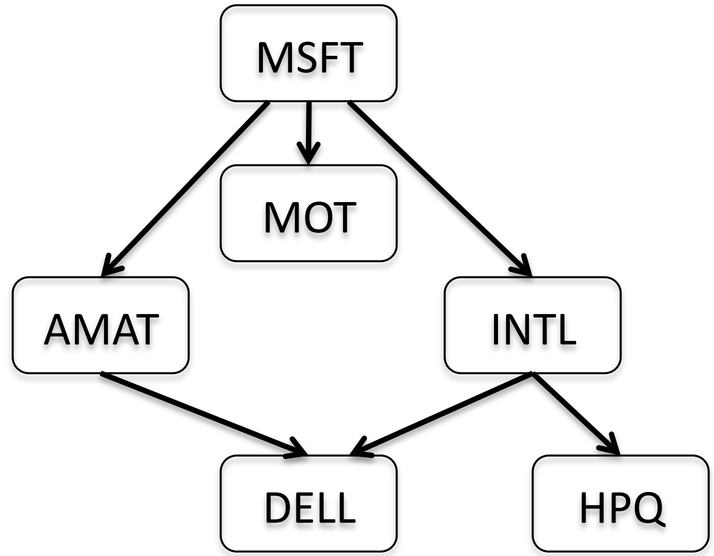
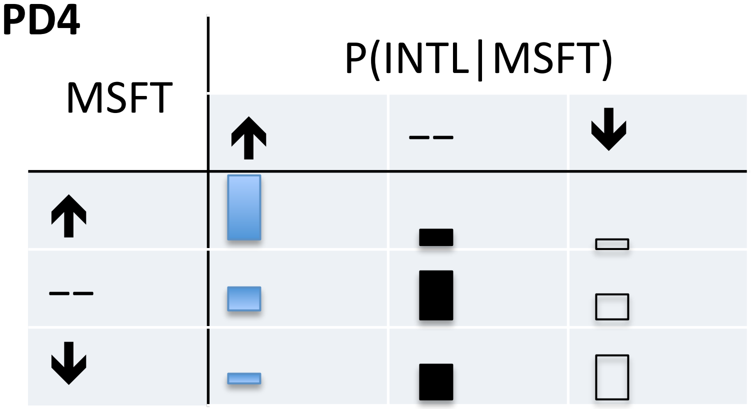
Module network

Bayesian network: a toy example

Variables X: STOCKS (space: {↑, --, ↓})
MSFT: microsoft
AMAT: Applied materials
INTL: Intel
MOT: Motorola
DELL: Dell
HPQ: Hewlett-Packard



CPD4



Conditional probability distribution (CPD)

- One for each variable:
  - CPD1: MSFT; CPD2: MOT
  - CPD3: AMAT; CPD4: INTL
  - CPD5: DELL; CPD6: HPQ
- BN defines the dependency relations, e.g.
  - CPD4: P(INTL) = P(INTL|MSFT)

Issues with Bayesian network

- Large search space for problems of many parameters variables (e.g. the *gene regulatory network modeling problem*)
  - Each CPD (parameters) to be learned from data for each variable
  - The search space of the putative network is even larger!
    - Training data is not sufficient to determine the optimal model structure
- Results are hard to be interpreted
  - a large network of thousands of nodes
    - Gene regulatory network

Module networks

- **Segal et al. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data.** Nat Genet. 2003 Jun;34(2):166-76.
  - Identifies modules of **coregulated** genes, their regulators and the conditions under which regulation occurs, generating hypotheses in the form "regulator X regulates module Y under conditions W".
- **Leveraging models of cell regulation and GWAS data in integrative network-based association studies.** Nature Genetics 44, 841–847 (2012)

Modular biological networks

- **Evolution of Complex Modular Biological Networks**
  - PLoS Comput Biol 4(2): e23. 2008
  - "One of the main contributors to the robustness and evolvability of biological networks is believed to be their **modularity** of function, with **modules defined as sets of genes that are strongly interconnected but whose function is separable from those of other modules.** "
- **Learning biological networks: from modules to dynamics**
  - Nature Chemical Biology 4, 658 - 664 (2008)

Modules on Bayesian network

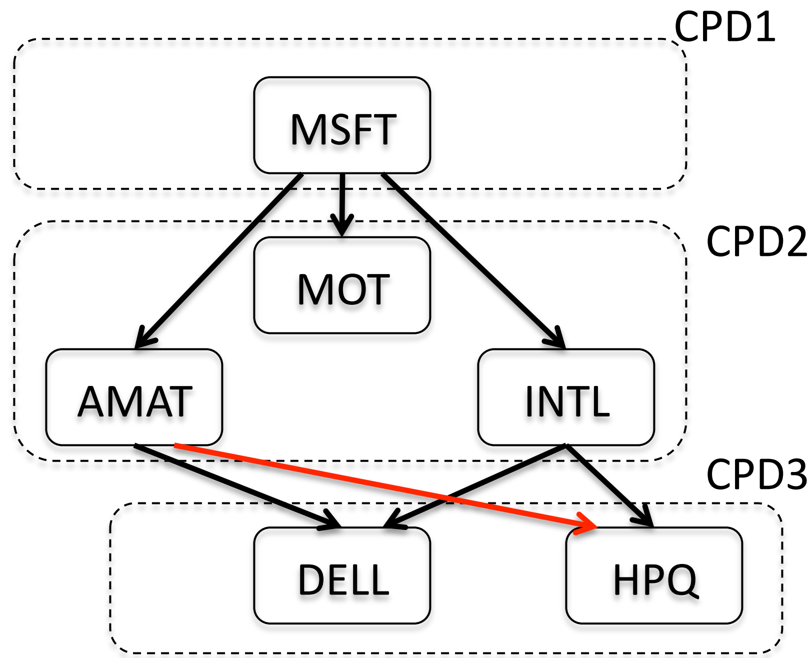
Modules: a set of variables with the same dependence (the same set of parents and the same CPD)

A *module set C:* $M_1, \ldots, M_K$
Val(M_j)*: possible values of* M_j

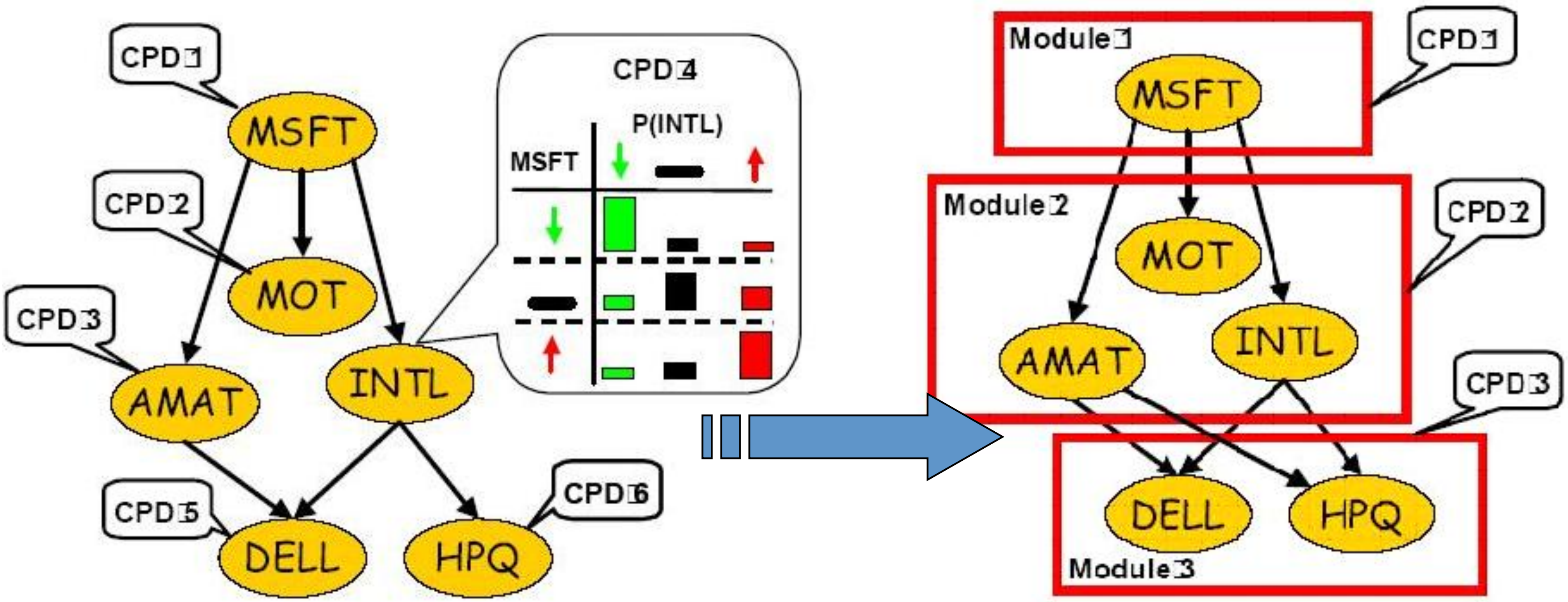
for each module M_j,
• a set of parents Pa_{Mj} ;
• a conditional probability distribution template $P(M_j | Pa_{Mj})$

a module assignment function A: assigns each variable X_i *to* one of the *K modules*
A(*MSFT) = 1, A(MOT) = 2, etc*

Unrolling a Bayesian network with a well-defined distribution, i.e. the resulting network must be acyclic.



From Bayesian To Module



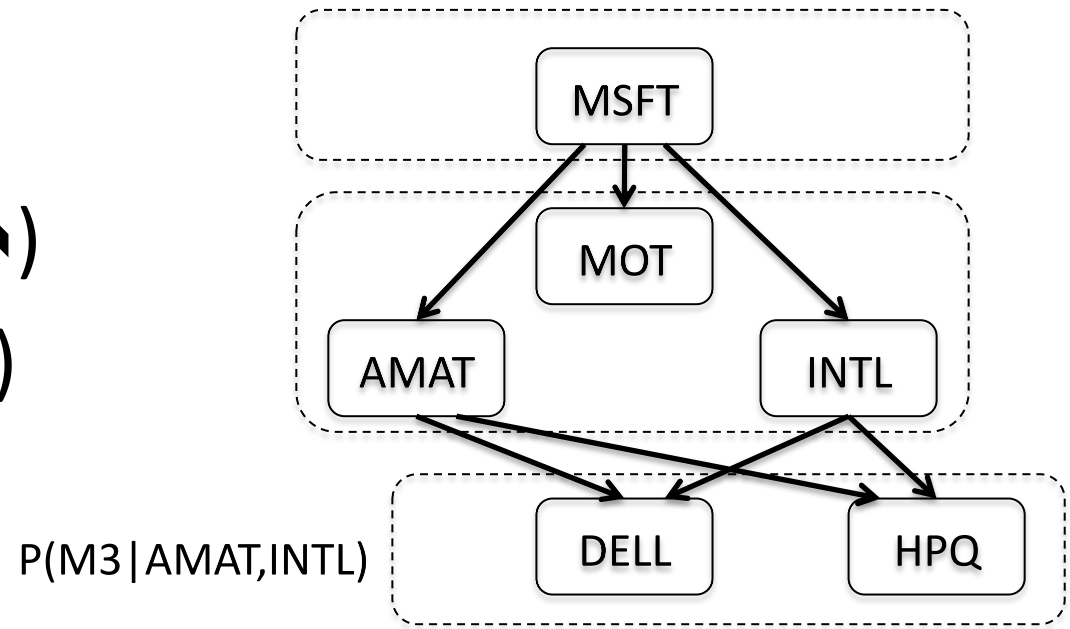
(a) Bayesian network (b) Module network

Module networks

- Module network template: $T = (S, \Theta)$, *s.t. for each module* M_j
  - *S: a set of parents* $\{Pa_{M_j}$ *in X for each* $M_j\}$
  - *Θ: a conditional probability distribution* $P(M_j \mid Pa_{M_j})$
- T → Module graph: G_M
- Module network M=(C, T, A)
  - *C: set of variables*
  - *T: module network template*
  - *A: module assignment function*
  - B_M*: underlined Bayesian network*
  - G_M is acyclic <-> B_M is acyclic

Reasoning: same as BN

- P(DELL🡅)
- P(DELL🡅|MSFT🡅)
- P(DELL🡅|MOT🡅)



Learning structures of module networks

- Likelihood score

$$L(M:D)=\prod_{j=1}^{K} L_j\left(Pa_{M_j}, A(X^j), \theta_{M_j \mid Pa_{M_j}} : D\right)$$

  - For each variable j, X^j: module assignment
  - $\Theta_{Mj|Pa_{Mj}}$: parameters of $P(M_j|Pa_{Mj})$
- Bayesian score (and priors)

$$\max_G P(G \mid D) \propto \max_G P(D \mid G)P(G) = \int_{\theta_G} P(D \mid \theta_G, G)P(\theta_G \mid G)d\theta_G$$

$$\max_{S,A,\theta} P(S,A,\theta \mid D) \propto \max_{S,A,\theta} P(D \mid S,A,\theta)P(S,A,\theta) = \sum_{\theta_S} P(D \mid \theta_S, S, A)P(\theta_S \mid S,A)P(S,A)$$

  - Global modularity: $P(\theta_S \mid S,A) = P(\theta_S \mid S)$ Only template is important for prior!

$$P(S,A)=\rho(S)\kappa(A)C(A,S)$$

C(A,S) is a constraint indicator function that is equal to 1 if the combination of structure and assignment is a legal one (i.e., the module graph induced by the assignment A and structure S is acyclic)

Assumptions

- *Parameter independence, parameter modularity, and structure modularity* are the natural analogues of standard assumptions in Bayesian network learning.
- *Parameter independence* implies that P(Θ | S, A) is a product of terms that parallels the decomposition of the likelihood, with one prior term per local likelihood term Lj.
- *Parameter modularity* states that the prior for the parameters of a module Mj depends only on the choice of parents for Mj and not on other aspects of the structure.
- *Structure modularity* implies that the prior over the structure S is a product of terms, one per each module.

Assumptions - Explainations

- These two assumptions are new to module networks.

- **Assignment independence:** makes the priors on the parents and parameters of a module independent of the exact set of variables assigned to the module.

- **Assignment modularity:** implies that the prior on A is proportional to a product of local terms, one corresponding to each module.

- Thus, the reassignment of one variable from one module Mi to another Mj does not change our preferences on the assignment of variables in modules other than i; j.

Learning structures of module networks

- Assuming global modularity, etc

$$Score(S,A,\theta \mid D) = \sum_j score_{M_j}\left(Pa_{M_j}, A(X^j), \theta_{M_j \mid Pa_{M_j}} : D\right)$$

  – Structure search step
    - learns the structure S (and θ), assuming that A is fixed
      – Similar as the learning of a Bayesian network structure (on a smaller set of nodes)
      – Update the dependency structure and parameters for each module (M_j) at a time
  – Module assignment search step
    - As clustering
    - Sequential update

Sketch of learning algorithm

Input:
D // Data set
K // Number of modules
Output:
M // A module network
Learn-Module-Network
$\mathcal{A}_0$ = cluster 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 $\mathbf{M} = (\mathcal{A}_t, \mathcal{S}_t)$

Converge to a local maximum

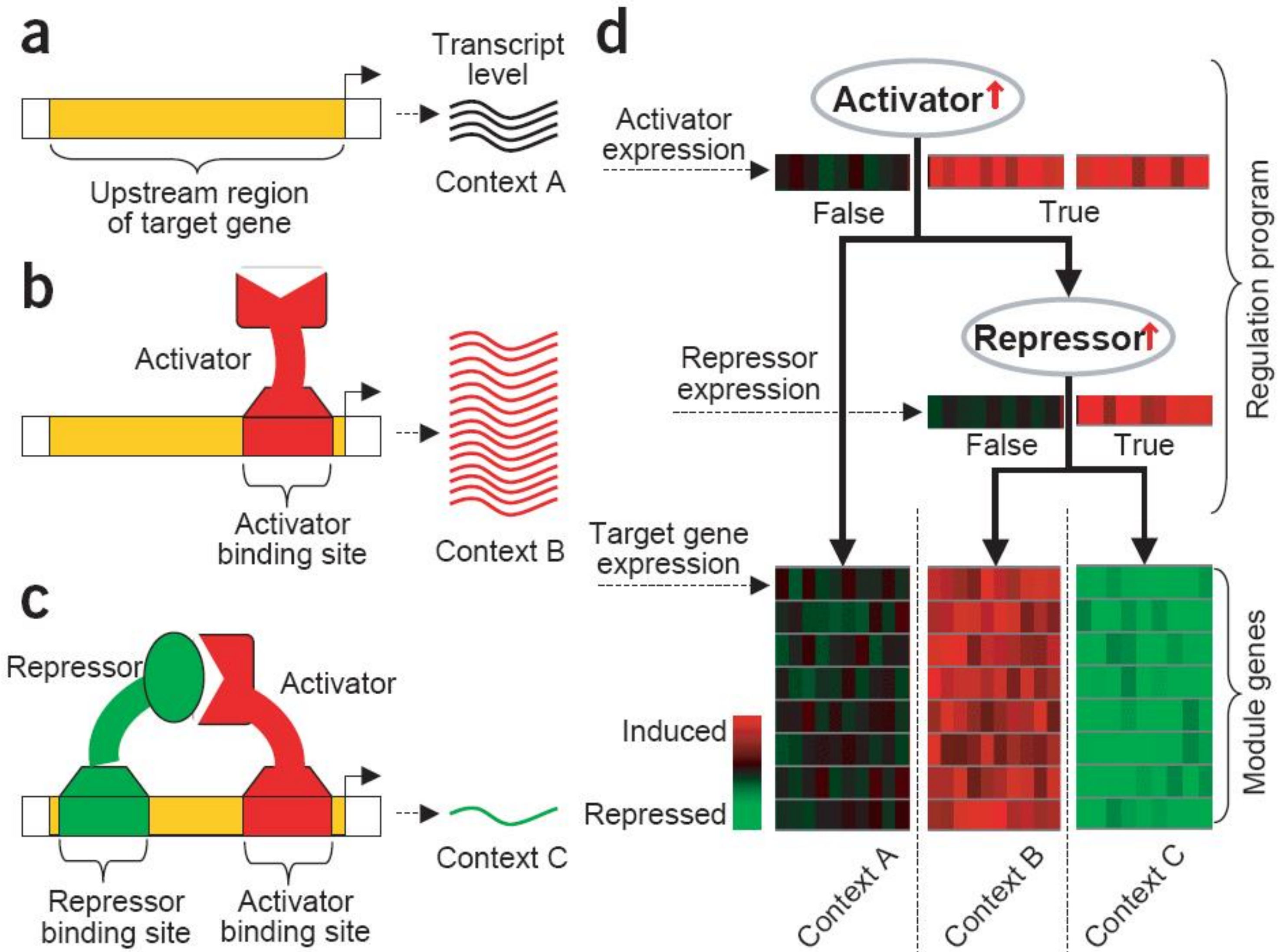
Sequential updates of assignment function

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 $\mathcal{A}$

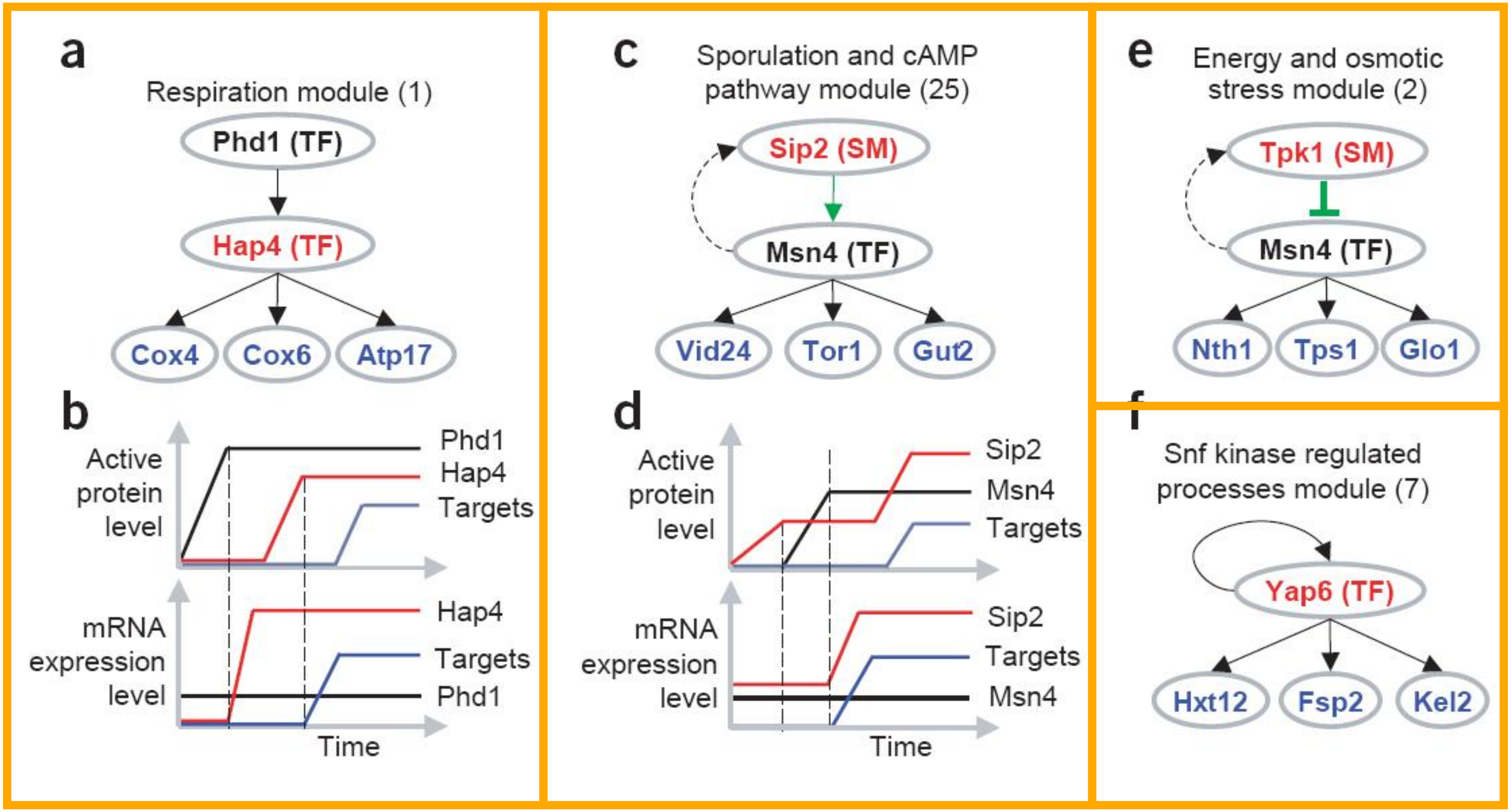
Gene regulatory network modeling problem

- Input: gene expression values on various conditions
  - Matrix with rows being genes and columns being conditions
  - Constraint: **a subset of genes that are regulators**
    - From domain knowledge, limit the search space
- Output: module network of gene expression
  - Modules: genes co-regulated
  - Parents: regulatory genes
  - Module network: pathways

Regulators

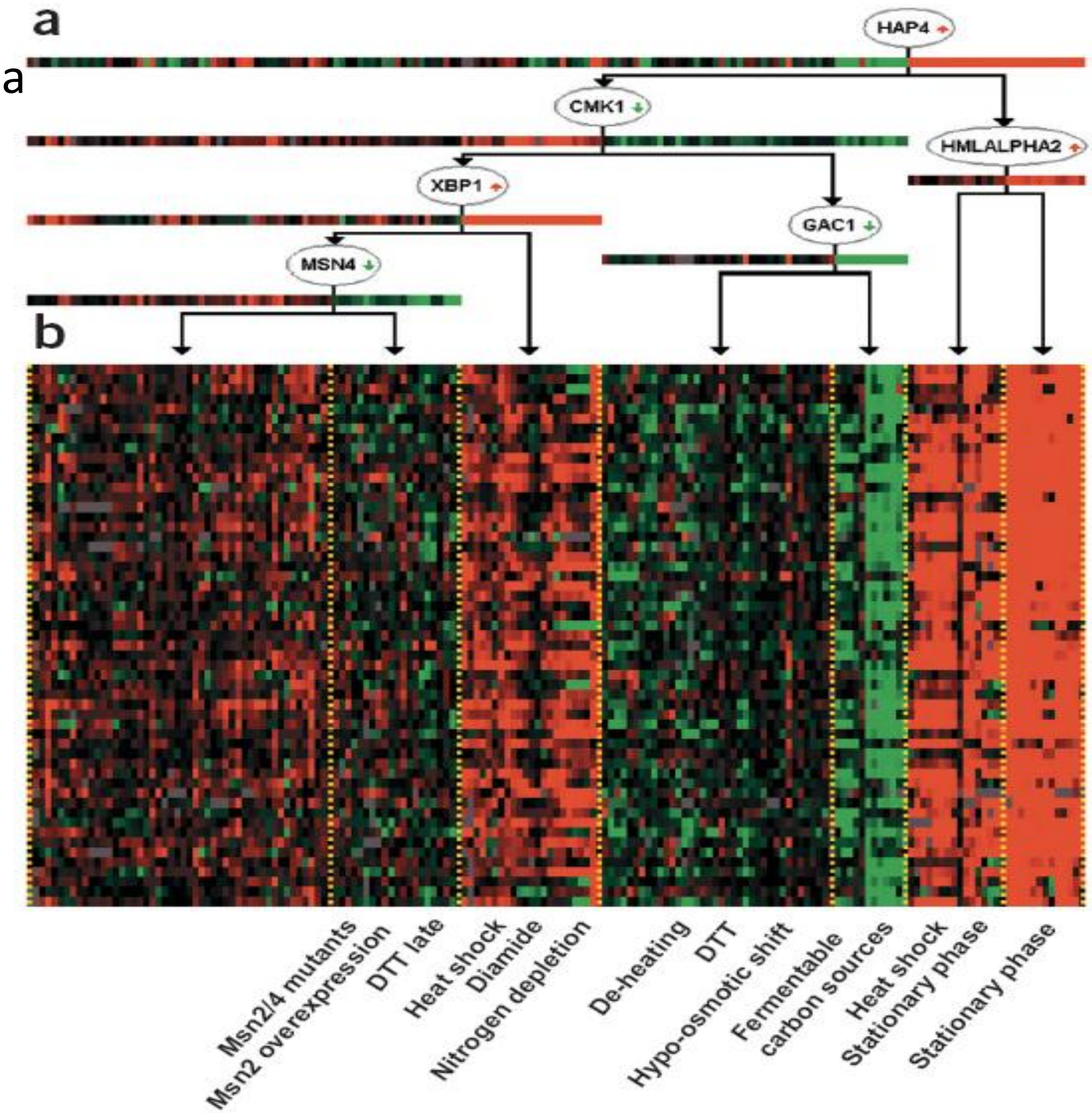

a
Transcript level
Upstream region of target gene
Context A
b
Activator
Activator binding site
Context B
c
Repressor
Activator
Repressor binding site
Activator binding site
Context C
d
Activator
Activator expression
False
True
Repressor
Repressor expression
False
True
Regulation program
Target gene expression
Induced
Repressed
Module genes
Context A
Context B
Context C

Regulation types



Regulators example

This is an example for a regulating module.



Gene Expression Data

- Expression data which measured the response of yeast to different stress conditions was used.

- The data consists of 6157 genes and 173 experiments.

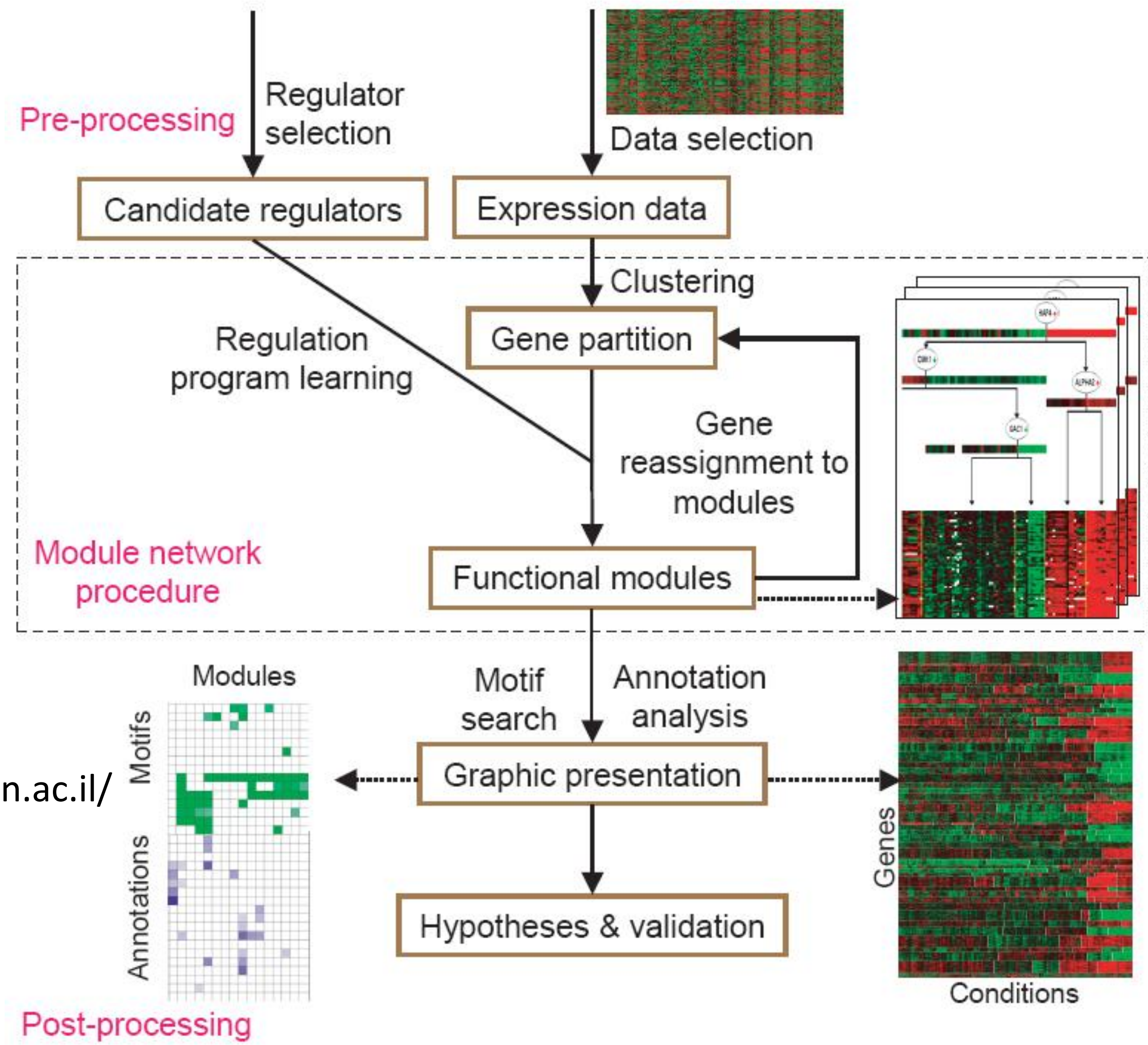
- 2355 genes that varied significantly in the data were selected and learned a module network over these genes.

- A Bayesian network was also learned over this data set.

Candidate regulators

- A set of 466 candidate regulators was compiled from SGD and YPD.
- Both transcriptional factors and signaling proteins that may have transcriptional impact.
- Also included genes described to be similar to such regulators.
- Excluded global regulators, whose regulation is not specific to a small set of genes or process.

Overview on Module Networks



Genomica (Segal lab)
http://genomica.weizmann.ac.il/