10-810 /02-710
Computational Genomics

Reconstructing regulatory networks: Module based algorithms

Motivation

High-level goal: Use high throughput data to discover patterns of combinatorial regulation and to understand how the activity of genes involved in related biological processes is coordinated and interconnected.

- Unlike methods that represent individual genes, module based methods group genes. This helps in:

  - Reducing dependence on individual measurements which could be noisy

  -Elevated statistical significance

  - Allows integration of different data sources

Goal: Discover structure and function of complex systems in the cell

- Identify the different components that are involved in the system.
- Determine how these components are connected.
- Assemble components into networks for different systems in the cell.

Today: Two methods

- Probabilistic relational modules (PRMs)

  - Factors activity is determined by their expression levels

- GRAM

  - Expression and binding data

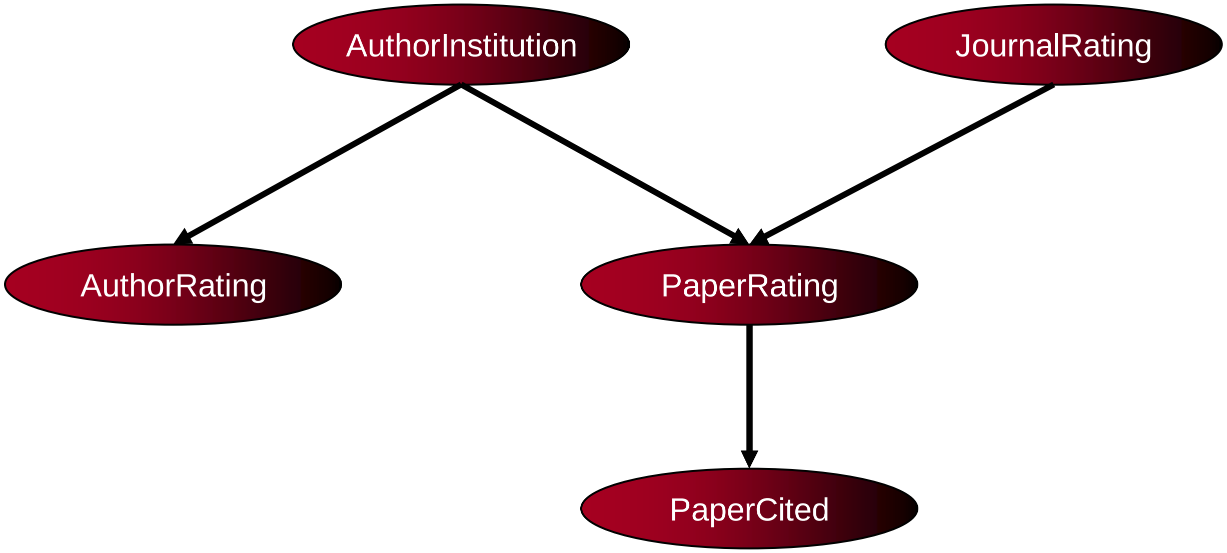
  - Factor activity is determined indirectly

Propositional Uncertainty

- To model uncertainty we would like to represent a probability distribution over possible worlds.
- To represent the full joint distribution we would need 2^{n}-1parameters (infeasible)
- Insight: the value of most propositions isn't affected by the value of most other propositions!
- More formally, some propositions are conditionally independent of each other given the value of other propositions

Bayesian Networks

- Use a directed acyclic graph to encode these independence assumptions



- This model encodes the assumption that each variable is independent of its non-descendents given its parents

Factorization

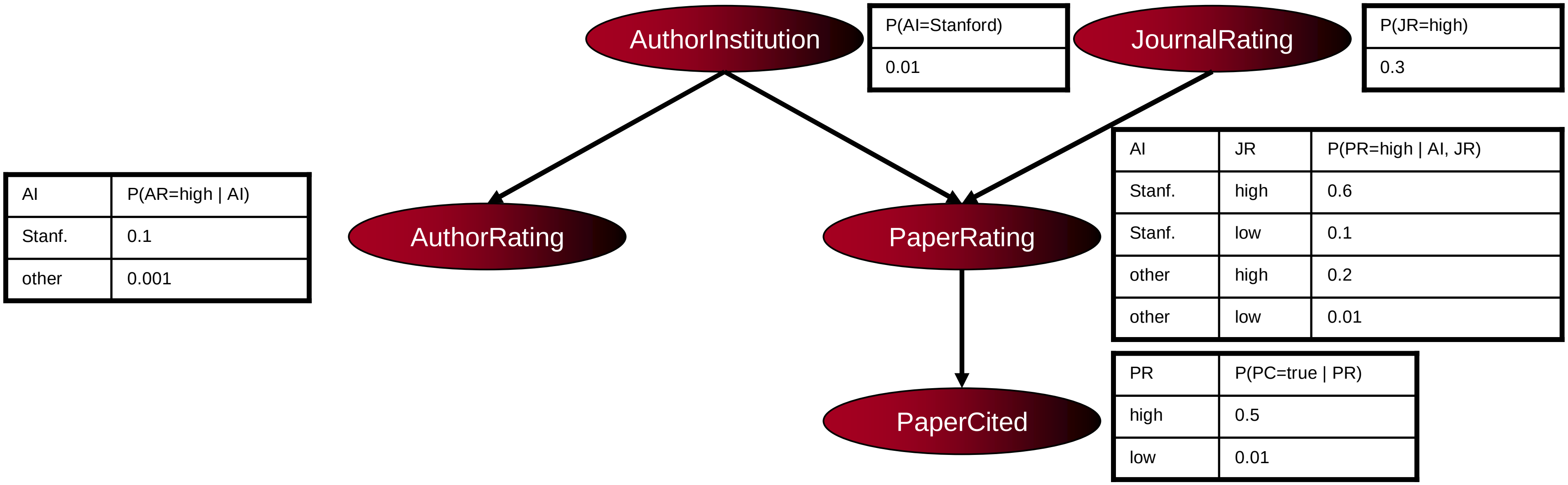
- If a BN encodes the true independence assumptions of a distribution, we can use a factored representation for the distribution:

$$\mathrm{P}(x_1,\ldots,x_n) = \prod_{i=1}^{n} \mathrm{P}(x_i \mid x_{i+1},\ldots,x_n)$$

$$= \prod_{i=1}^{n} \mathrm{P}(x_i \mid \mathrm{Pa}(x_i))$$

- To specify the full joint we need only the conditional probabilities of a variable given its parents

Bayesian Networks



P(AI=Stanford)
0.01

P(JR=high)
0.3

AI	P(AR=high | AI)
Stanf.	0.1
other	0.001

AI	JR	P(PR=high | AI, JR)
Stanf.	high	0.6
Stanf.	low	0.1
other	high	0.2
other	low	0.01

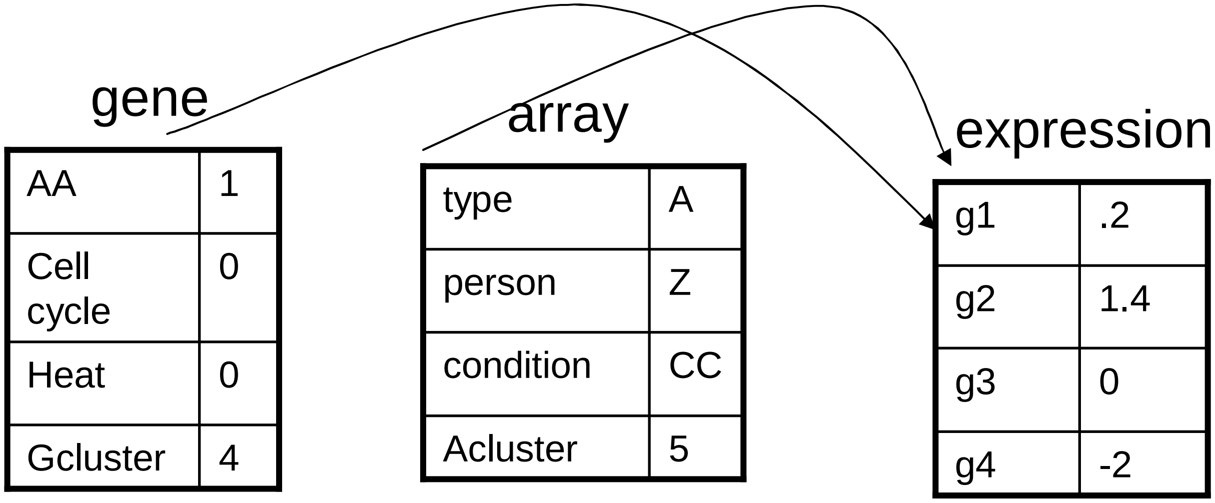
PR	P(PC=true | PR)
high	0.5
low	0.01

Bayes Net Shortcomings

- BNs lack the concept of an object
- Cannot represent general rules about the relations between multiple similar objects

Modeling Uncertainty

- More formally, we define:
  - A relational *skeleton*, S, to be a set of objects and relations between them (defined as reference slot values)
  - What are the differences in objects that we may see?
  - An *instance*, *I*, to be an assignment of values to attributes
- A PRM defines a probability distribution over possible completions *I* of a skeleton S



gene

AA	1
Cell cycle	0
Heat	0
Gcluster	4

array

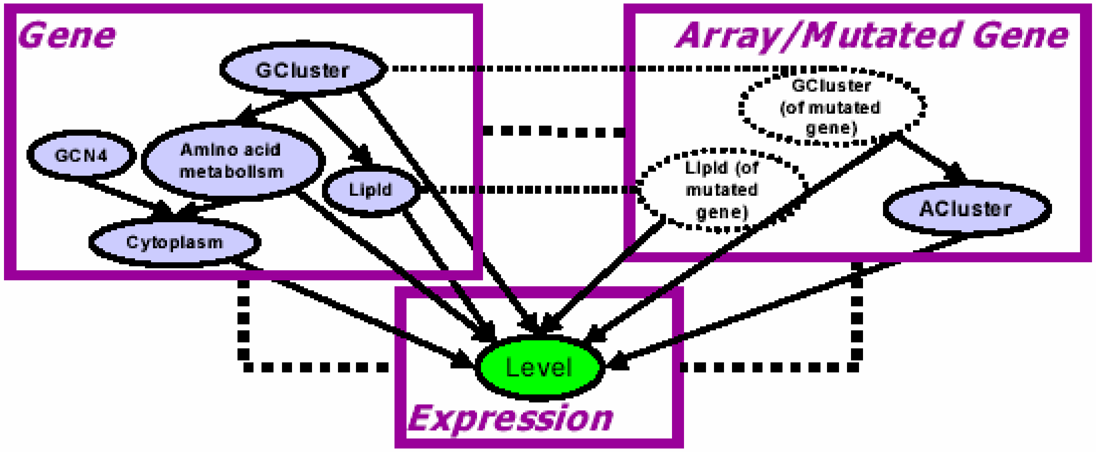
type	A
person	Z
condition	CC
Acluster	5

expression

g1	.2
g2	1.4
g3	0
g4	-2

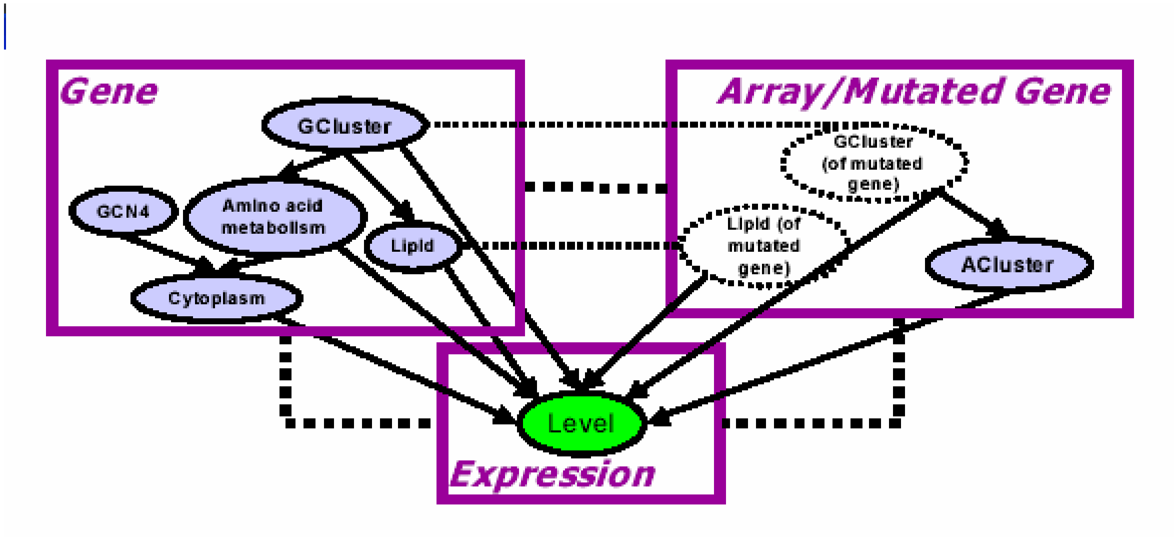
PRM Dependency Structure

- PRMs assume that the attribute values of objects are each influenced by only a few other attribute values (as in a BN)
- Thus we associate with each attribute *X*.*A* a set of *parents* Pa(*X*.*A*)
- These are *formal* parents; they will be instantiated differently for different objects
- These sets of parents (one for each attribute) define the *dependency structure S* of the PRM



Types of Parents

- We define two types of parents for *X.A*:
  - Another attribute *X.B* of the same class *X*
    - E.g., GO function could depend on *GCluster*
  - An attribute of a related object *X.B* where B is in a different class
    - E.g., *expression level* could depend on ACluster

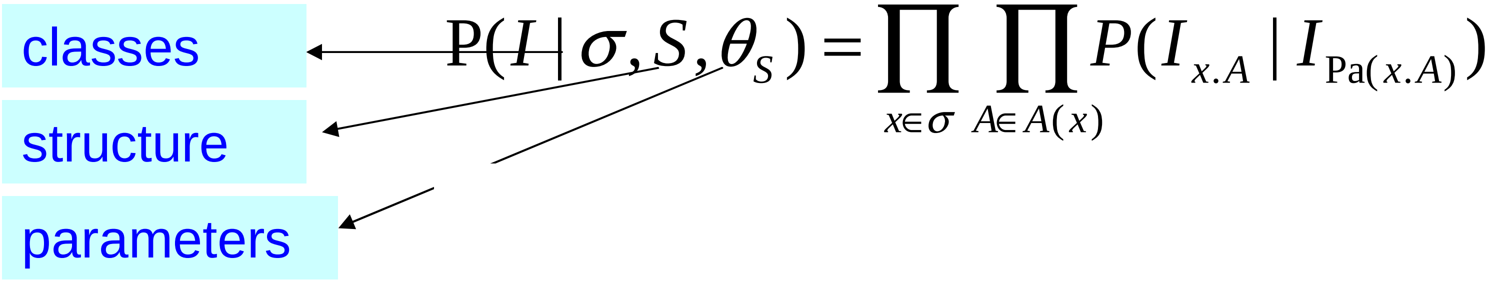


PRM Parameters

- As in a BN, for each attribute we define a conditional probability distribution (CPD) over the values of the attribute given the values of the parents
- More precisely, let U = Pa($X.A$) be the set of parents, V(U) be the possible values of U, and $u \subseteq$V(U) be some set of of values
- Then we can define a distribution P($X.A|u$)

Joint Probability Distribution

- Now we can use the following factored representation for the joint probability distribution over possible instances consistent with skeleton S :



$$P(I \mid \sigma, S, \theta_S) = \prod_{x \in \sigma} \prod_{A \in A(x)} P(I_{x.A} \mid I_{\mathrm{Pa}(x.A)})$$

- Where $O^{\sigma}(X)$ denotes the set of objects in skeleton S whose class is X

Acyclicity

- Problem: Distribution is not coherent when dependency structure *S* has cycles
- Naïve approach: Require acyclic class dependency graph
    - This would prohibit a dependency of the genotype of a person (child) on the genotype of a person (parent), even though it is clearly acyclic
- Better: require certain *guaranteed acyclic* slots
    - The parent slot above is guaranteed acyclic
- Graph coloring algorithm for checking legality of dependency structures

Inference in PRMs

- Given a skeleton (set of objects and relations), a PRM defines a distribution over possible instances (assignments of values to attributes)
- Same query types as in BNs
- To answer queries, we compile the PRM to its associated BN, and use BN inference.

Parameter Estimation

- We do maximum likelihood estimation for the BN induced by the structure given the skeleton

$$\hat{\theta} = \arg\max_{\theta}\left[\mathrm{L}(\theta_S; I, \sigma, S)\right]$$

$$\begin{aligned}\mathrm{L}(\theta_S; I, \sigma, S) &= \mathrm{P}(I \mid \sigma, S, \theta_S) \\ &= \prod_{X_i} \prod_{A \in A(X_i)} \prod_{x \in O^{\sigma}(X_i)} P(I_{x.A} \mid I_{\mathrm{Pa}(x.A)})\end{aligned}$$

- Parameters are tied for nodes of same class
- As in BNs, the likelihood function can be decomposed and learned separately
- As in BNs, can also take a Bayesian approach

Structure Learning

- Similar to BN, we let the search algorithm decide on the parent set

- We must ensure that the dependency structures we learn is acyclic
  - No problem, just test each dependency structure before we consider it

Structure Learning (2)

- We need a scoring function to evaluate the "goodness" of each candidate hypothesis:
  - We use Bayesian model selection, where the score of a structure *S* is defined as the posterior probability of the structure given the data *I*:

$$\mathrm{P}(S \mid I, \sigma) \propto \mathrm{P}(I \mid S, \sigma)\,\mathrm{P}(S \mid \sigma)$$

  - The second component of the score P(*S*|σ) = P(*S*) is a prior over structures
    - To penalize adding parameters we set log P(*S*) to be proportional to the total number of parameters in *S*

Structure Learning (3)

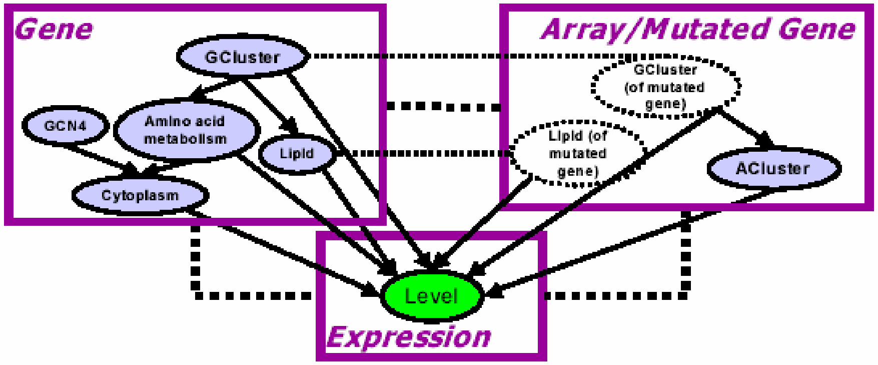
- The first component of the score is the marginal likelihood:

$$\mathrm{P}(I \mid S,\sigma) = \int \mathrm{P}(I \mid S,\sigma,\theta_S) P(\theta_S, S) d\theta_S$$

- If we use a parameter independent Dirichlet prior (over parameters), this integral decomposes into a product of integrals each of which has a simple closed form solution
  - E.g., uniform Dirichlet prior over parameters

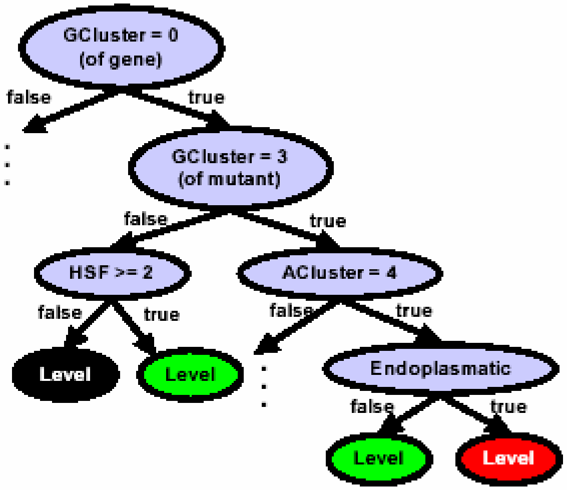
Rich probabilistic networks for gene expression

- Apply PRMs to gene expression and other data sources.
- Data sources includes:
  - Functional assignment for gene (from MIPS)
  - Binding site information for known TFs
- Gene classes are latent variables.
- Array classes are known (different class to each array).



Probability model

- Decision tree for each of the expression levels.
- Decision can be based on expression levels of other genes or on discrete values from the other data sources.
- Can use the node in the tree to determine parents for a given node.



Issues:

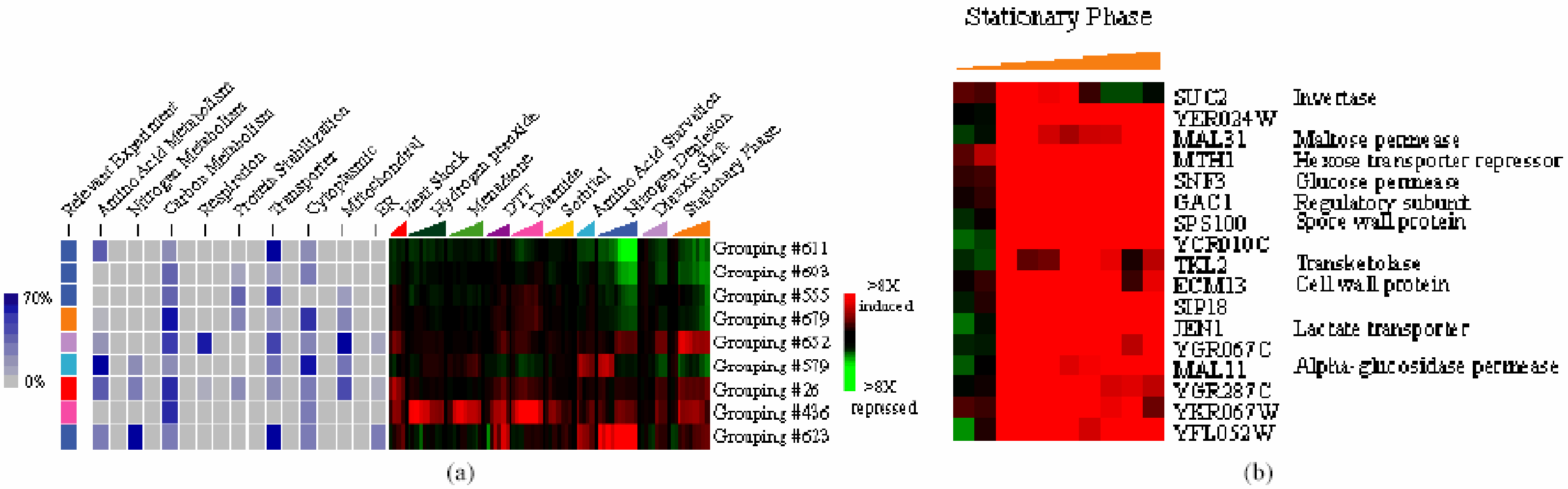
- Acyclic graph
- Learning the tree for each gene

Determining significance of results

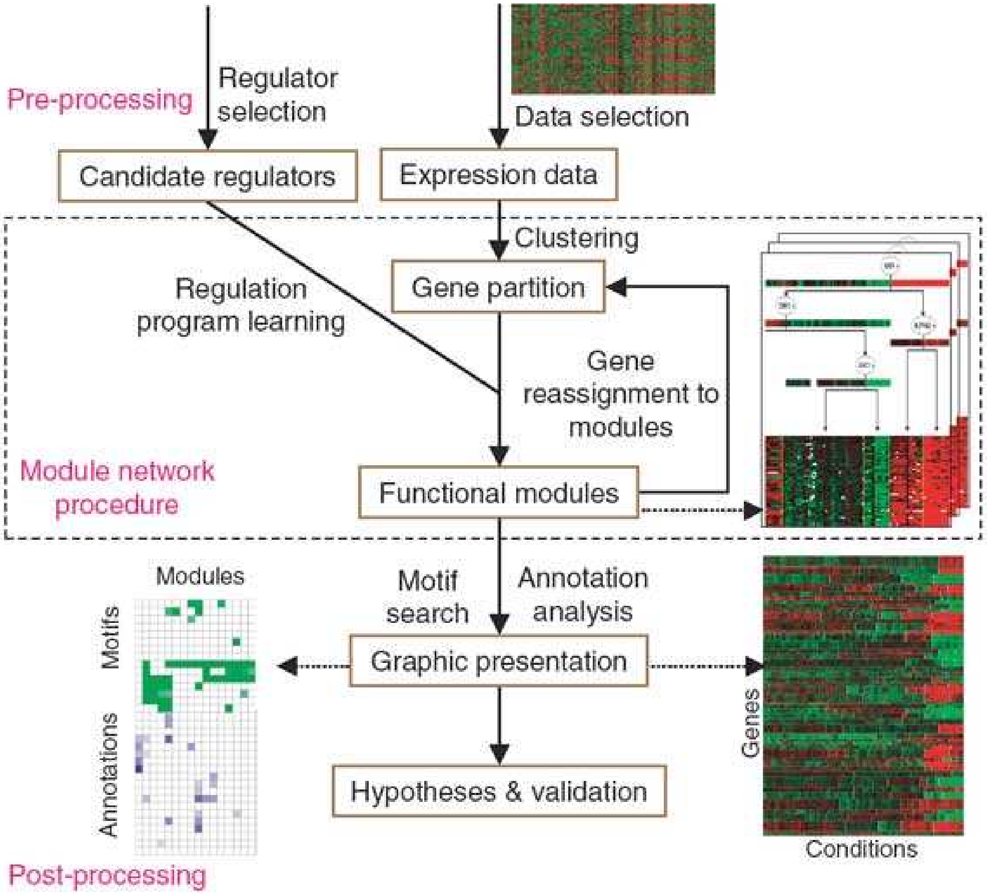
- Use permutation data to determine if the structure observed was present in the data.
- Apply the same algorithm to a randomized version of the data.
- Use likelihood of generated model to test the relevance of the learned structure.

Testing the clusters

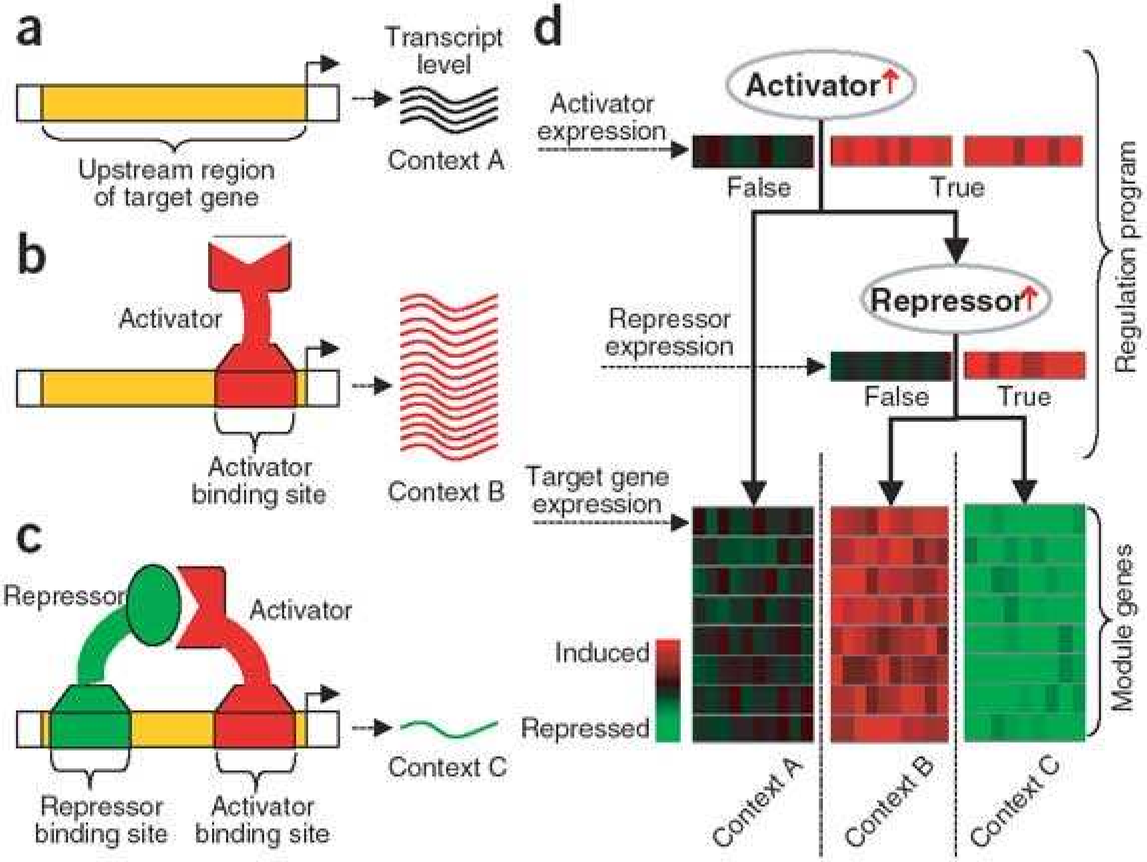
- Test the variance of the expression in each cluster.
- Remove functional annotation after initial step to allow for new annotations for unknown genes.



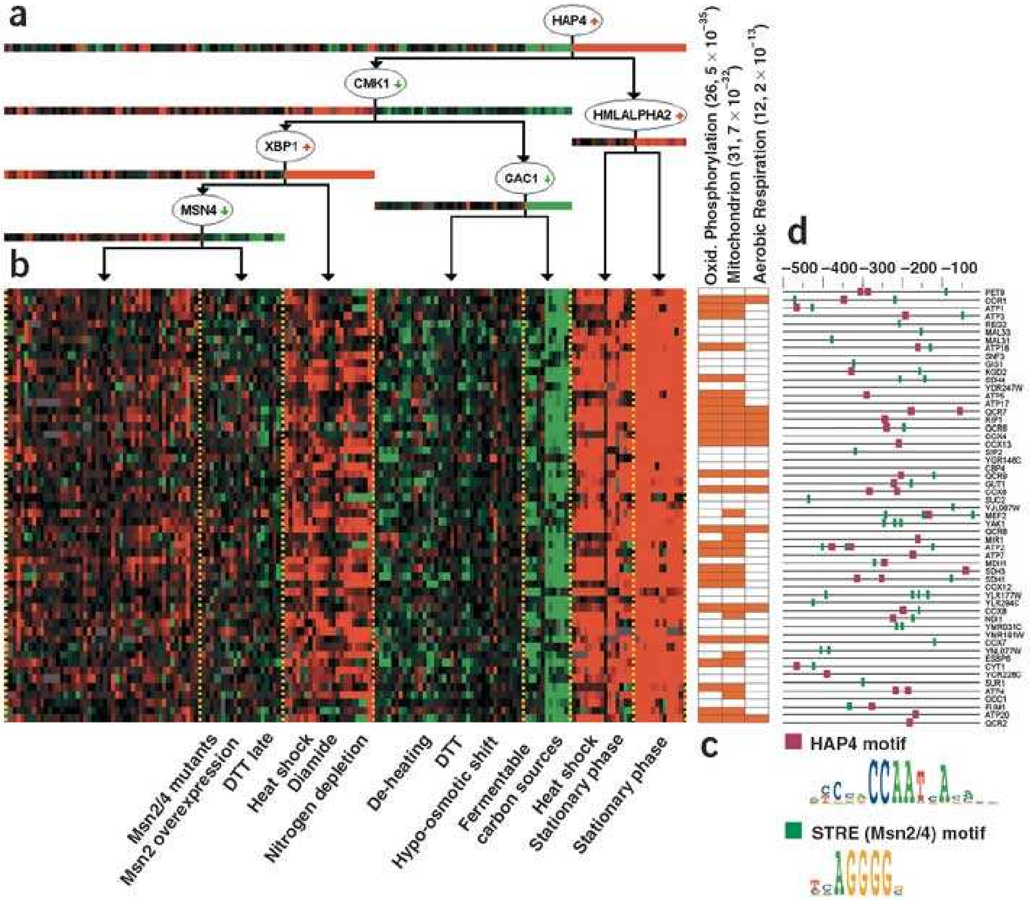
From modules to networks



Determine combinatorial control

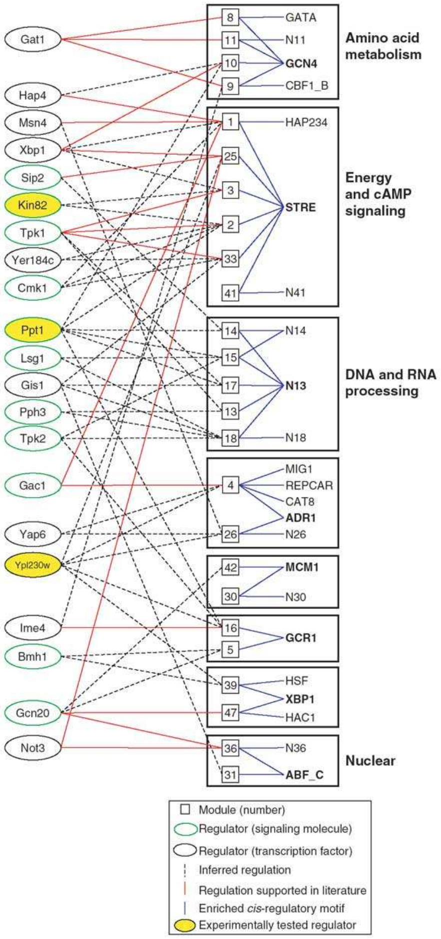


Resulting module



Segal et al Nature Genetics 2003

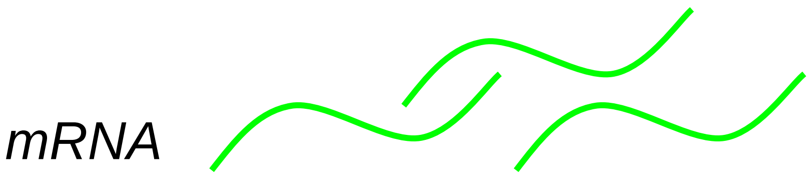
More combinatorial regulation



GRAM

Expression and Binding Data

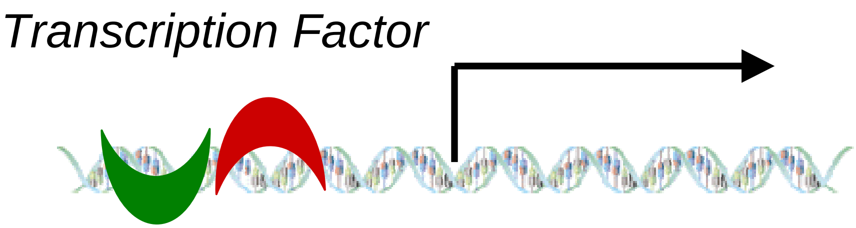
Gene expression data



expression = number of

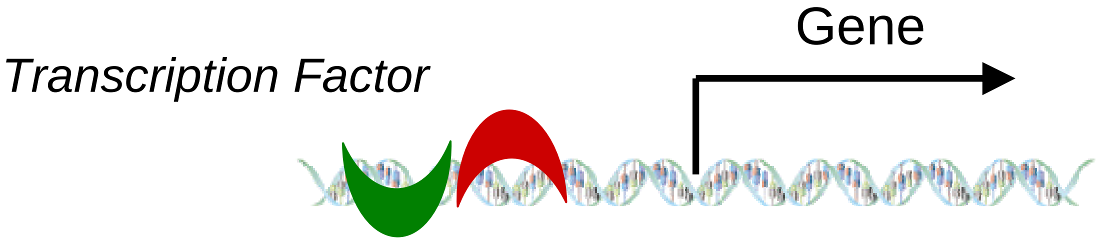
mRNA copies for a gene

Protein-DNA binding data



binding = connectivity

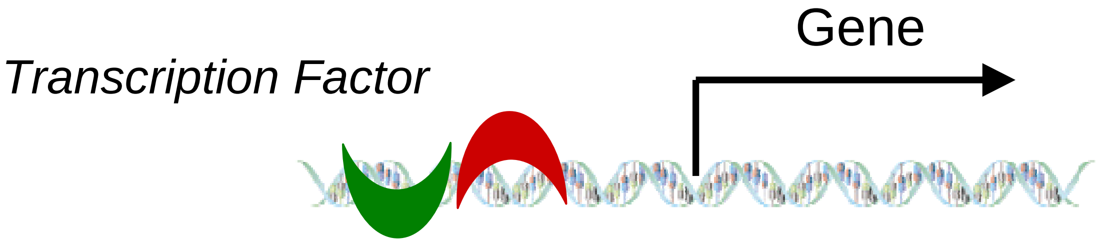
Protein-DNA Binding Data

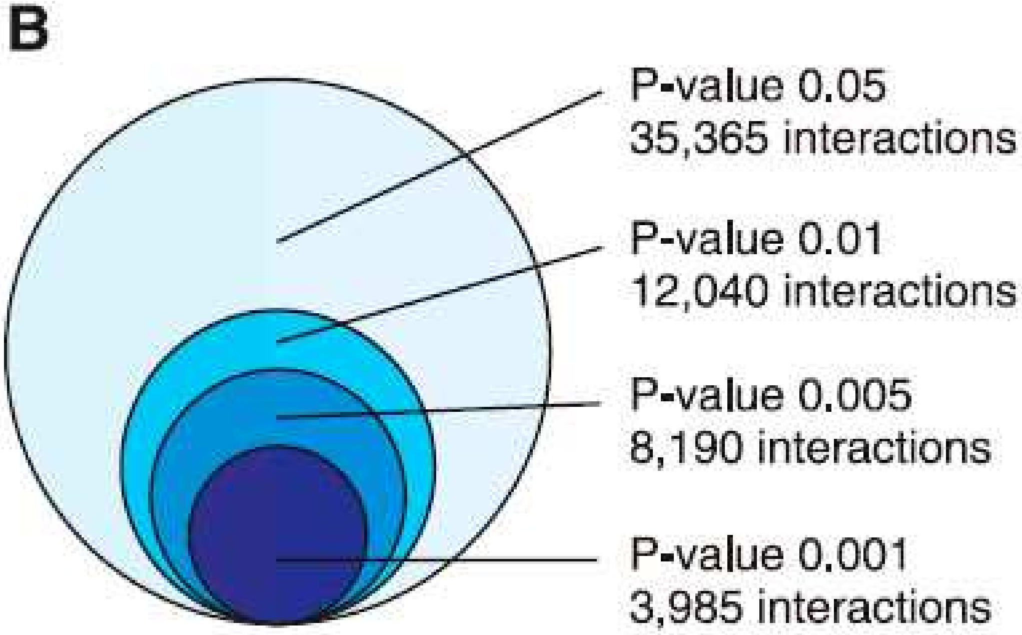


	TF_1	TF_2	TF_3	TF_4
g_1	.001	1	.08	.7
g_2	.06	.05	1	.5
g_3	.34	.001	1	1
g_4	.78	.5	.0003	0.07
g_5	0	.0006	.3	.001
g_6	1	.4	.5	.1

Ren *et al*, Science 2000

Protein-DNA Binding Data



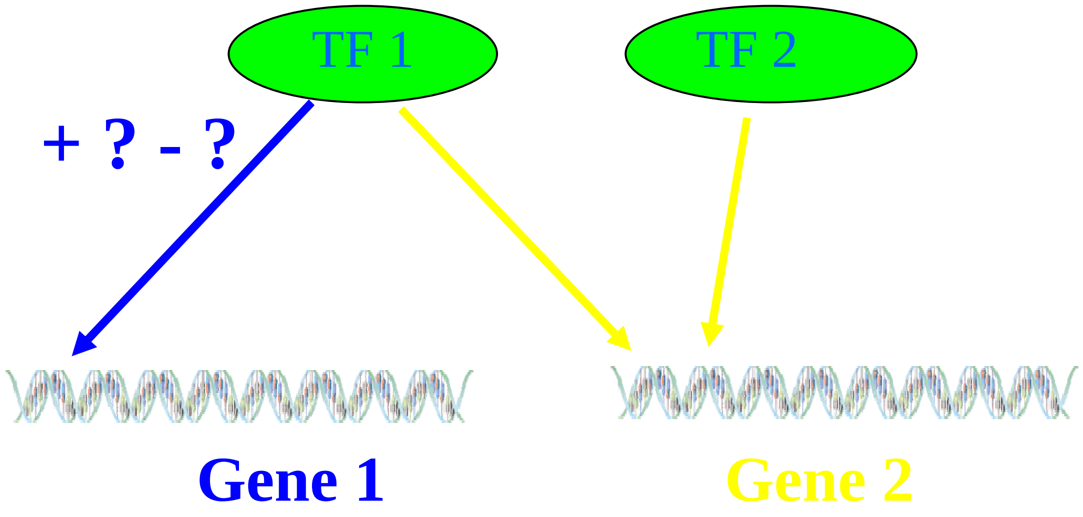


Previous work used an error model for binding data and a p-value cutoff to determine binary relationships.

Lee *et al*, *Science*, 2002

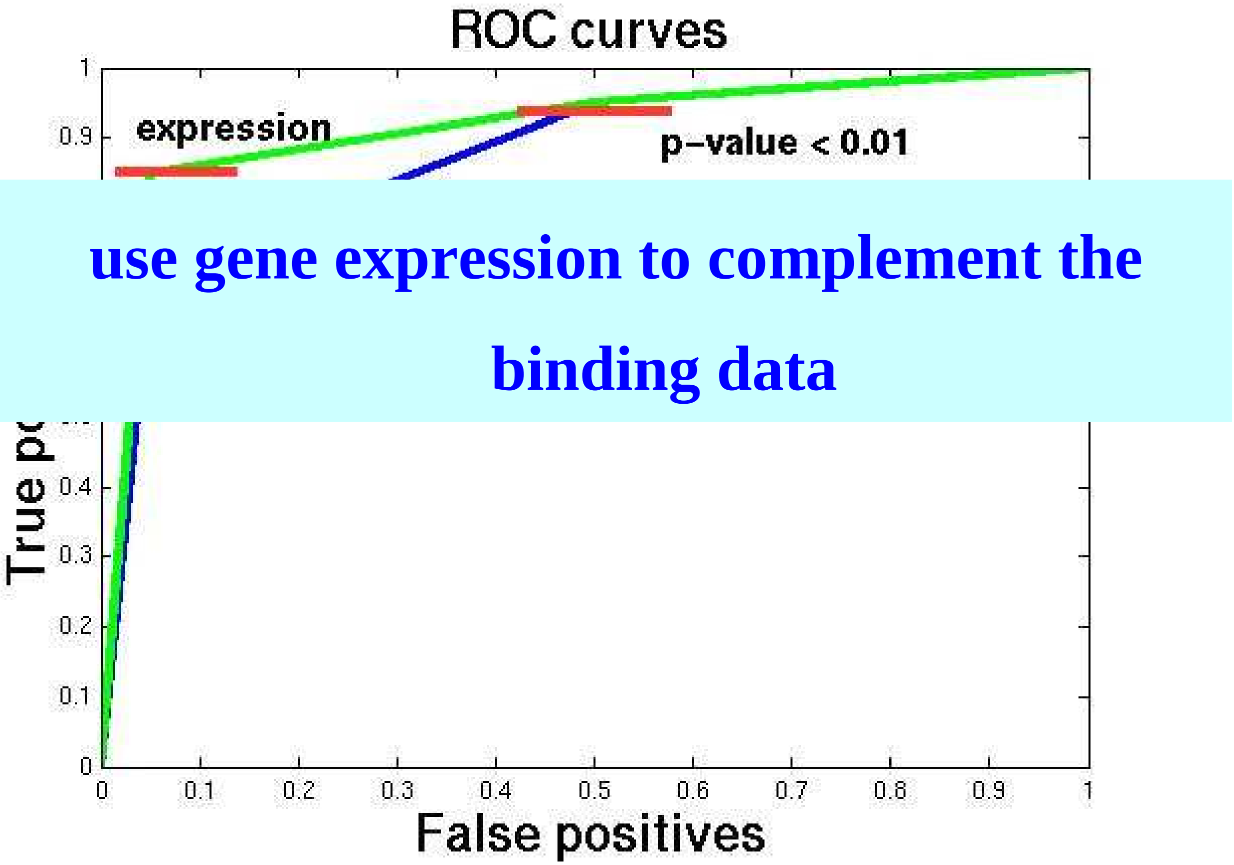
Limitations of the Binding Data

- Relationships between TFs and genes (activators, repressors).
- Type of combinatorial regulation.



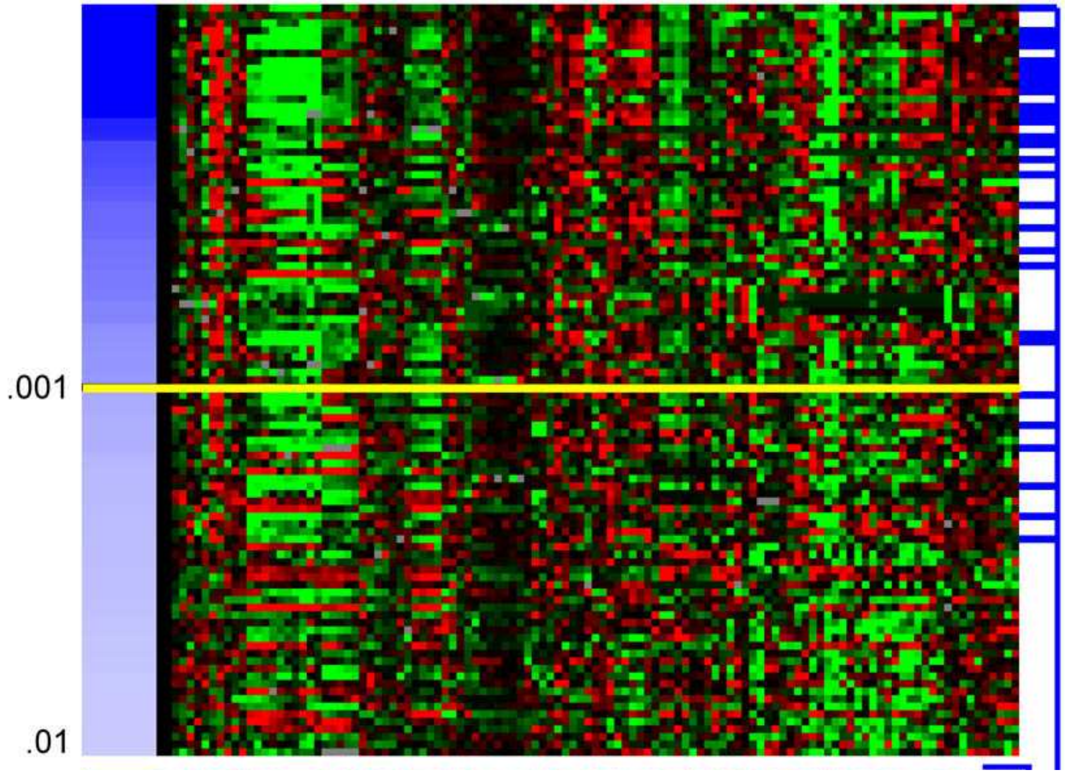
Limitations of the Binding Data

- Low false positive rate (5%) but also a low true positive rate (70%).



use gene expression to complement the binding data

Limitations of Binding Data Alone

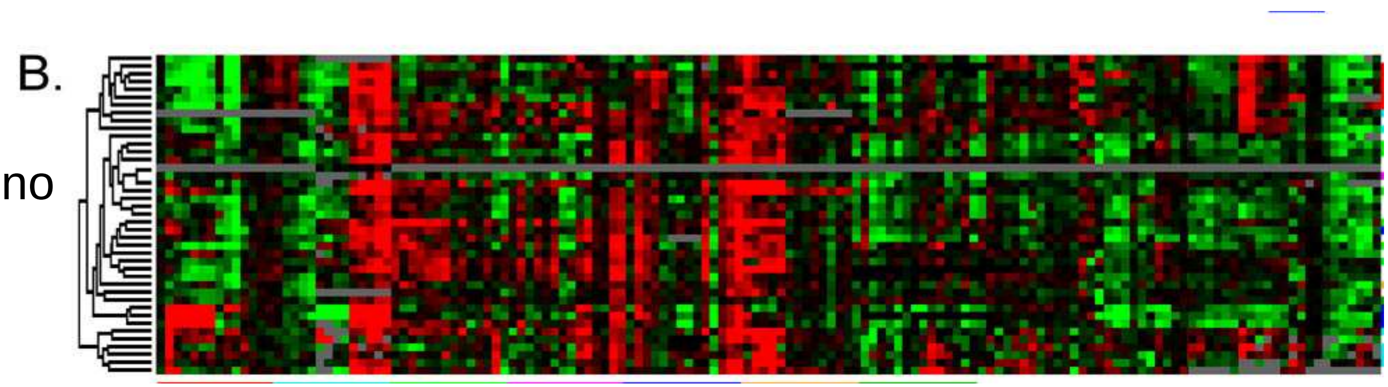


Binding p-values form a continuum – where do you draw the cut-off line?

99 genes bound by Hap4
with a p-value < .01

Limitations of Expression Data Alone

Hierarchical clustering of amino acid synthesis genes



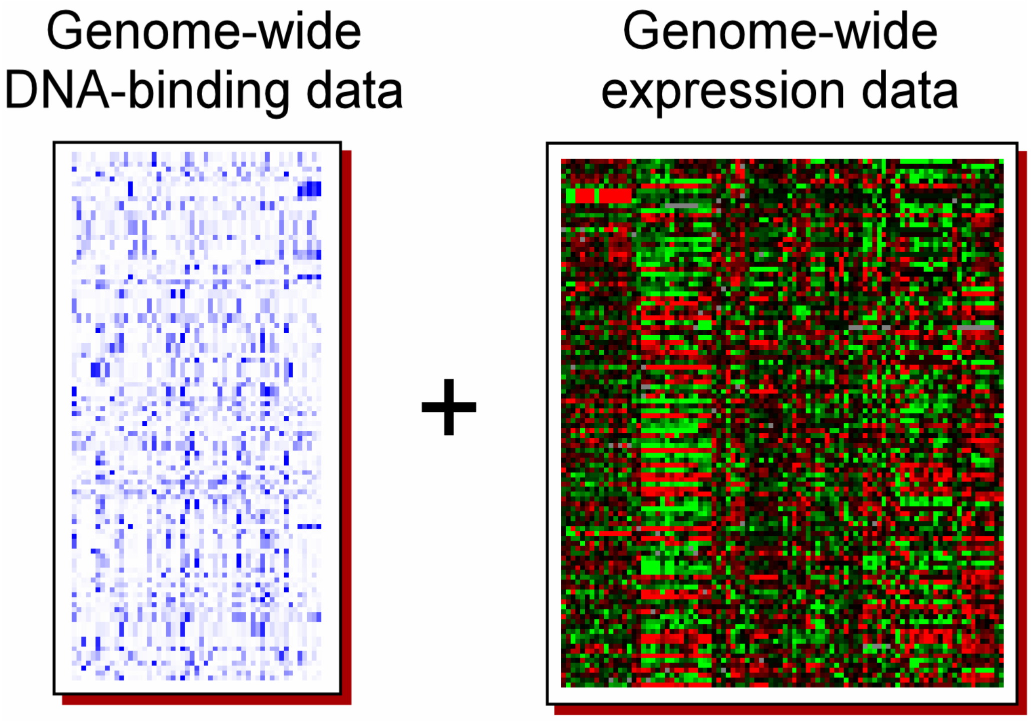
Expression data alone can’t effectively distinguish among genes that have similar expression patterns but are under the control of different regulatory networks.

Modules

- Gene Module
  - Set of genes that are co-regulated and co-expressed.
- Functional Module
  - Collection of gene modules with related function.

Modules provide an abstraction which reduces genetic network complexity without significant loss of explanatory power, and allows us to determine the significance of the model.

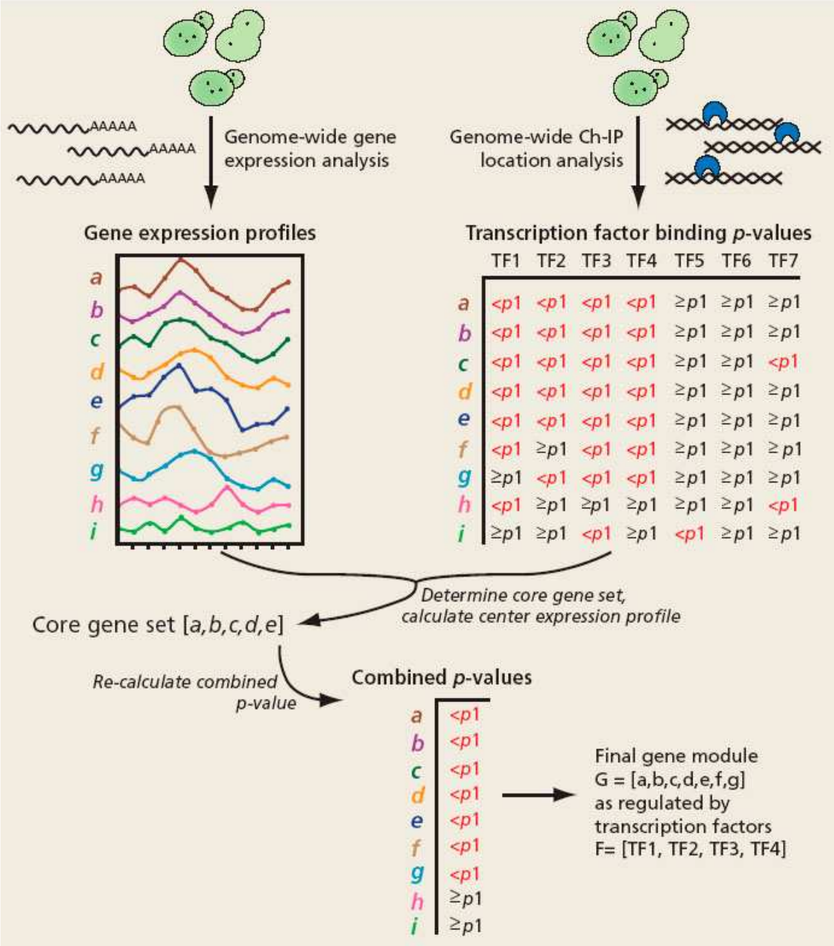
Genetic RegulAtory Modules (GRAM)



Input data to the algorithm

GRAM Algorithm Overview

- For each regulator combination, look at all genes bound (using a strict binding p-value).
- Find a core gene expression profile.
- Remove genes far away from core.
- Add genes close to the core (with relaxed p-value threshold).


AAAAA
AAAAA
AAAAA
Genome-wide gene expression analysis
Genome-wide Ch-IP location analysis
Gene expression profiles
a
b
c
d
e
f
g
h
i
Determine core gene set, calculate center expression profile
Core gene set [a,b,c,d,e]
Re-calculate combined p-value

Transcription factor binding *p*-values

	TF1	TF2	TF3	TF4	TF5	TF6	TF7
a	<p1	<p1	<p1	<p1	≥p1	≥p1	≥p1
b	<p1	<p1	<p1	<p1	≥p1	≥p1	≥p1
c	<p1	<p1	<p1	<p1	≥p1	≥p1	<p1
d	<p1	<p1	<p1	<p1	≥p1	≥p1	≥p1
e	<p1	<p1	<p1	<p1	≥p1	≥p1	≥p1
f	<p1	≥p1	<p1	<p1	≥p1	≥p1	≥p1
g	≥p1	<p1	<p1	<p1	≥p1	≥p1	≥p1
h	<p1	≥p1	≥p1	≥p1	≥p1	≥p1	<p1
i	≥p1	≥p1	<p1	≥p1	<p1	≥p1	≥p1

Combined *p*-values

a	<p1
b	<p1
c	<p1
d	<p1
e	<p1
f	<p1
g	<p1
h	≥p1
i	≥p1

Final gene module
G = [a,b,c,d,e,f,g]
as regulated by
transcription factors
F= [TF1, TF2, TF3, TF4]

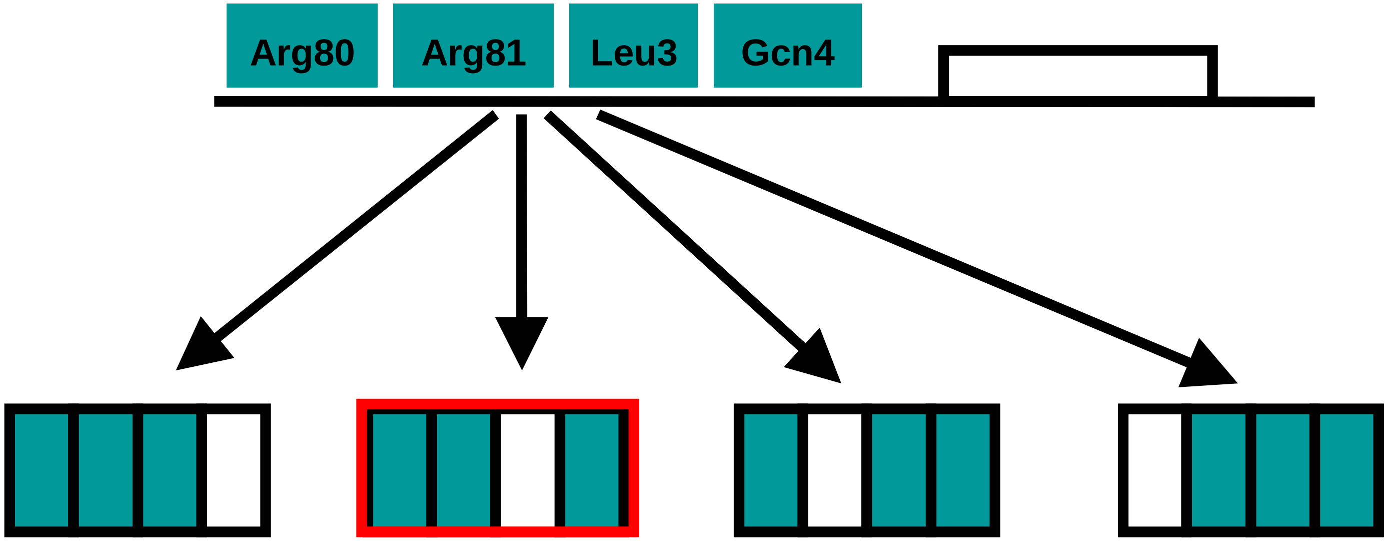
GRAM step 0:

- For each gene *i*:

  Generate all possible subsets of factors that bind to gene *i* with p-value < 0.001. Associate the gene with all the TF subsets via a hash-table.

- Result is the set of all possible binding patterns (as indicated by strict binding p-values), with the corresponding genes mapped to the patterns.

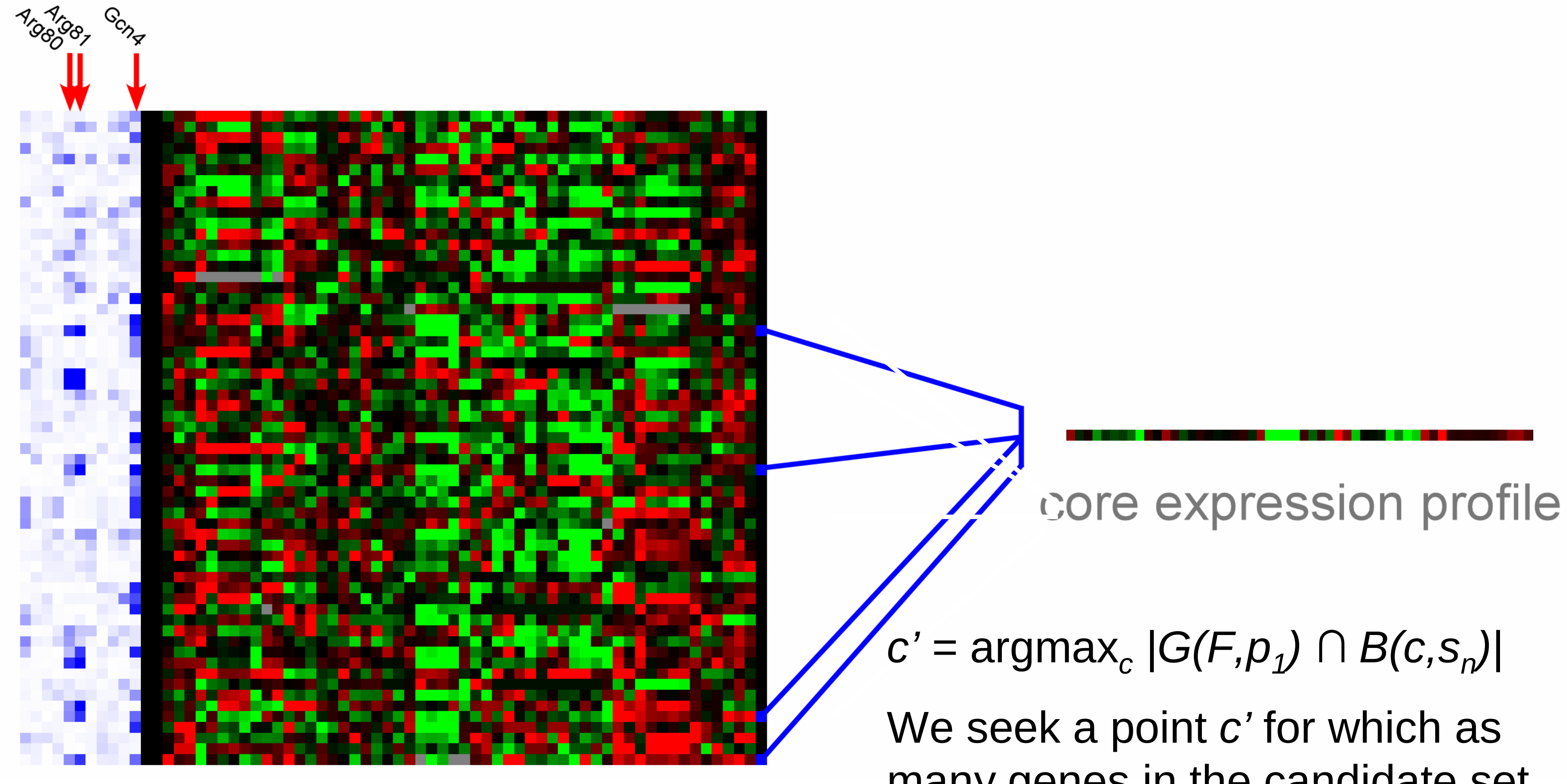
GRAM Algorithm Step 1: exhaustively search all subsets of TFs (starting w/ the largest sets)



	Arg80	Arg81	Leu3	Gcn4
g_1	1	1	0	1
g_2	1	0	1	1
g_3	1	1	1	1
g_4	0	0	0	0
g_5	1	1	1	1
g_6	1	1	0	1

For every set of transcription factors *F*, the genes in $G(F,p_1)$ serve as candidates for a module regulated by the factors in *F*.

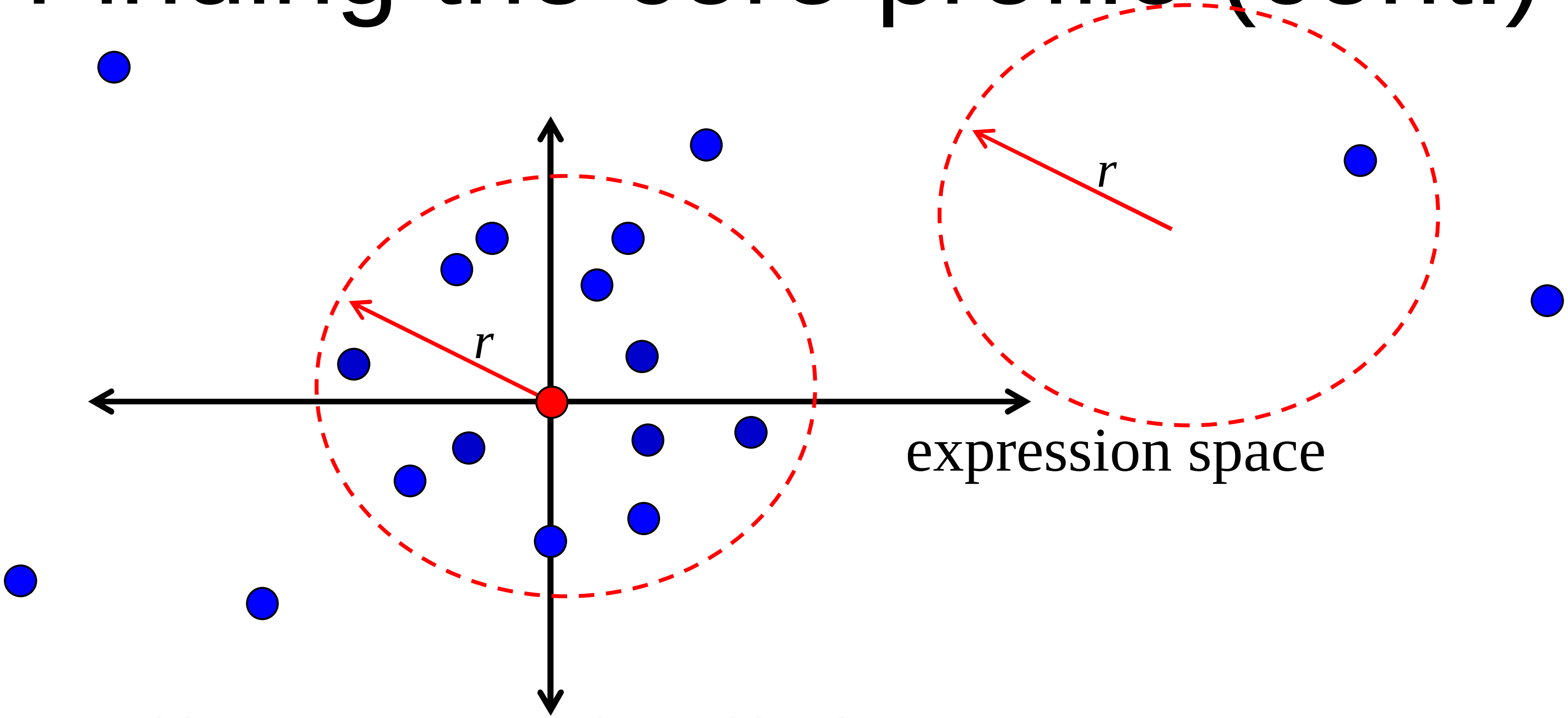
GRAM Algorithm Step 2: find a core expression profile for the module



$c' = \mathrm{argmax}_c \, |G(F,p_1) \cap B(c,s_n)|$

We seek a point *c'* for which as many genes in the candidate set are within distance s_n of the point *c'*.

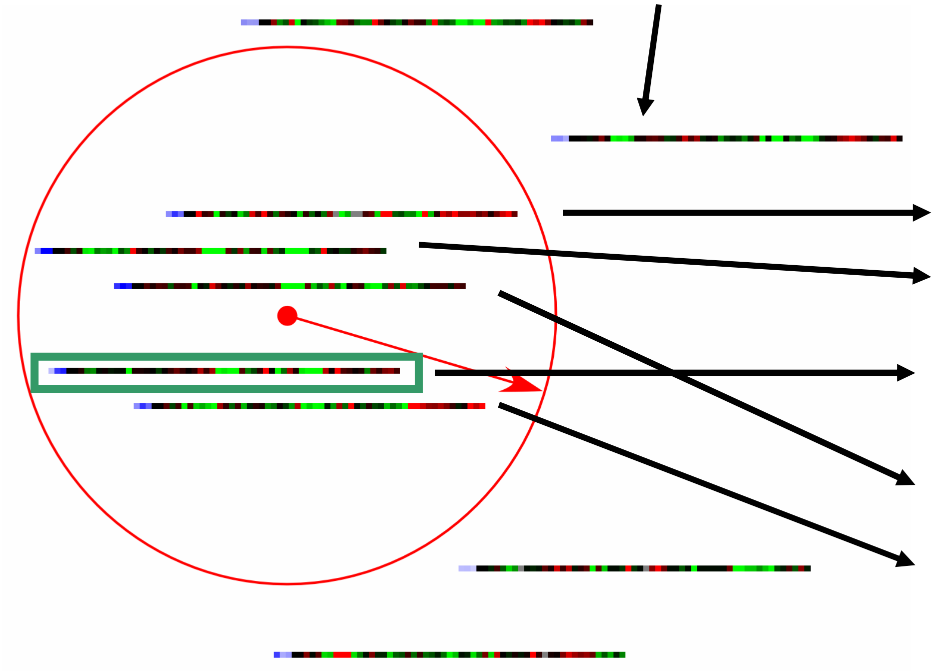
Finding the core profile (cont.)



- Consider a set of genes bound by the same TFs.
- The core profile is a point in expression space that describes a ball containing the maximal number of genes within a distance *r*.
- This estimate is **robust**, in the sense that it is insensitive to outliers (think of a median versus a mean).
- To compute it exactly requires an $O(2^n)$ algorithm (n=# of genes).
- Using results from computational geometry, we get an $O(n^3)$ approximation algorithm (with provable error bounds).

Step 3: add/remove genes

2. Remove genes with significantly far expression profiles



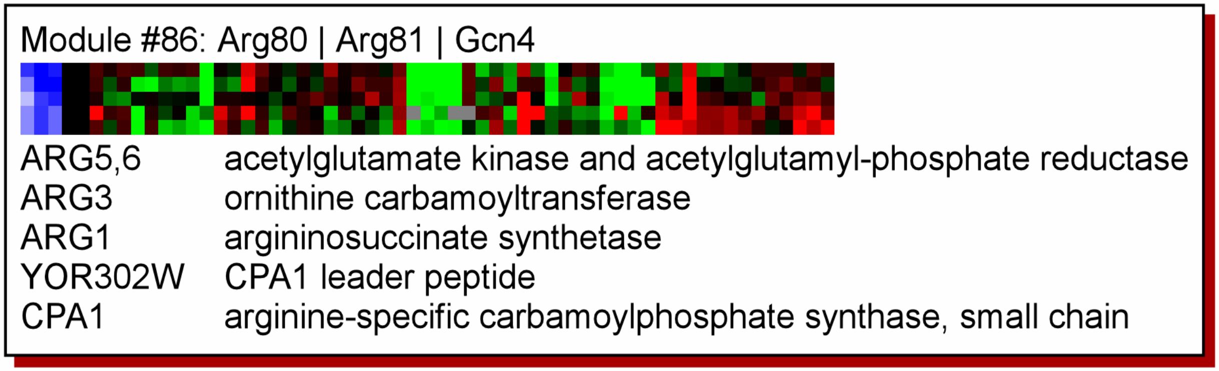
Expanded set = $G(F,p_2) \cap B(c',s_n)$, where $p_2 > p_1$.

1. Include genes that are close and are bound by same TFs (binding p-value < 0.001)

	Arg80	Arg81	Leu3	Gcn4	
g_1	.0004	.00003	.33	.0004	√
g_2	.00002	0.0006	.02	.0001	√
g_3	.0007	.002	.15	.0002	√
g_4	.007	.2	0.04	.7	x
g_5	.00001	.00001	.0001	.0002	√
g_6	.00001	.00007	.5	.0001	√

3. Relax the binding threshold/ add genes with significantly close expression profiles

GRAM: Final Module



Module #86: Arg80 | Arg81 | Gcn4

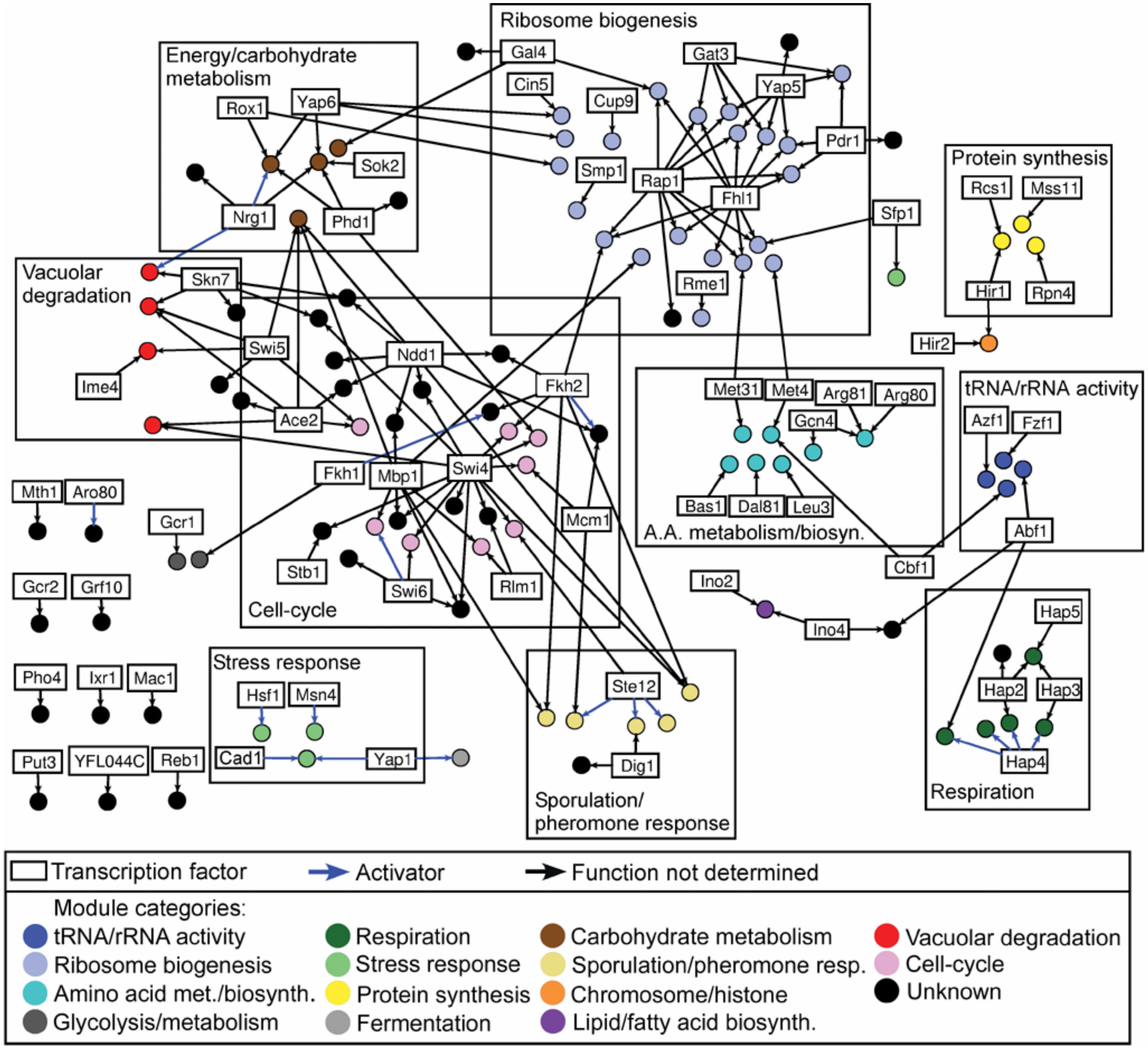
ARG5,6	acetylglutamate kinase and acetylglutamyl-phosphate reductase
ARG3	ornithine carbamoyltransferase
ARG1	argininosuccinate synthetase
YOR302W	CPA1 leader peptide
CPA1	arginine-specific carbamoylphosphate synthase, small chain

GRAM results

- 106 modules ranging in size from 52 genes to 5.
- These modules are controlled by 68 factors and contain 655 genes.
- 627 out of 1560 unique regulator-gene interactions (40%) had binding p-value > .001

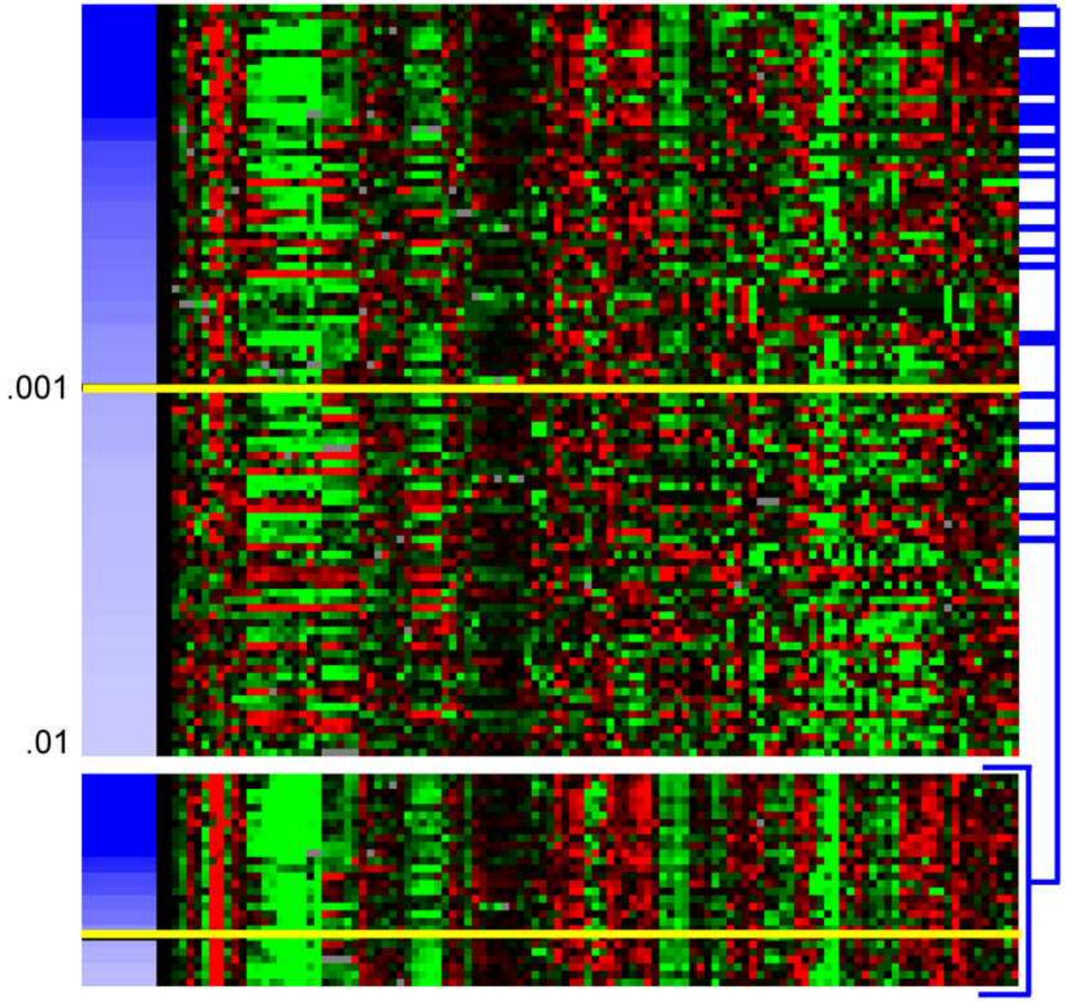
Results:

Rich
Media
Modules



Bar-Joseph et al Nature Biotechnology 2003

The Importance of Information Fusion: Using Only Binding



99 genes bound by Hap4 with a p-value < .01

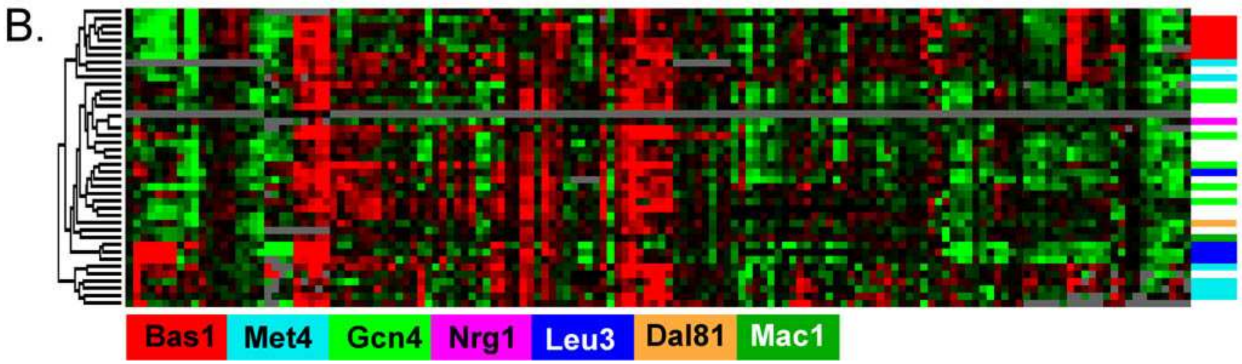
Binding p-values form a continuum – where do you draw the cut-off line?

28 genes were selected by the GRAM algorithm; all are involved in respiration. Six of these genes (PET9, ATP16, KGD2, QCR6, SDH1, and NDI1) would not have been identified as Hap4 targets using the stringent .001 p-value threshold (p-values range from .0011 to .0036).

The Importance of Information Fusion: Using Only Expression

A cluster of amino acid

synthesis genes



Eleven Significant Activators Found; Ten Previously Identified in Literature

Factor	**Module function**	**Correlation**	**Comments**
Ste12	Pheromone response	+0.64	Activator, required for pheromone response
Hap4	Respiration	+0.60	Activator of CCAAT box containing genes
Yap1	Detoxification	+0.53	Activator, possibly involved in oxidative stress response
Nrg1	Carbohydrate transport	+0.50	Previously identified as a repressor
Fkh1	Cell cycle	+0.49	Activator of cell cycle genes
Cad1	Detoxification	+0.47	Activator, involved in multi-drug resistance
Aro80	Energy and metabolism	+0.40	Activator, regulation of amino acid synthesis
Swi6	Cell cycle	+0.39	Activator of cell cycle genes
Msn4	Stress response	+0.38	Activator, involved in stress response
Fkh2	Cell cycle	+0.37	Activator of cell cycle genes
Hsf1	Stress response	+0.36	Activator of heat shock related genes

Validation Ideas

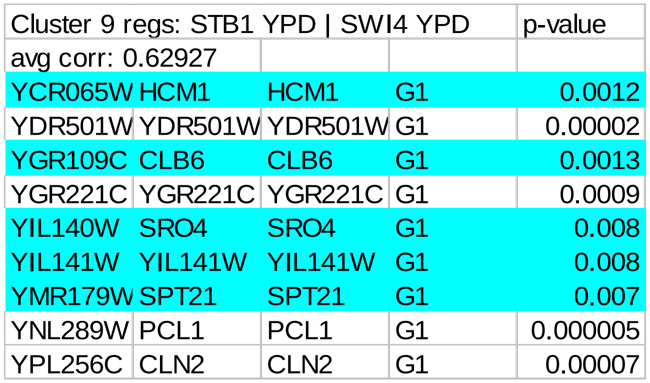
- Literature.
- Curated databases (e.g., GO/MIPS/TRANSFAC).
- Other high throughput data sources.
- “Randomized” versions of data.
- New experiments.

GRAM Network Validation

- Literature:
  - Many TF interactions predicted by modules corresponded well to literature (but what about ones that didn’t…)
- Curated databases:
  - Computed enrichment for genes in modules for MIPS categories using the hypergeometric distribution.
  - Modules belong to diverse array of categories corresponding to cellular processes such as amino acid biosynthesis, carbohydrate and fatty acid metabolism, respiration, ribosome biogenesis, stress response, protein synthesis, fermentation, and the cell cycle.
- “Randomized” data:
  - When compared to results generated using binding data alone, there was 3-fold increase in modules significantly enriched in MIPS categories.

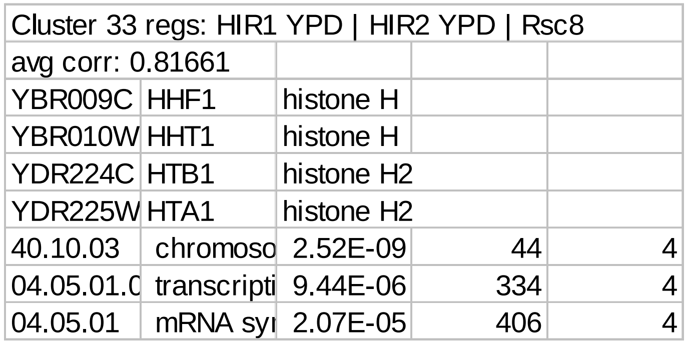
Predicting Mechanisms of Transcription Factors Regulation

Binding Predictions

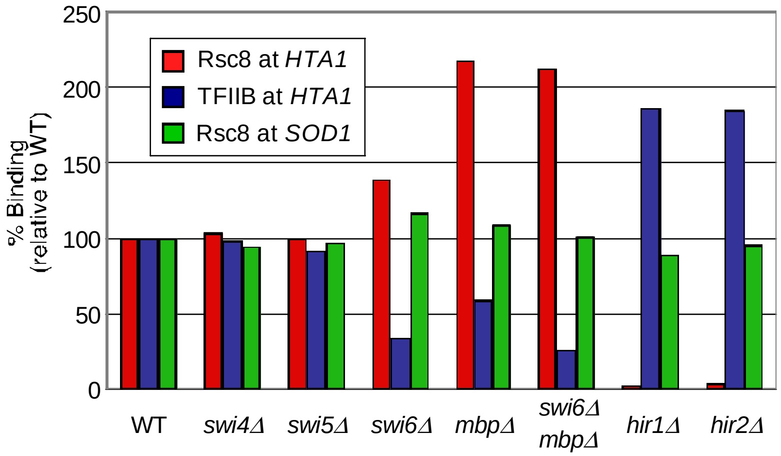


Cluster 9 regs: STB1 YPD | SWI4 YPD				p-value
avg corr: 0.62927				
YCR065W	HCM1	HCM1	G1	0.0012
YDR501W	YDR501W	YDR501W	G1	0.00002
YGR109C	CLB6	CLB6	G1	0.0013
YGR221C	YGR221C	YGR221C	G1	0.0009
YIL140W	SRO4	SRO4	G1	0.008
YIL141W	YIL141W	YIL141W	G1	0.008
YMR179W	SPT21	SPT21	G1	0.007
YNL289W	PCL1	PCL1	G1	0.000005
YPL256C	CLN2	CLN2	G1	0.00007

Combinatorial regulation



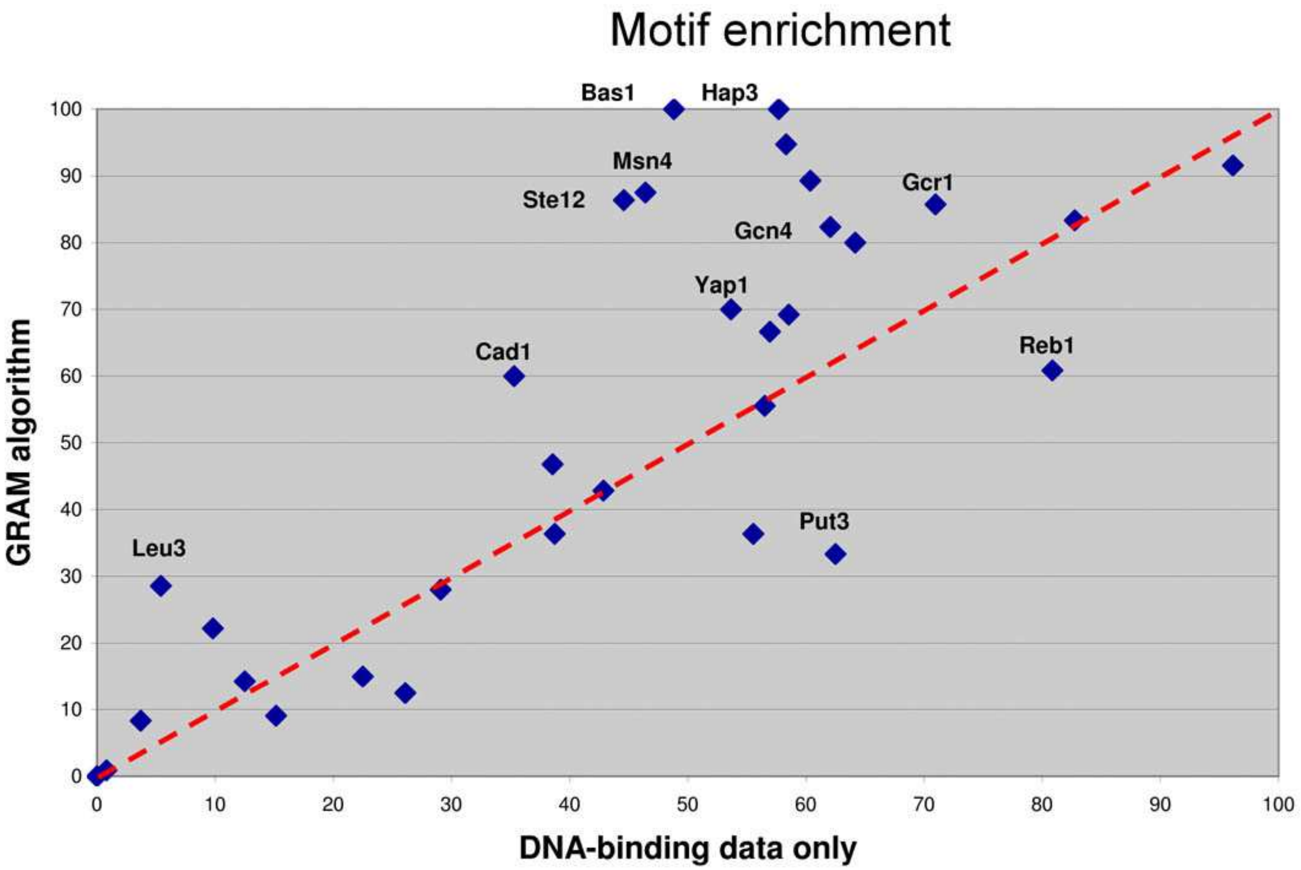
Cluster 33 regs: HIR1 YPD | HIR2 YPD | Rsc8				
avg corr: 0.81661				
YBR009C	HHF1	histone H		
YBR010W	HHT1	histone H		
YDR224C	HTB1	histone H2		
YDR225W	HTA1	histone H2		
40.10.03	chromoso	2.52E-09	44	4
04.05.01.0	transcripti	9.44E-06	334	4
04.05.01	mRNA syı	2.07E-05	406	4



Ng *et al*, *Genes Dev*. *2002*

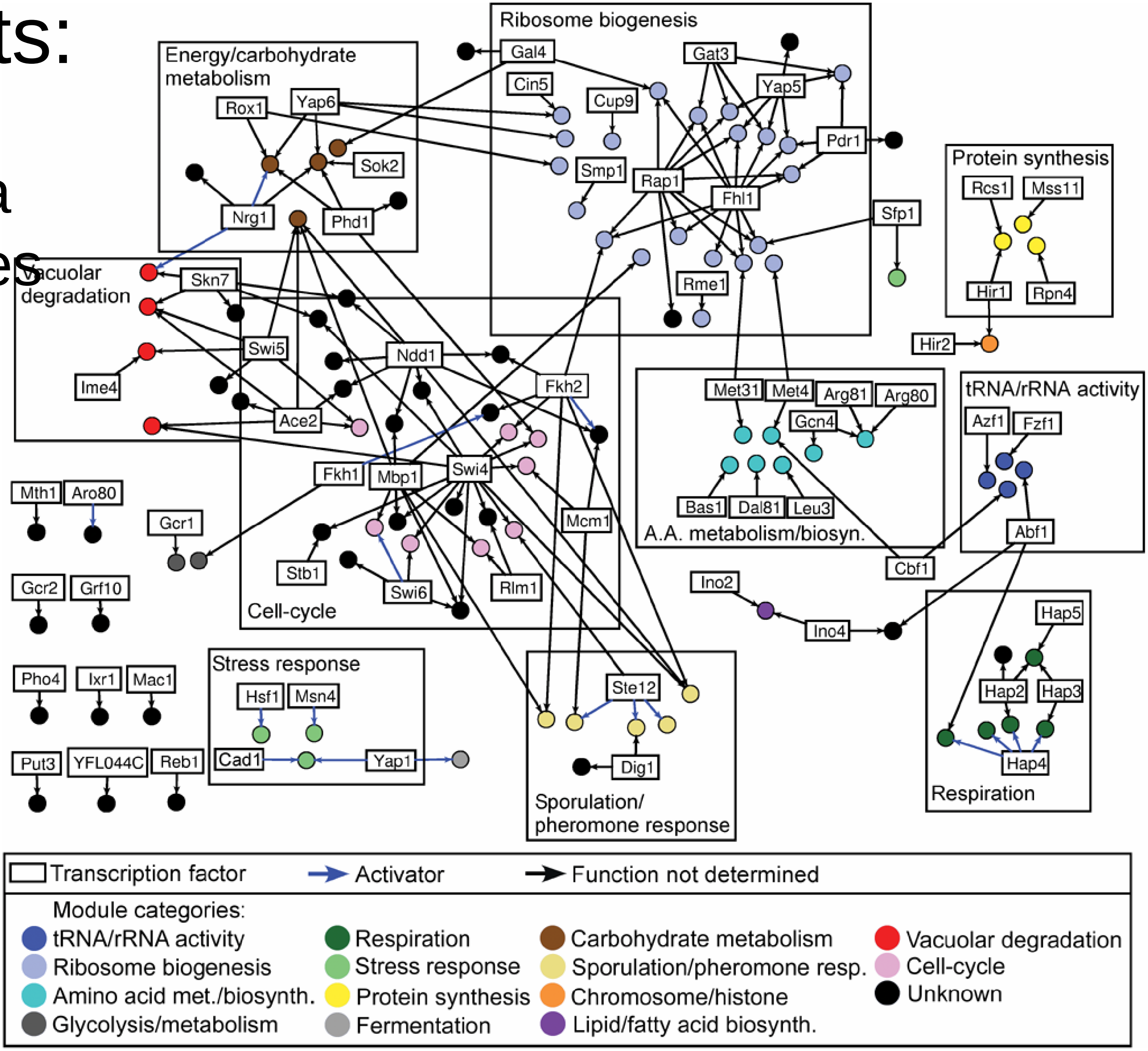
Connection to Sequence Data

What is the percentage of genes bound by factors with known motifs that contain the motif ?



Results:

Rich Media Modules



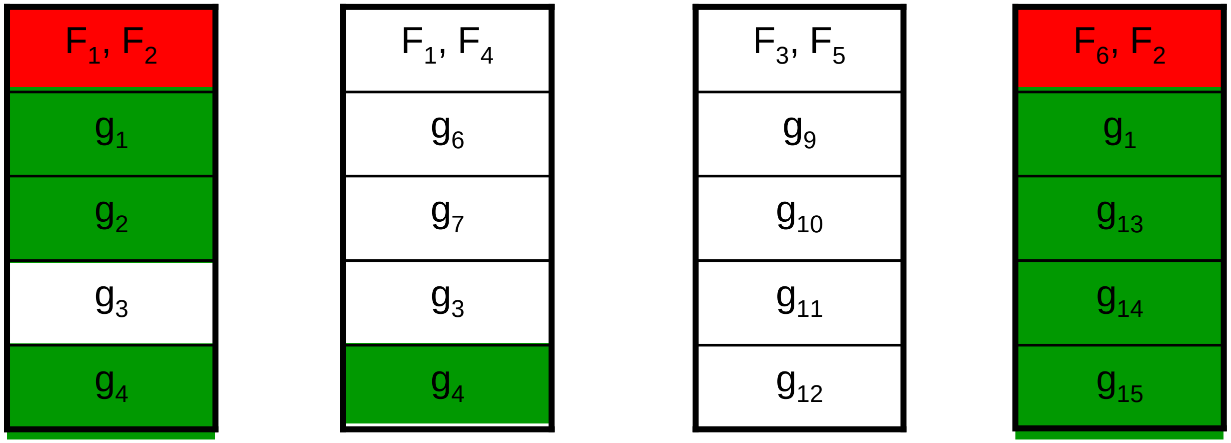
Sub-Network Discovery

- Identify genes involved in the system.
- Identify the factors controlling the system, and the modules involved.
- Determine a dynamic model for the activation of the modules by the identified factors.

We extend GRAM and combine it with our continuous representation and alignment algorithms to construct a dynamic model for a sub-network

Sub-Networks Discovery Algorithm

1.

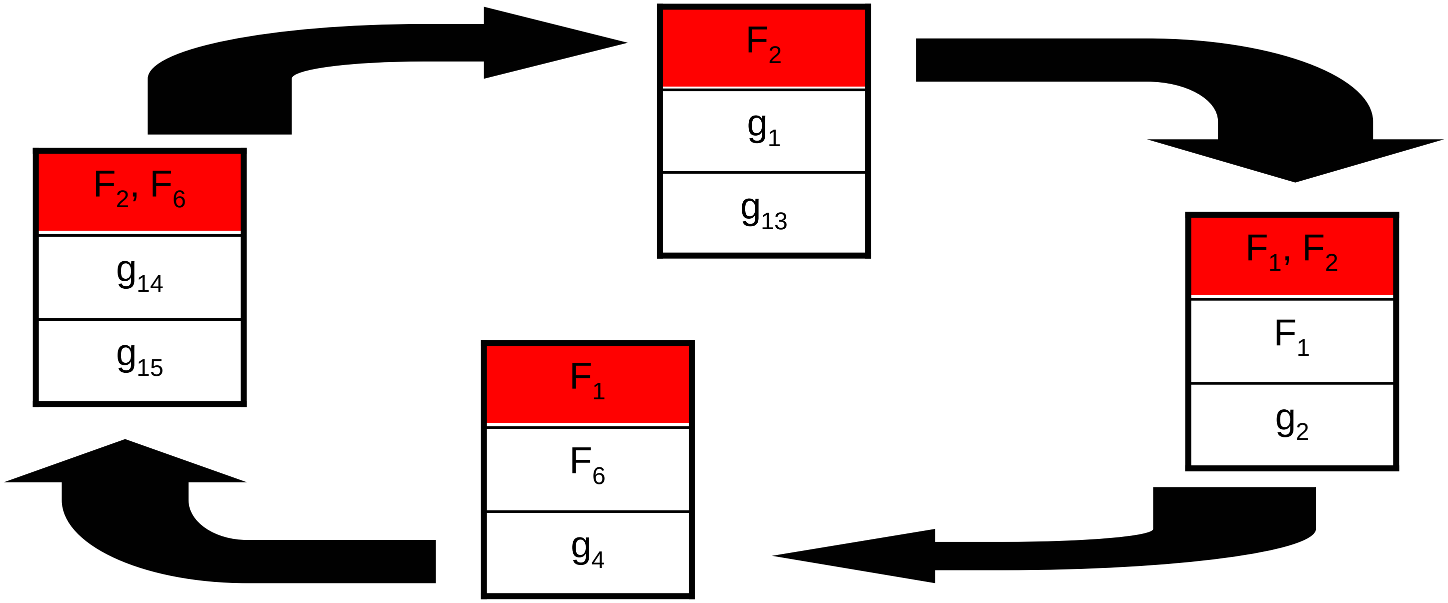


F_1, F_2	F_1, F_4	F_3, F_5	F_6, F_2
g_1	g_6	g_9	g_1
g_2	g_7	g_{10}	g_{13}
g_3	g_3	g_{11}	g_{14}
g_4	g_4	g_{12}	g_{15}

p-value: 10^{-4} p-value: 0.2 p-value: 0.7 p-value: 10^{-6}

Factors = $\{F_1, F_2, F_6\}$ Genes = $\{g_1, g_2, g_4, g_{13}, g_{14}, g_{15}, g_{22}, g_{24}\}$

2.



F_2
g_1
g_{13}

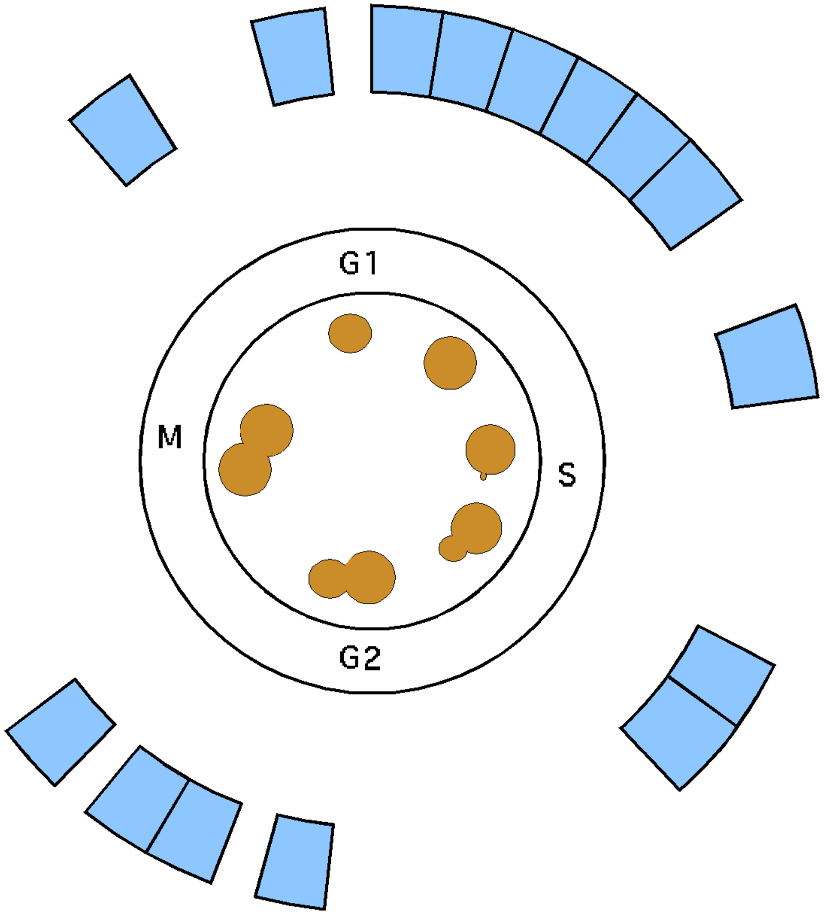
F_1, F_2
F_1
g_2

F_1
F_6
g_4

F_2, F_6
g_{14}
g_{15}

Assembly of the Cell Cycle Transcriptional Regulatory Network

Blue boxes: gene modules

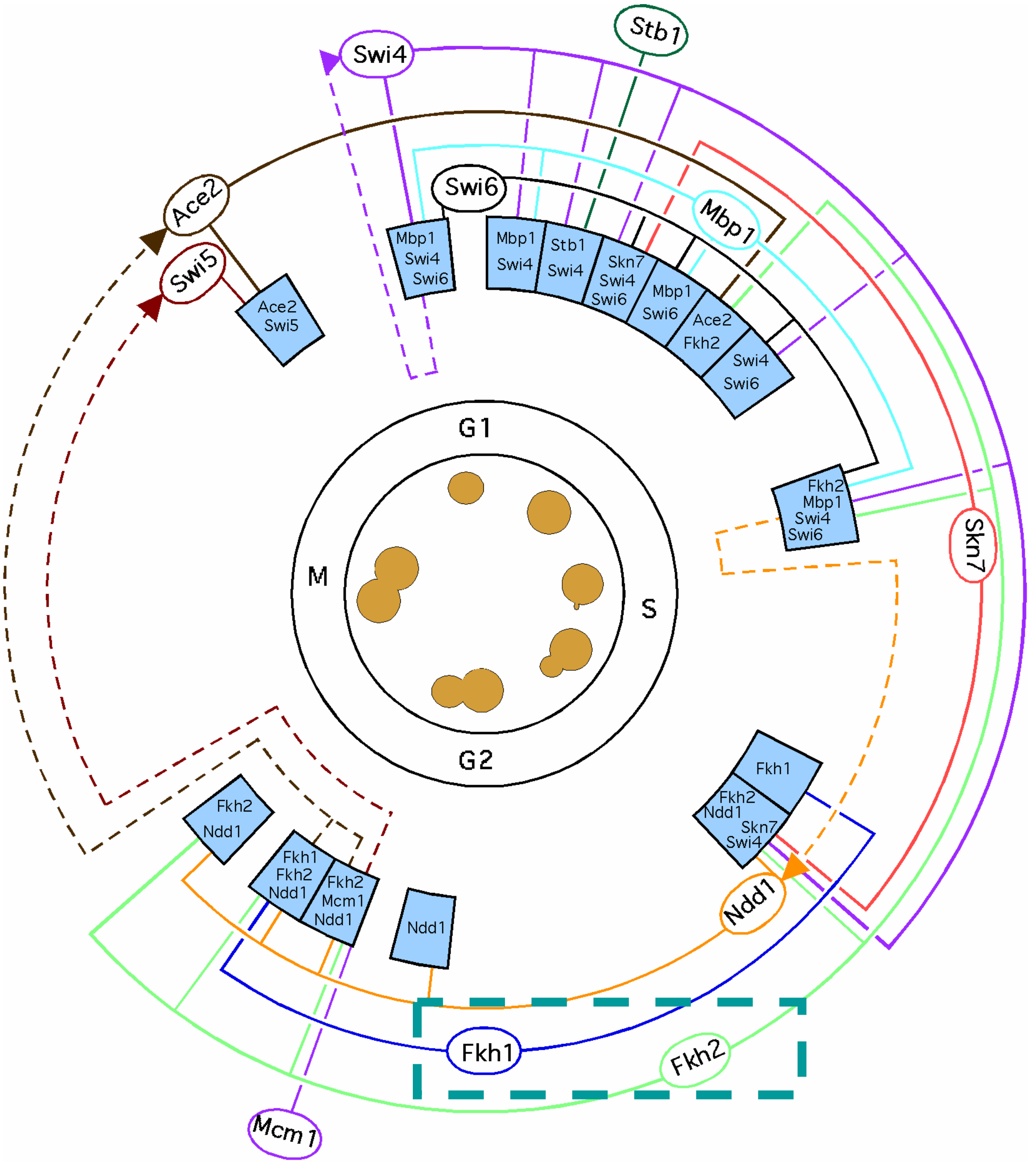


Assembly of the Cell Cycle Transcriptional Regulatory Network

Blue boxes: gene modules

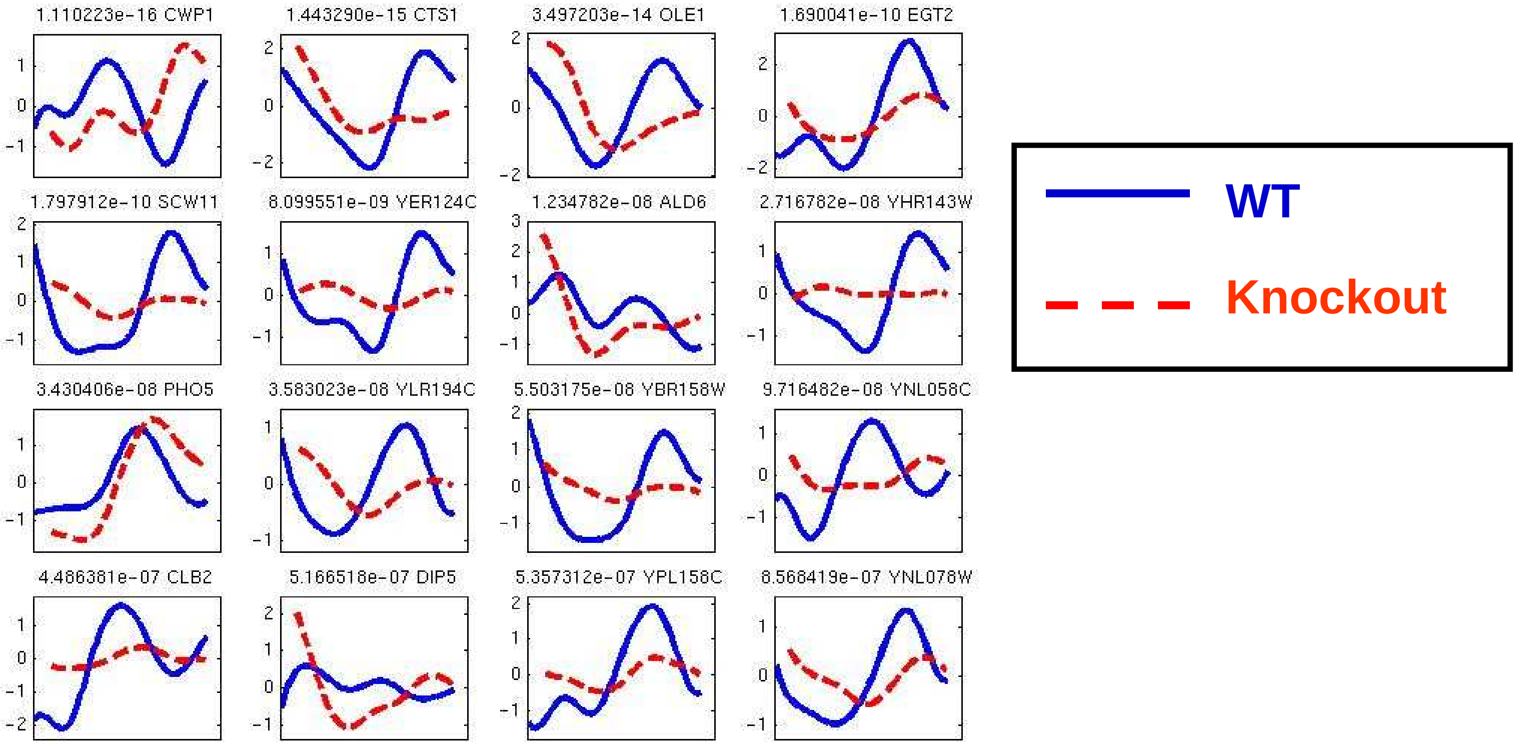
Individual regulators: ovals, connected to their modules

Dashed line: extends from module encoding a regulator to the regulator protein oval



Lee et al Science 2002

Results for the Fkh1/2 Knockout



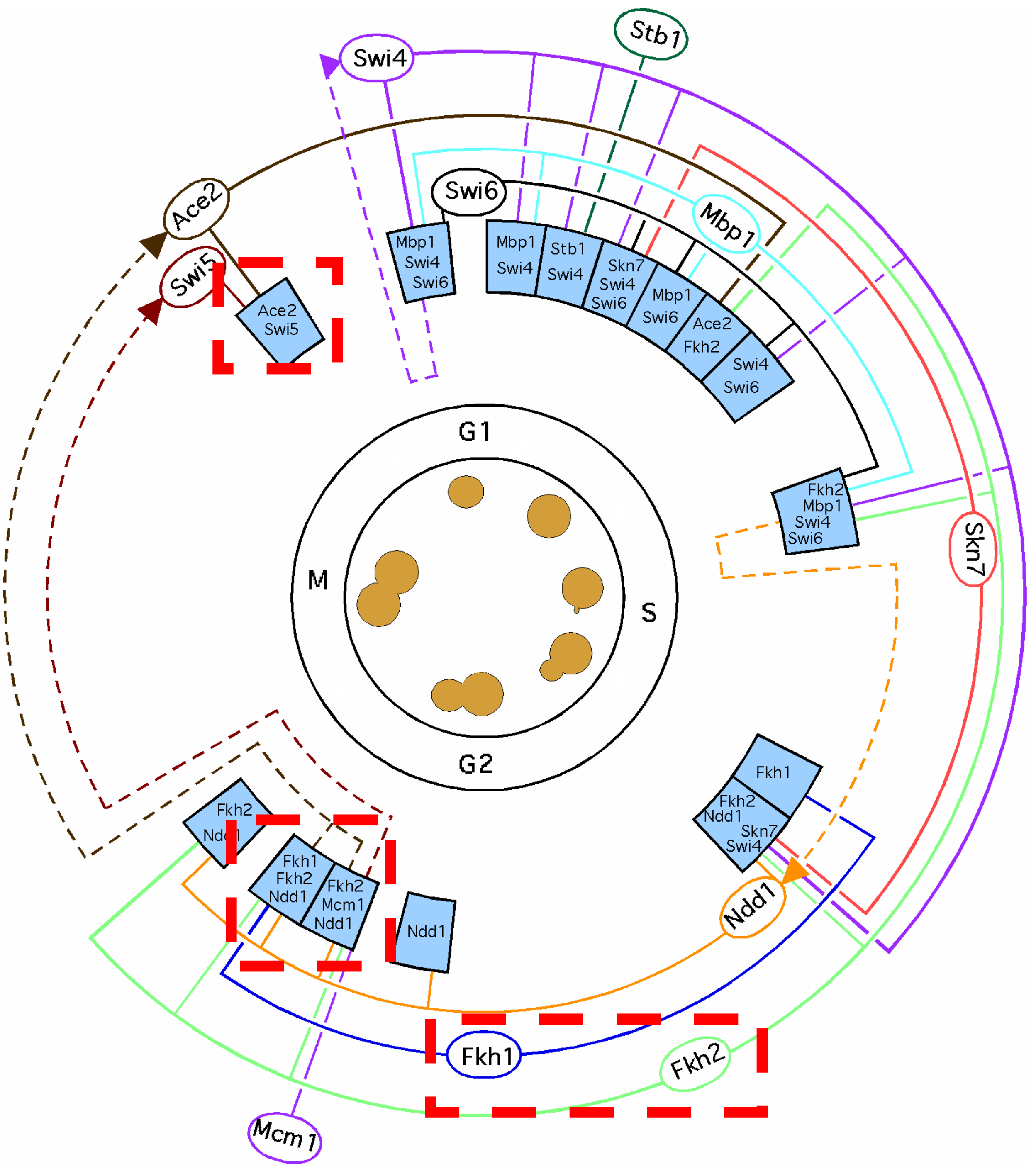
Bar-Joseph et al PNAS 2003

Assembly of the Cell Cycle Transcriptional Regulatory Network

Blue boxes: gene modules

Individual regulators: ovals, connected to their modules

Dashed line: extends from module encoding a regulator to the regulator protein oval



Projects

- Poster session: May 8th 1:00p until 2:30p in NSH 1507
- Each group should be ready to present their poster at that time
- Writeups (6-8 pages, no more than 8) are due at the poster session