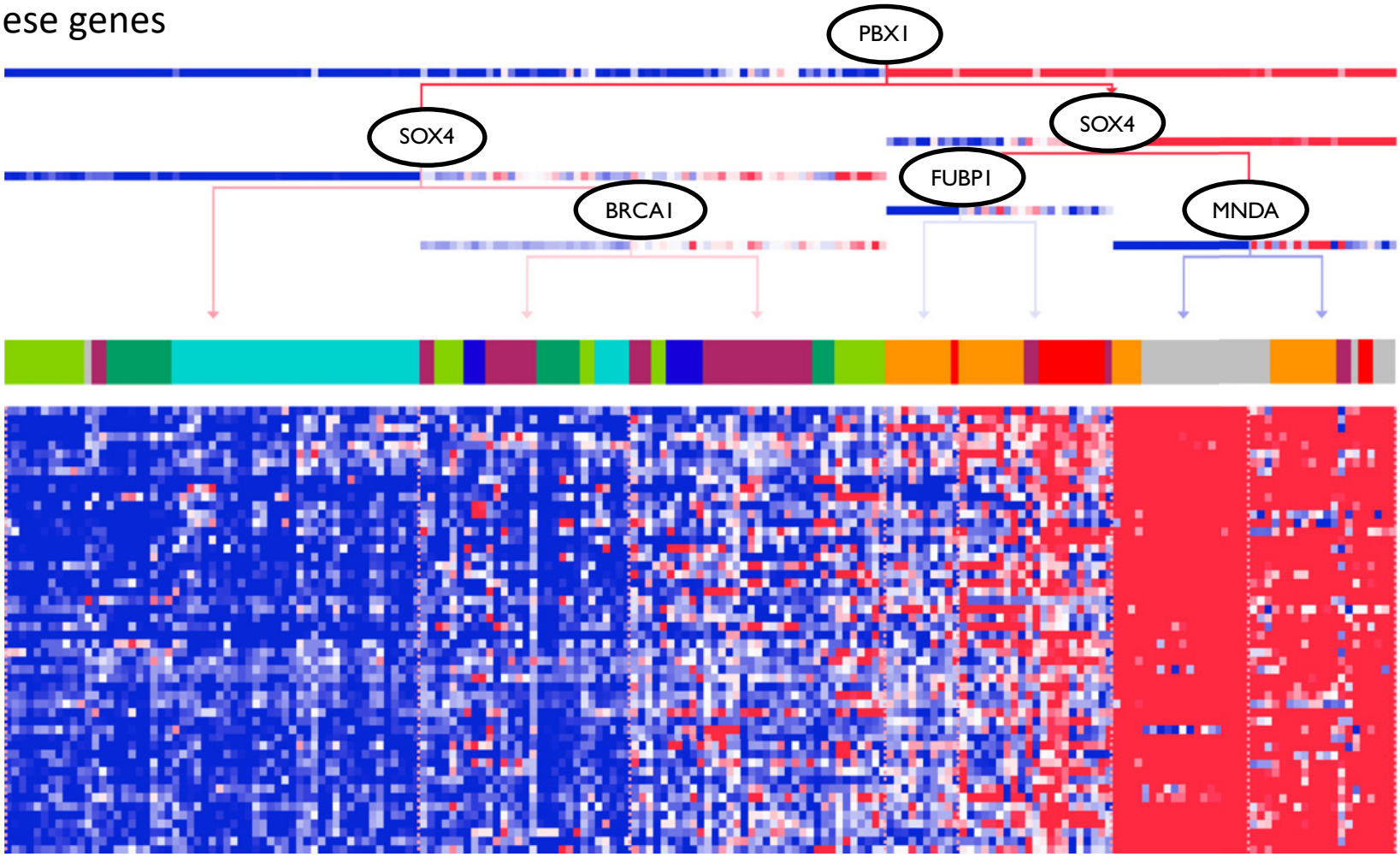
Novershtern et al., Cell 2011

Expression profiles of 80 transcriptional modules



An HSCs, MEPs, and Early Erythroid-Induced Module

PBX1, SOX4 need to be high and MNDA need to be low for the highest expression of these genes



Other key points from this analysis

- Many novel regulators associated with the hematopoietic lineage
- Several regulators were validated based on shRNA and ChIP-seq analysis

Extensions to module networks

- Physical module networks
  - Novershtern et al., Bioinformatics 2011
- Integrating sequence variants with expression modules
  - Lee et al., PLOS Genetics 2009
- Combining module networks with per-gene methods
  - Roy et al., PLOS computational biology 2013

Limitations with Bayesian networks

- Cannot model cyclic dependencies
- In practice have not been shown to be better than dependency networks
  - However, most of the evaluation has been done on structure not parameters
- Directionality is often not associated with causality
  - Too many hidden variables in biological systems

Take away points

- Network inference from expression provides a promising approach to identify cellular networks
- Graphical models are one representation of networks that have a probabilistic and graphical component
  - Network inference naturally translates to learning problems in these models
- Bayesian networks were among the first type of PGMs for representing networks
- Applying Bayesian networks to expression data required several additional considerations
  - Too few samples: Sparse candidates, Module networks
  - Too many parents: Sparse candidates
  - Imposing modularity: Module networks

Plan for next lectures

- Gaussian graphical models
- Dependency networks
  - GENIE3

References

- Koller & Friedman, Probabilistic Graphical Models. Principles and techniques.
- Kim, Harold D., Tal Shay, Erin K. O'Shea, and Aviv Regev. "Transcriptional Regulatory Circuits: Predicting Numbers from Alphabets." *Science* 325 (July 2009): 429-432.
- De Smet, Riet and Kathleen Marchal. "Advantages and limitations of current network inference methods.." *Nature reviews. Microbiology* 8 (October 2010): 717-729.
- Markowetz, Florian and Rainer Spang. "Inferring cellular networks-a review." *BMC bioinformatics* 8 Suppl 6 (2007): S5+.
- N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601-620, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1089/106652700750050961
- E. Segal, D. Pe'er, A. Regev, D. Koller, and N. Friedman, "Learning module networks," *Journal of Machine Learning Research*, vol. 6, pp. 557-588, Apr. 2005. [Online]. Available: http://www.jmlr.org/papers/volume6/segal05a/segal05a.pdf
- E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nature Genetics*, vol. 34, no. 2, pp. 166-176, May 2003. [Online]. Available: http://dx.doi.org/10.1038/ng1165
- N. Novershtern, et al., "Densely interconnected transcriptional circuits control cell states in human hematopoiesis." *Cell*, vol. 144, no. 2, pp. 296-309, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.cell.2011.01.004