Module Networks

BMI/CS 776

www.biostat.wisc.edu/bmi776/

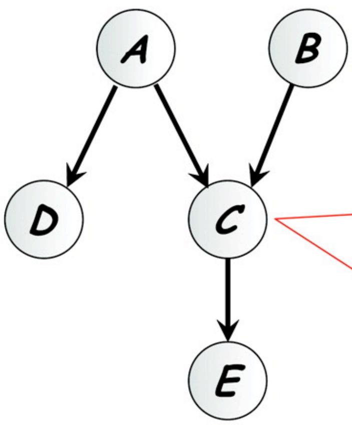
Mark Craven

craven@biostat.wisc.edu

Spring 2009

Bayesian Networks

A



C

$P(C \mid A, B)$

A	B	0	1
0	0	0.9	0.1
0	1	0.2	0.8
1	0	0.9	0.1
1	1	0.01	0.99

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

Figure from Friedman, *Science*, 303:799 – 805, 2004.

Bayesian Networks

- a BN is a Directed Acyclic Graph (DAG) in which
  - the nodes denote random variables
  - each node *X* has a *conditional probability distribution* (CPD) representing P(*X* | *Parents*(*X*))
- the intuitive meaning of an arc from *X* to *Y* is that *X* *directly influences Y*
- formally: each variable *X* is independent of its non-descendants given its parents
- a BN provides a *factored* representation of the joint probability distribution

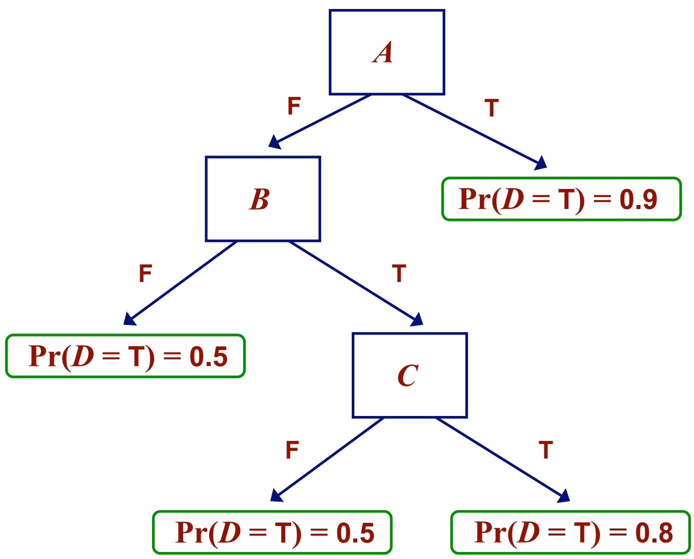
Representing CPDs for Discrete Variables

- CPDs can be represented using tables or trees
- consider the following case with Boolean variables *A*, *B*, *C*, *D*

Pr(*D* | *A*, *B*, *C*)

A	*B*	*C*	T	F
T	T	T	0.9	0.1
T	T	F	0.9	0.1
T	F	T	0.9	0.1
T	F	F	0.9	0.1
F	T	T	0.8	0.2
F	T	F	0.5	0.5
F	F	T	0.5	0.5
F	F	F	0.5	0.5

Pr(*D* | *A*, *B*, *C*)



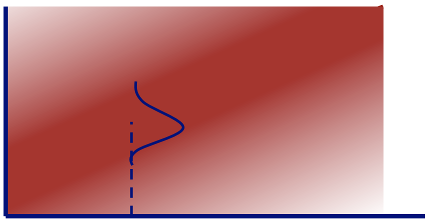
Representing CPDs for Continuous Variables

- we can also model the distribution of continuous variables in Bayesian networks
- one approach: *linear Gaussian models*

$$\Pr(X \mid u_1, \ldots, u_k) \sim N(a_0 + \sum_i a_i \times u_i,\ \sigma^2)$$

- X normally distributed around a mean that depends linearly on values of its parents u_i

$\Pr(X \mid u_1)$



u_1

BN Architectures

unconstrained acyclic network

two-level network: parents must be from a defined set

module network

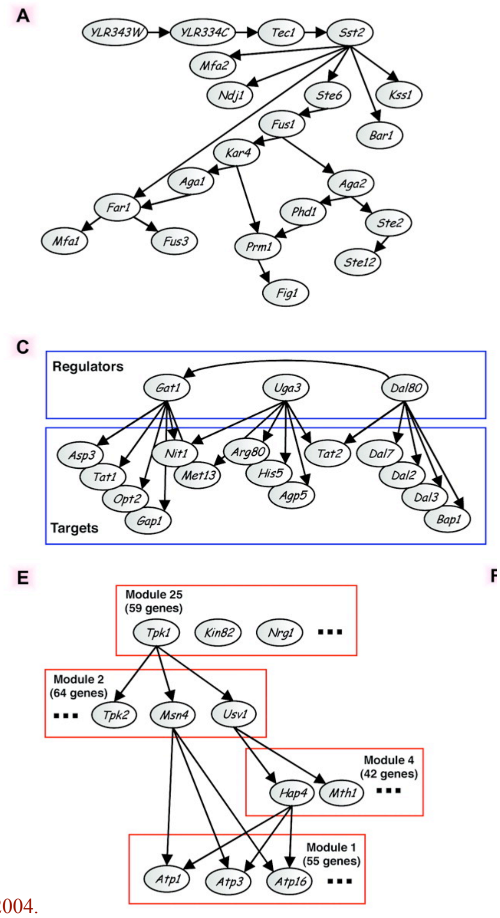


Figure from Friedman, *Science*, 303:799 – 805, 2004.

Module Networks Motivation

- *sets* of variables often have the same behavior
- consider this simple stock example

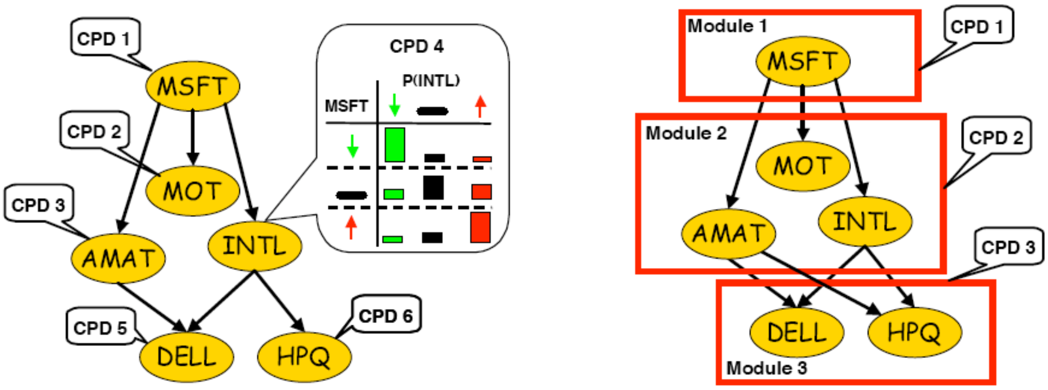


Figure from Segal et al., *UAI, 2003*.

- we can group variables into modules, have the members of a module share the same CPD

Module Networks

- a module network is defined by
  - a specified number of modules
  - an assignment of each variable to a module
  - a shared CPD for the variables in each module

- the learning task thus entails*
  - determining the assignment of variables to modules
  - inducing a CPD for each module

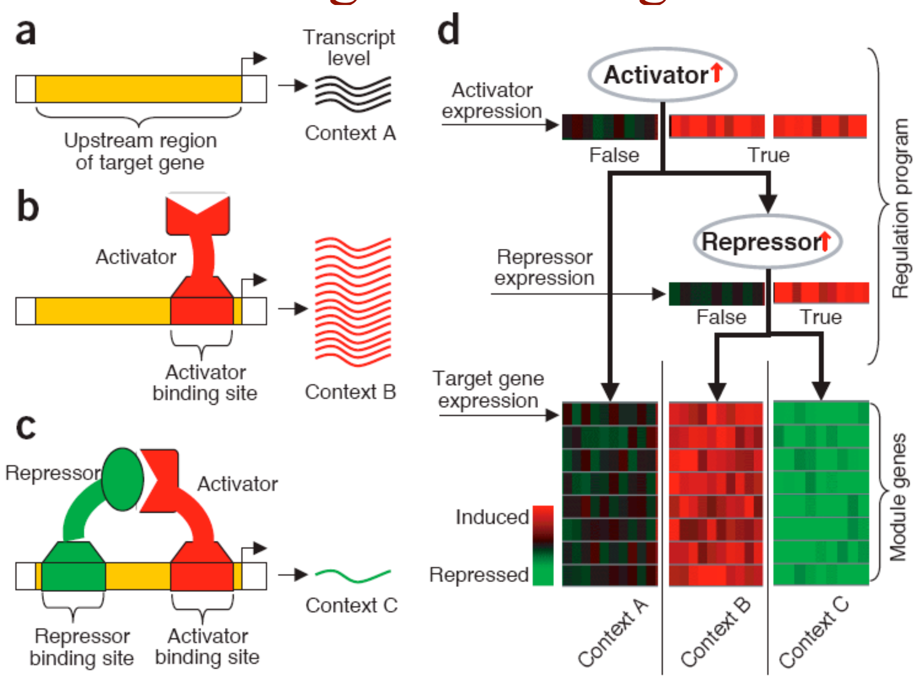
*assuming we're given the number of modules

Module Networks: Identifying Regulatory Modules and their Condition-Specific Regulators from Gene Expression Data. E. Segal et al., *Nature Genetics* 34(2):166-176, 2003

- given:
  - gene expression data set

- their method identifies:
  - sets of genes that are co-expressed (assignment to modules)
  - a "program" that explains expression profile for each set of genes (CPD for each module)

A Regulation Program



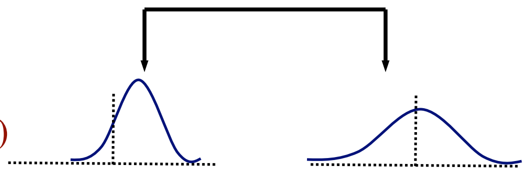
- suppose we have a set of (8) genes that all have in their upstream regions the same activator/repressor binding sites

Regulation Programs as CPDs

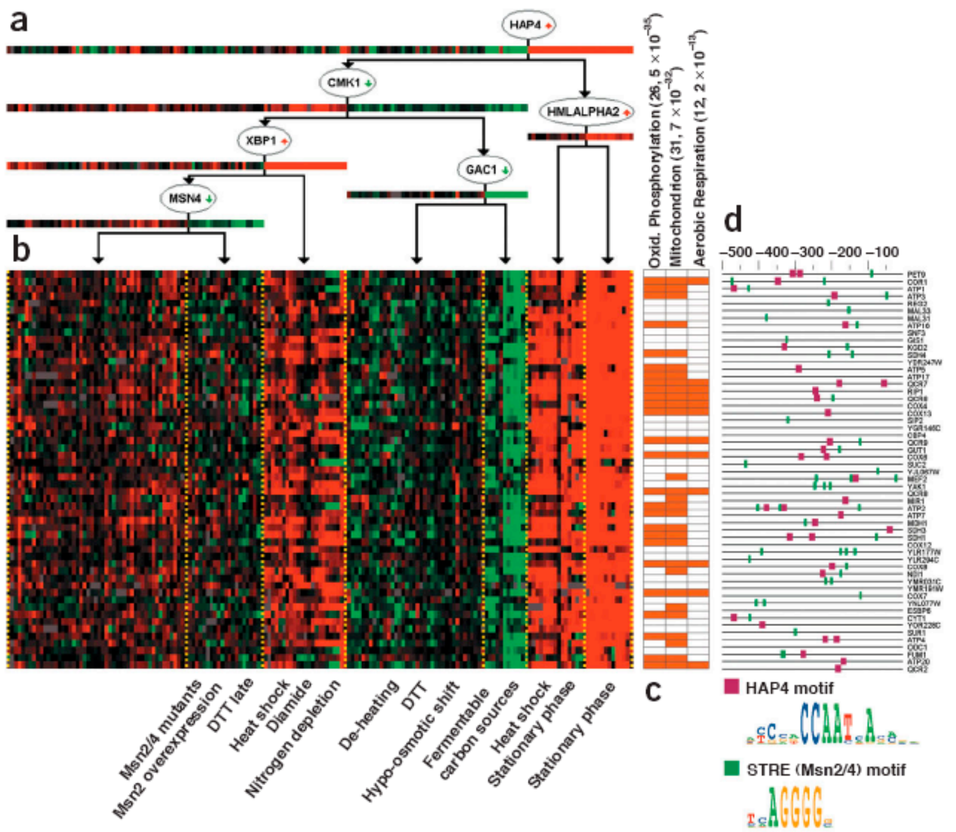
- each of these regulation programs is actually a CPD represented using a tree
  - internal nodes are tests on continuous variables

  - leaves contain conditional distributions for the genes in the module, represented by Gaussians

P(expression| leaf)

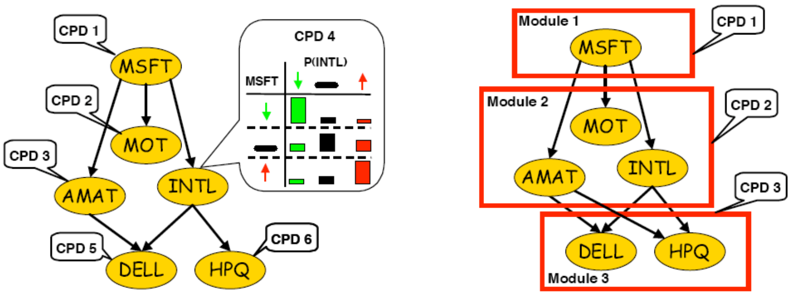


The Respiration and Carbon Module



Tree CPDs in Gene Module Networks

- the parents of each module can be other modules



- in this study, Segal et al. limit parents to a set of *candidate regulator genes* (genes known to be transcription factors and signaling components)

Module Network Learning Procedure

given: expression profile for each gene, set of candidate regulator genes

initialize module assignments by clustering expression profiles
repeat until convergence
    structure search step:
        for each module
            learn a CPD tree using splits on candidate regulators

    module assignment step:
        repeat until convergence
            for each gene
                find the module that best explains it
                move the gene to this module
                update Gaussians at leaves

Structure Search Step

- the method for the *structure search* step is very similar to the general decision-tree procedure
  - splits are on genes in the candidate regulator set
  - leaves represent distributions over continuous values

- the name for this step is somewhat misleading
  - it does involve learning structure – selecting parents for variables in the module
  - it also involves learning the parameters of the Gaussians at the leaves
  - the *module assignment* step heavily influences the structure

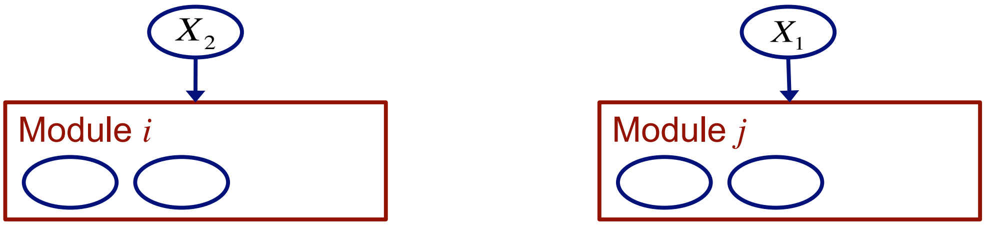
Module Assignment Step

- Can we independently assign each variable to its best module?
  - No – we might get cycles in the graph.
  - the score for a module depends on all of the genes in the module

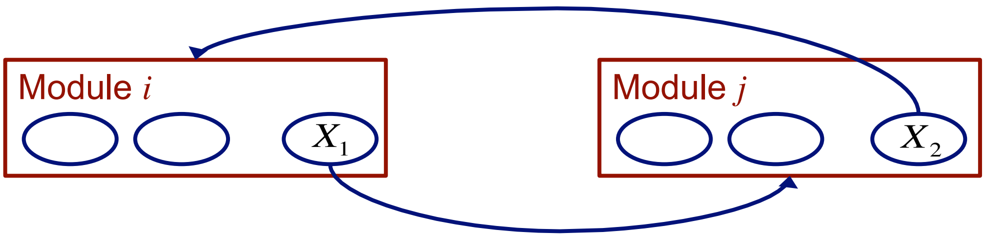
- therefore we use a sequential update method (moving one gene at a time)
  - can ensure that each change is a legal assignment that improves the score

Module Assignment Step

- suppose we have the current (partial) structure, and we independently re-assign X_1 to Module i and X_2 to Module j



- then we have a cycle



Module Assignment Step

- in order to decide a candidate re-assignment, we need a valid structure

$$score(S, A : D) = P(A)P(S \mid A)P(D \mid S, A)$$

S: the dependency structure
A: the assignment of genes to modules
D: the data (gene expression observations)

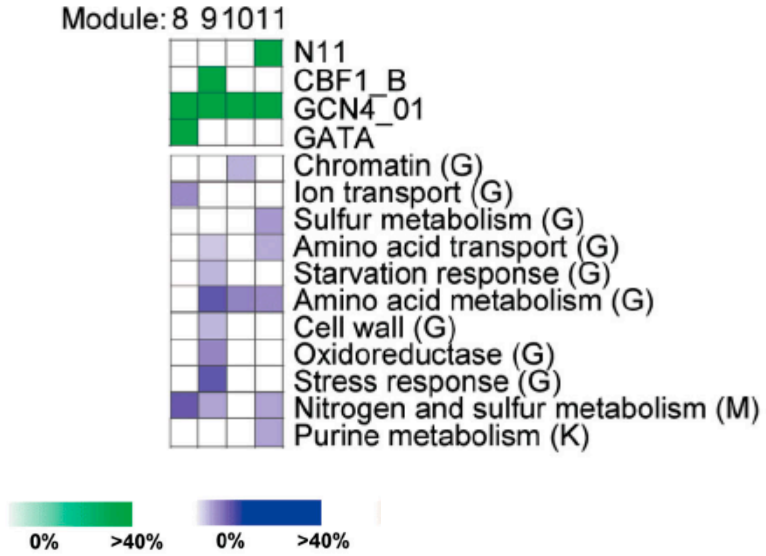
- reassign gene to another module if doing so improves score

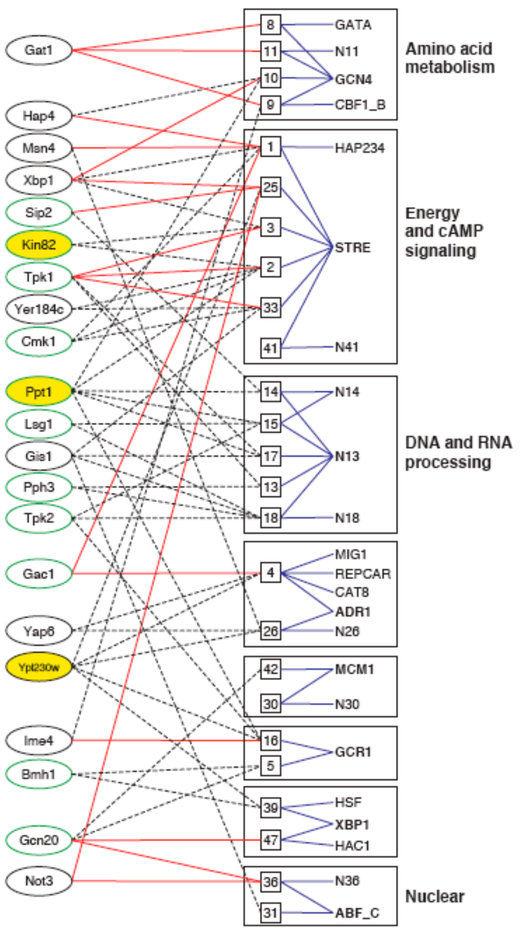
- we can efficiently score local changes because the scoring function is modular

$$score(S, A : D) = \sum_j score_{M_j}(\mathrm{Pa}_{M_j}, A_{M_j} : D)$$

Empirical Evaluation

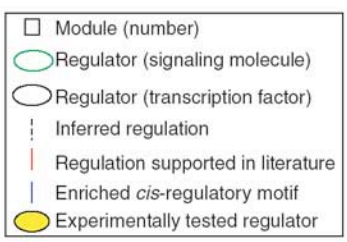
- many modules are *enriched* for
  - binding sites for associated regulators
  - common gene annotations





Global View of Modules

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



Comments on Module Networks

- module networks exploit the fact that many variables (genes) are determined by the same set of variables

- this application exploits the fact that we may have background knowledge about the variables that can be parents of others (the candidate regulators)

- the learning procedure is like EM, but <u>hard</u> decisions are made (each gene is completely assigned to a module)