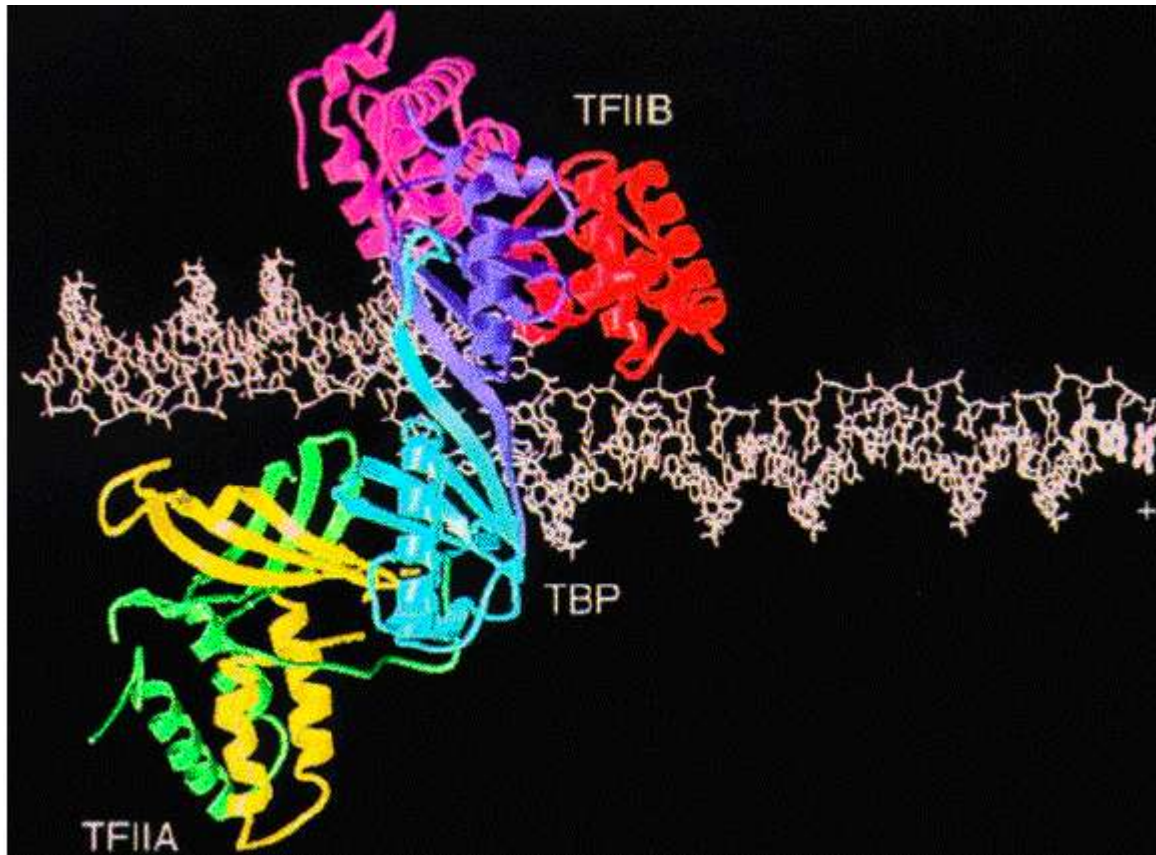
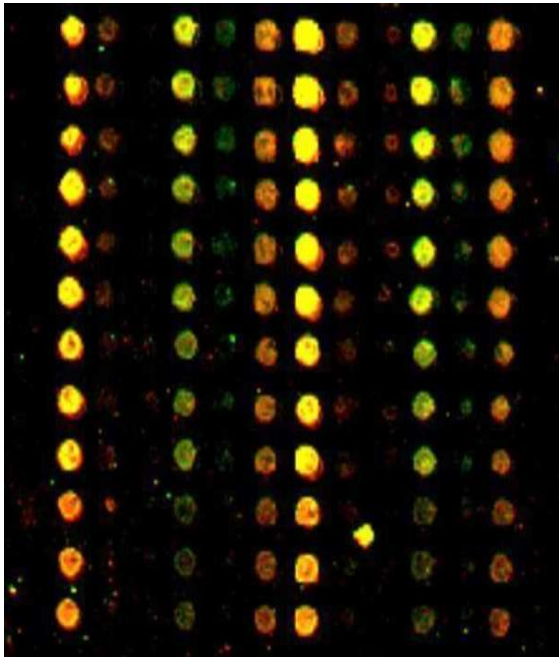


The Biological Context of Transcription Modules

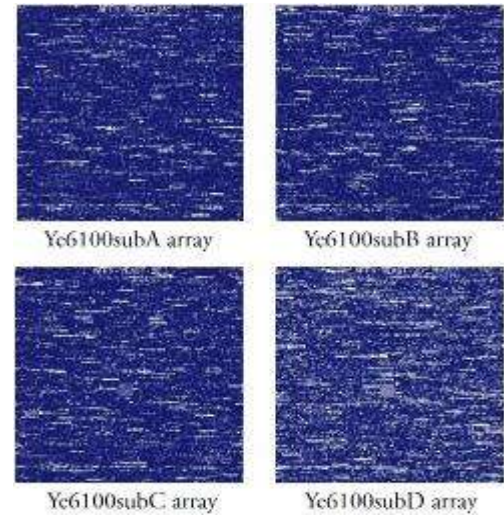


The primary goal of the iterative signature algorithm (ISA) is to attempt to gain novel biological insights from gene expression data

cDNA microarray



Affymetrix oligo array



Although multiple datasets exist, it is not always easy to correlate them in a statistically sound way that also makes sense biologically

Gene Expression data is the result of multiple interactions within a cell, and may not always be explained by simple models

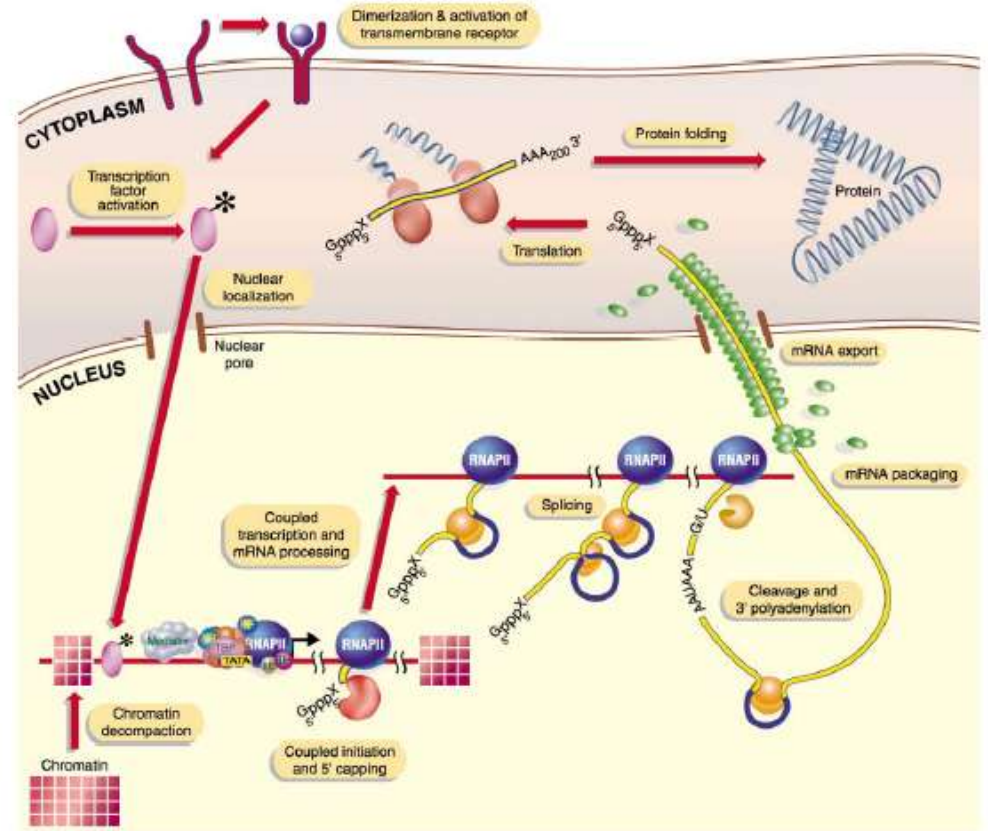
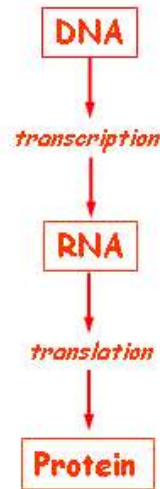
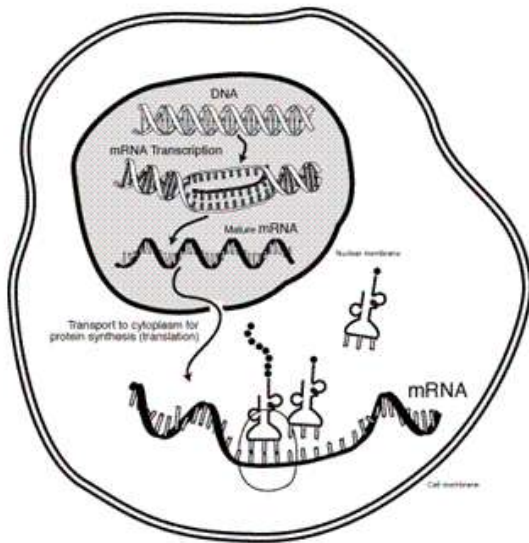
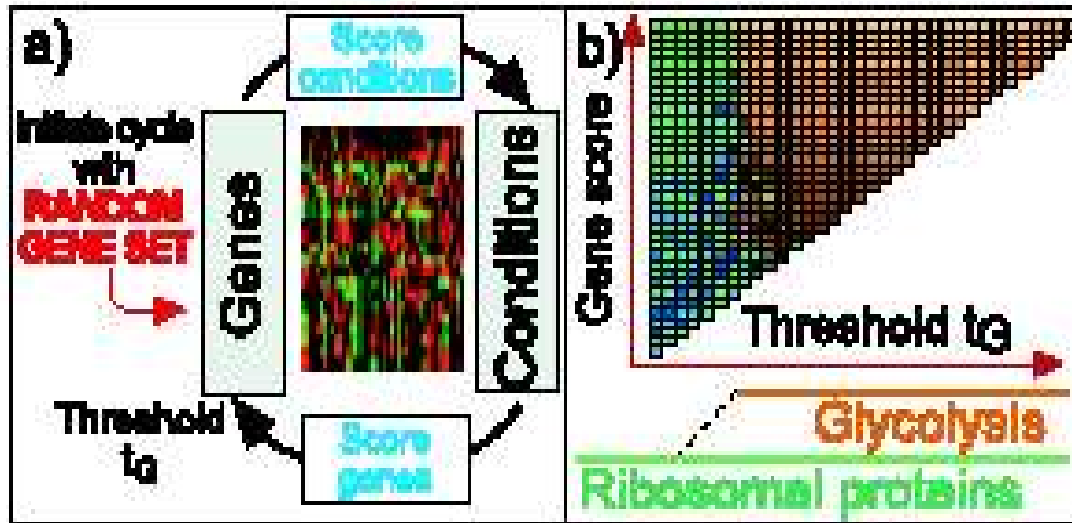


Figure 2. A Contemporary View of Gene Expression

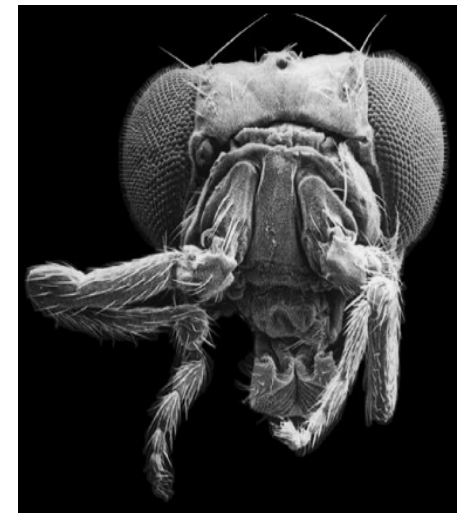
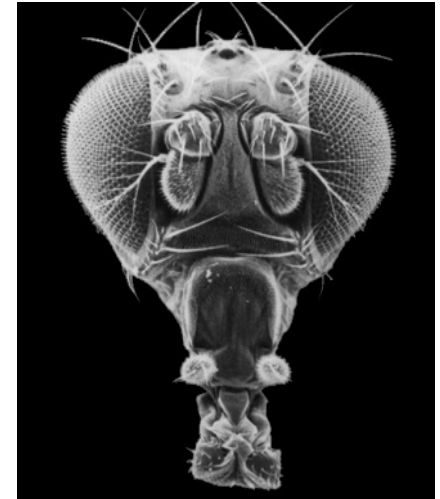
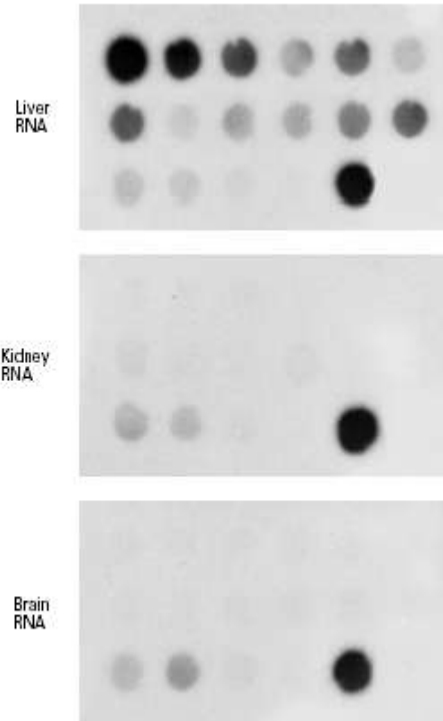
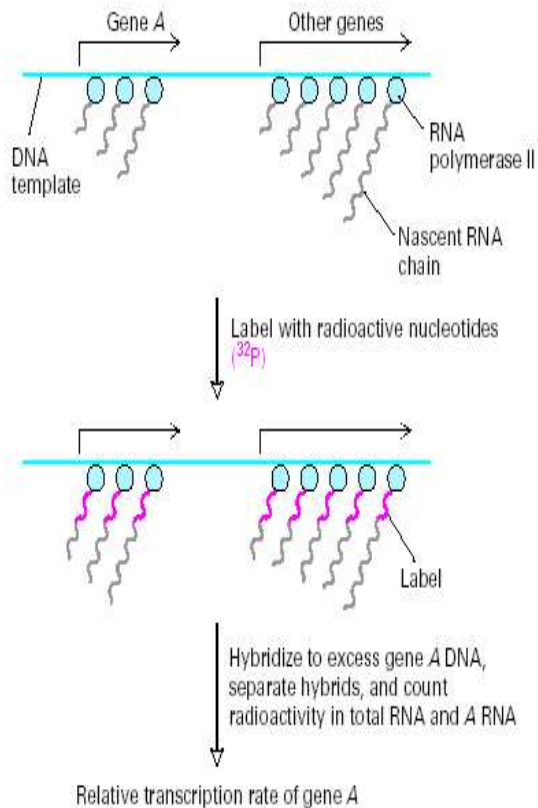
The method proposed by Ihmels *et al* is based upon the idea of a transcription module



The assumption is that correlations between genes and conditions is due to an underlying transcription network

Most of eukaryotic gene expression is regulated at the level of transcription

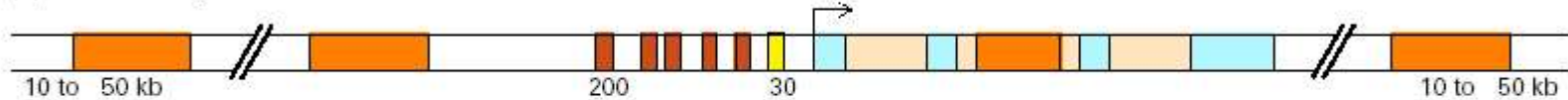
We know this based on early nuclear run on assays, and mutation experiments



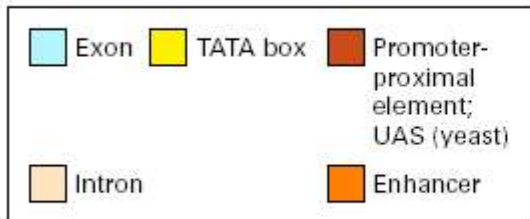
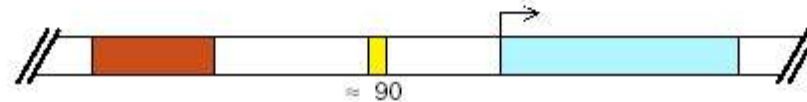
A key regulatory feature within transcription is the binding of transcription factors, which aid in recruiting the RNA polymerase

The binding of transcription factors can be regulated by *Cis* elements

(a) Mammalian gene



(b) *S. cerevisiae* gene

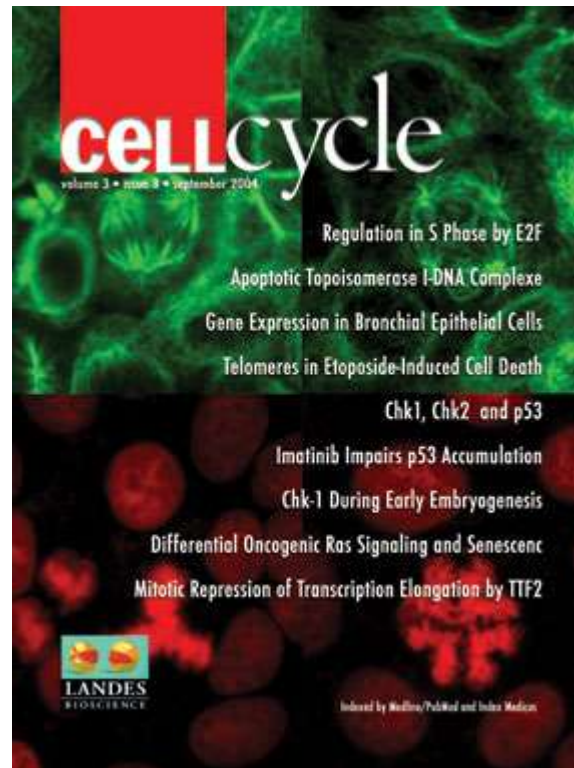


Since *Cis* elements are sequence based, we can often find them computationally

It makes sense biologically to group genes in overlapping modules with their correlated conditions

Transcription factors generally activate the expression of series of genes

Genes can be activated by more than one transcription factor



Application of the Iterative Signature Algorithm

Iterative Signature Algorithm (ISA)

- Defines the score of a set of genes and conditions.
- Iteratively refines the set of genes and conditions until a “stable” transcription module is obtained.

How Does an ISA Work?

- 1) Choose a random set of genes $G^{(0)}$
- 2) A uniform scoring to the genes is calculated
- 3) The subset of conditions for which the absolute score exceeds the conditional threshold t_c are selected and denoted as $C^{(0)}$

- 7) The set of genes whose score exceeds t_c are kept and denoted $G^{(1)}$

- 10) This procedure is repeated until $G^{(0)} = G^{(n+1)}$ and thus convergence has been reached

How Is The ISA Applied?

- Generate a large sample of input seeds.
- Find the fixed points corresponding to each seed through iterations.
- Collect the distinct fixed points to put the expression data into modules.

Principle Idea of ISA

Search for solutions of the *consistency equation*
in

$$\exists(t_C, t_G): \begin{cases} c_m = f_{t_C}(c_m^{proj}), \\ g_m = f_{t_G}(g_m^{proj}), \end{cases}$$

through a map defined by

$$c^{(n+1)} = f_{t_C}(E_G \cdot g^{(n)}),$$

$$g^{(n+1)} = f_{t_G}(E_C^T \cdot c^{(n+1)}).$$

Apply the maps iteratively

ISA Strategy

$$c^{(n+1)} = f_{t_C}(E_G \cdot g^{(n)}),$$

Reusing the $c^{(n)}$ as input to f_{t_G}
 output sets (i.e. $g^{(1)}$ and $c^{(1)}$)

$$g^{(n+1)} = f_{t_G}(E_C^T \cdot c^{(n+1)}).$$

Obtaining $c^{(2)}$ and $g^{(2)}$

Until $\{g^{(3)}, c^{(3)}\}$

converges

$$\{g^{(0)}, g^{(1)}, g^{(2)}, g^{(3)}, \dots\}$$

ISA Strategy

A 'fixed point' gene vector $\mathbf{g}^{(n^*)}$

Satisfying

$$\frac{|\mathbf{g}^{(*)} - \mathbf{g}^{(n)}|}{|\mathbf{g}^{(*)} + \mathbf{g}^{(n)}|} < \epsilon$$

Why Choose ISA ?

- Computation time is shortened.
 - Only multiplications of matrices
 - Very few iterations are needed to find fixed points
 - By using previous runs vs random input seeds can improve the algorithm

How to make sense of the massive expression data containing millions of numbers?

Approach

– **Iterative Signatures Algorithm (ISA)**

Method of ISA:

- Assign genes into context-dependent regulatory units.

– Introduce **Transcription Module**



» Defined by **self-consistent** regulatory unit



set of **Coregulated genes** → $g^{(m)}$

set of **Experimental condition** → $c^{(m)}$

ISA Focuses on:

1. Properties of individual co-regulated units

Resulting in:

- Linear computation time due to:
 - **Rigorous definition of transcription module allows for**

**All possible sets of genes and conditions
(in principle to be evaluated)**

Heuristic approach

Iteratively apply random set of genes(conditions)
(until self-consistency is met)

Use of threshold conditions

Selects genes that are well expressed

Eliminates unnecessary data

Conventional clustering Focus on:

1. Assign each gene to one cluster

Problem – genes may participate in several biological functions

Should be included in multiple clusters

2. Correlation in expression pattern measured over all condition

Problem – genes typically regulated only in specific experimental conditions.

- Exponential computation time
 - Optimizing all clusters simultaneously

Apply ISA to over 1000 expression profiles of yeast *Saccharomyces cerevisiae*

Compare results with commonly used clustering methods by,

1. Identifying all hexamers that are significantly overrepresented, P_{cer}
2. Repeat identification for 4 related yeast strains
3. Repeat identification for 4 related yeast strains

Where gene containing the hexamer in the promoter region is evaluated, $P_{all}(h)$

Assumption is certain sequence pattern is conserved then is an important regulatory sequence

5. Using $\text{BFM} = -\log_{10} P_{all}(h_{sig})$
A conservation p-value P_{all}

Of most significant hexamer h_{sig}

Only gene sets with 20 to 400 genes and only overlapping clusters with the largest BFM considered

Identification through ISA

Identify the transcription module by Iteratively applying the signature algorithm:

1. Use random set of reference genes G_0 and assign a uniform score to its genes.

Score all conditions in the data set by:

*Averaging the expression of each gene over the module conditions,

$$s_c = \langle s_c E_C^{gc} \rangle_{c \in C_m}$$

- * Keep the set of conditions C_0 with absolute scores $|S_c|$ that exceed a threshold condition t_c

2. Use reference set condition C_0

Score all genes within C_0 :

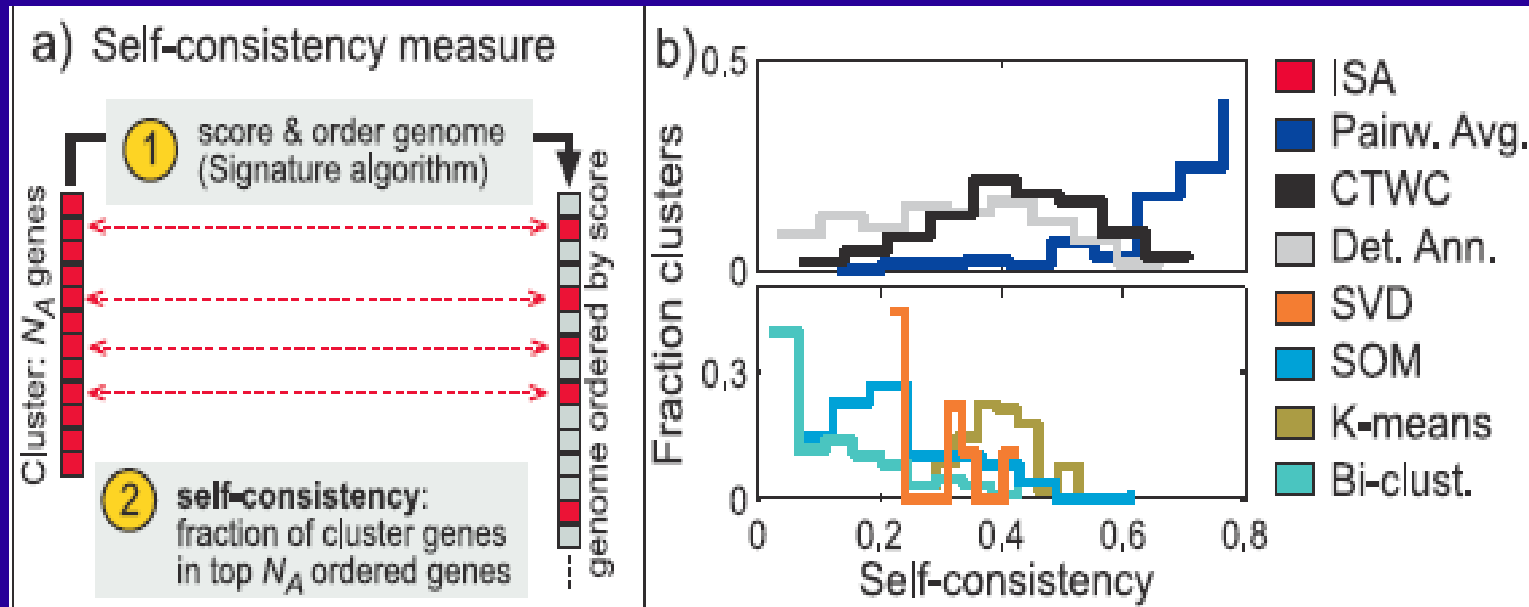
$$s_g = \langle s_g E_G^{gc} \rangle_{g \in G_m}$$

- * Keep the set of gene with scores s_g that exceed a threshold t_g

Do all modules satisfy the self-consistency criterion?

And

Do clusters generated by common methods satisfy the self-consistency criterion?



Self-consistent modules are Transcription modules

Comparison to other available methods (using same dataset)

Hierarchical clustering (pair-wise average-linkage) – approximately self-consistent

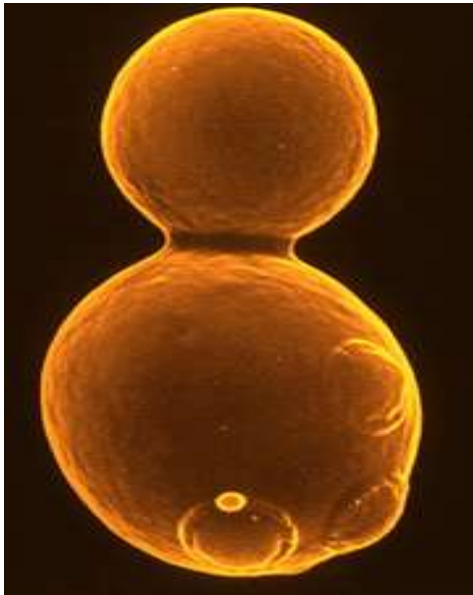
Deterministic annealing

K-means

Two bi-clustering methods

Bergmann *et al* 1st obtained some preliminary results by using the ISA on yeast expression data

They used more than 1000-DNA chip experiments

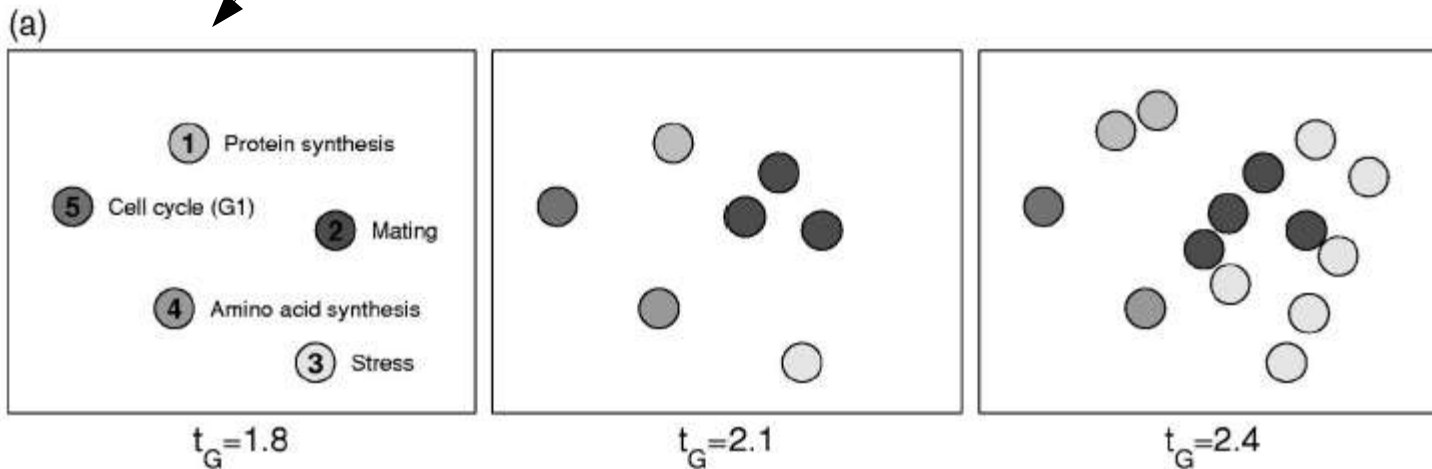


experiments were composed primarily of cell cycle and stress perturbations

In applying different gene threshold levels, Bergmann *et al* altered the resolution of the subsequent transcription modules

~20,000 initial gene sets were used for each t_G value in order to find fixed points

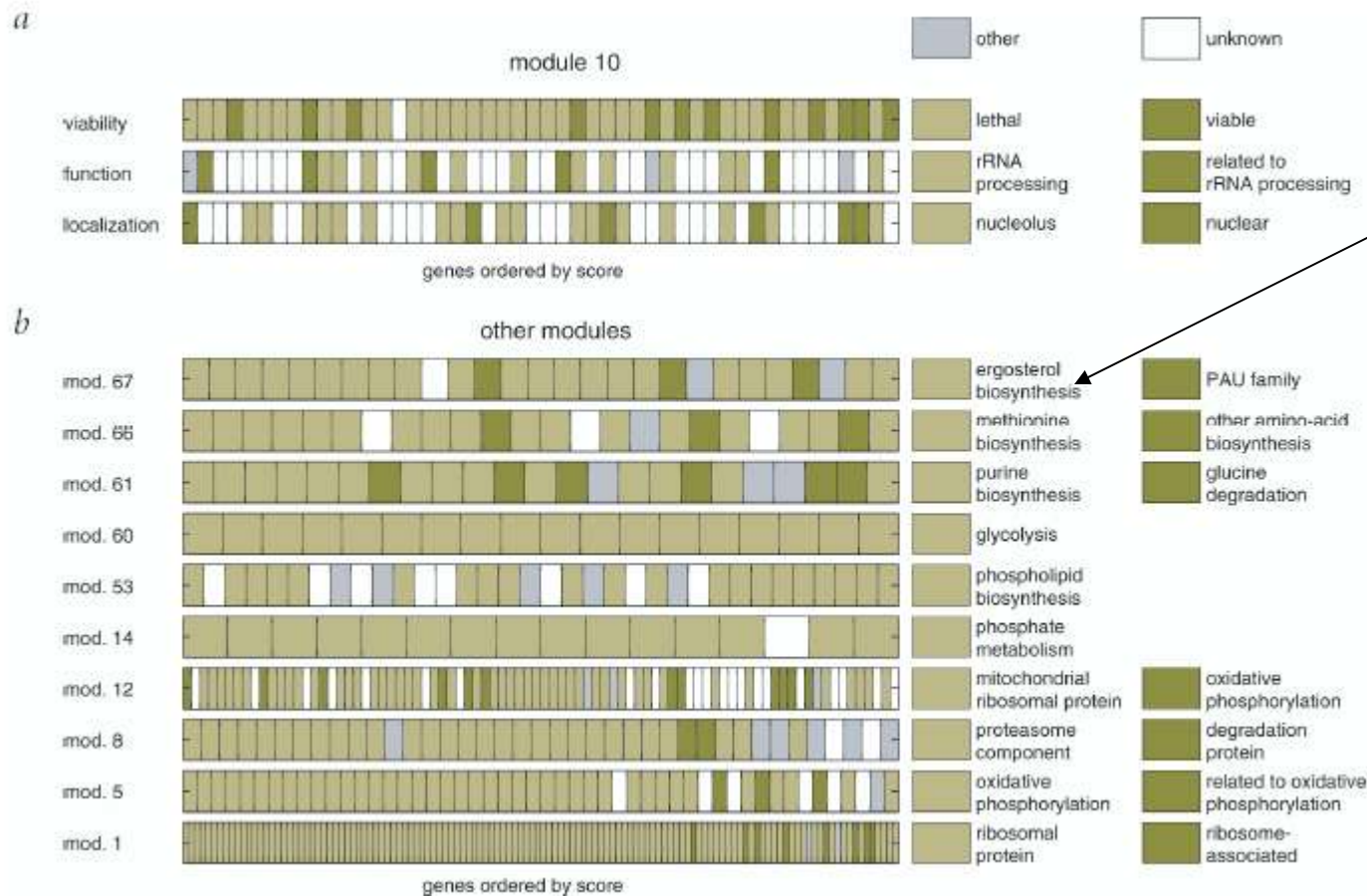
At the lowest level,
they recovered the central functions



As t_G increases, the number of modules increases and the size of those modules decreases.

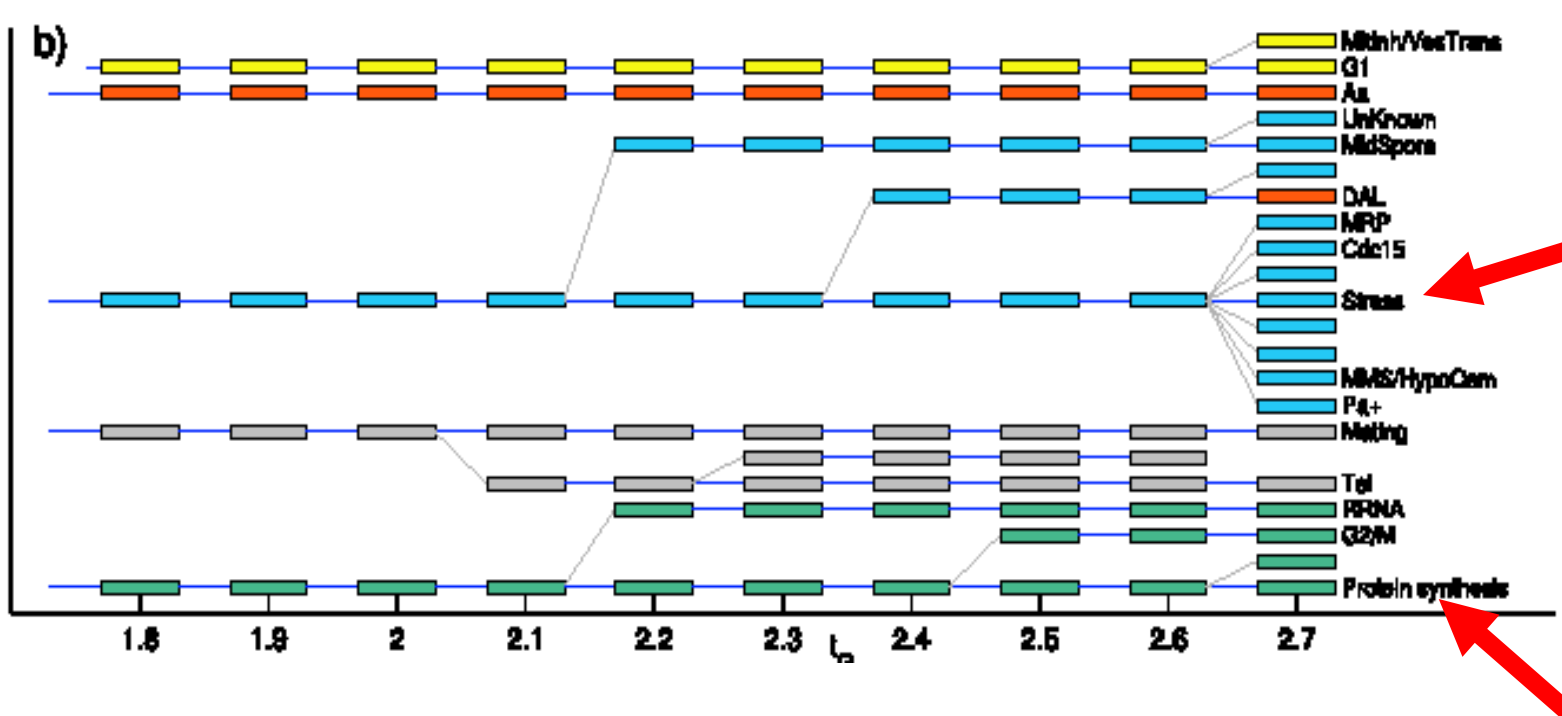
The modules recovered using this unsupervised approach were essentially the same as those obtained previously while using prior knowledge

Example of some of the modules constructed in previous work



Note the prevalence of biosynthesis

Transcription modules recovered using the unsupervised ISA approach over gene threshold values indicate that certain modules are conserved



The protein synthesis and stress modules remain fixed across all thresholds tried

They take this to mean that these modules represent the backbone of the transcriptional network

Recall that stress response and cell cycle experiments dominate the data analyzed

The gene threshold value is extremely important in obtaining the correct result

When the threshold is too low

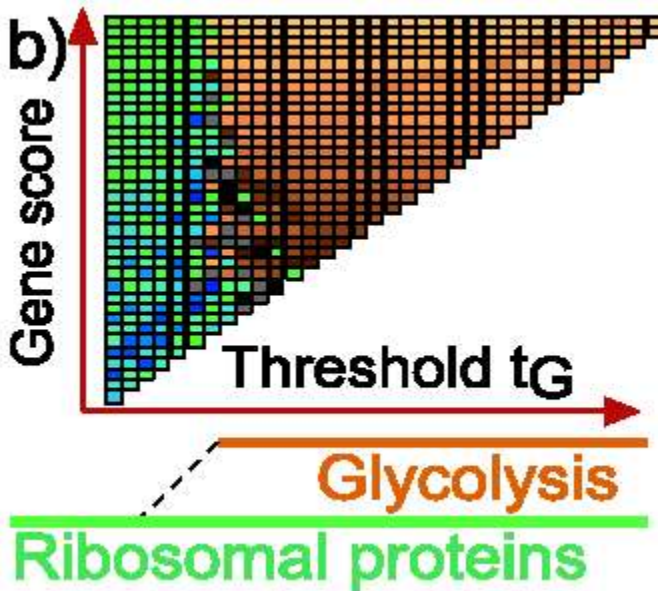


Module pulled to different fixed point

When the threshold is too high



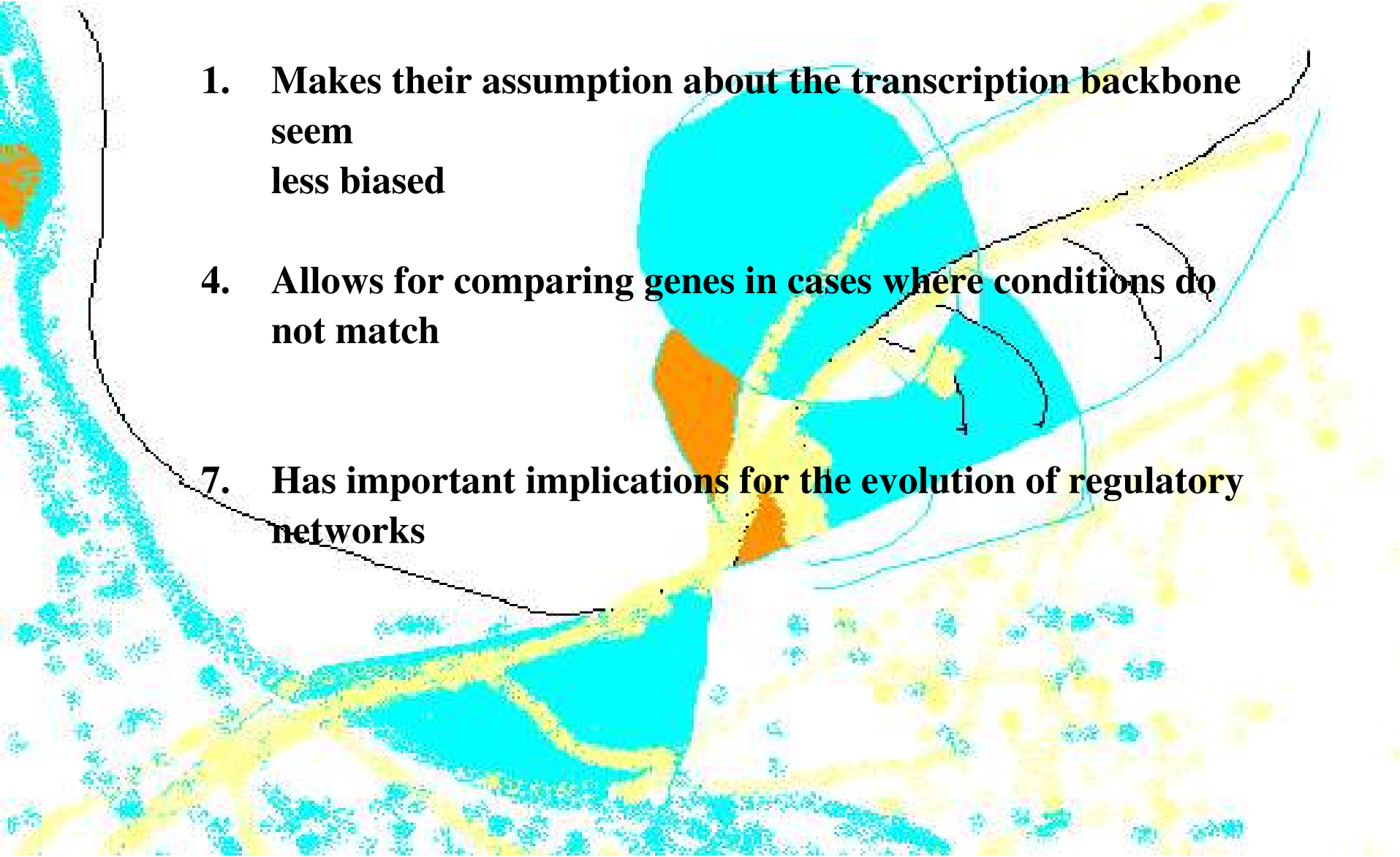
Core modules dissipate



However, the same was not true for the conditions threshold

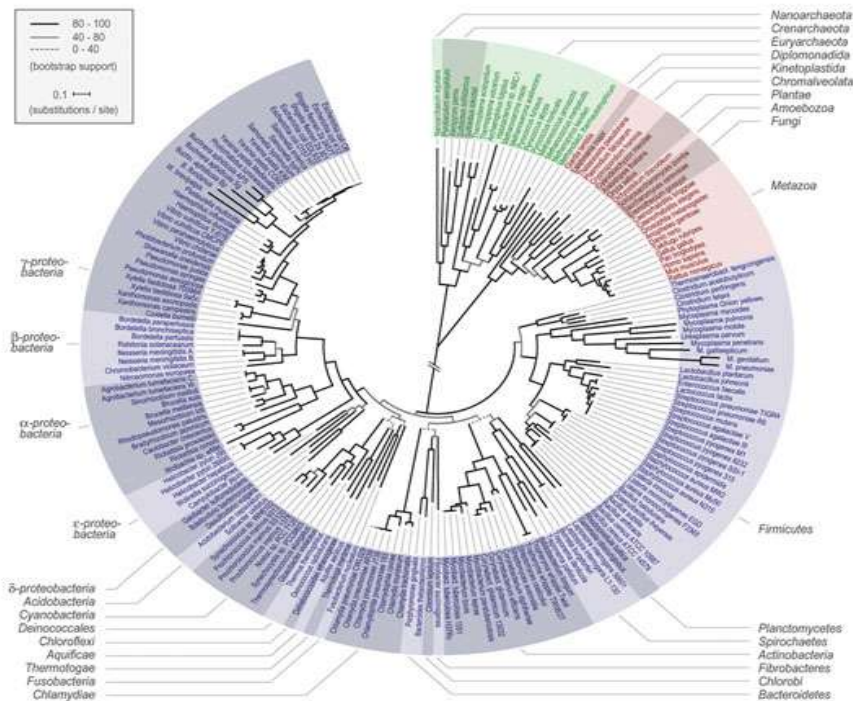
What does it imply biologically for the modules to form with less sensitivity to the threshold for conditions than it does for the genes?

- 1. Makes their assumption about the transcription backbone seem less biased**
- 4. Allows for comparing genes in cases where conditions do not match**
- 7. Has important implications for the evolution of regulatory networks**



Bergmann *et al* go on to use the ISA algorithm in a comparative systems biology study

Their goal is to combine sequence data with expression data within an evolutionary context



Genomic
sequence

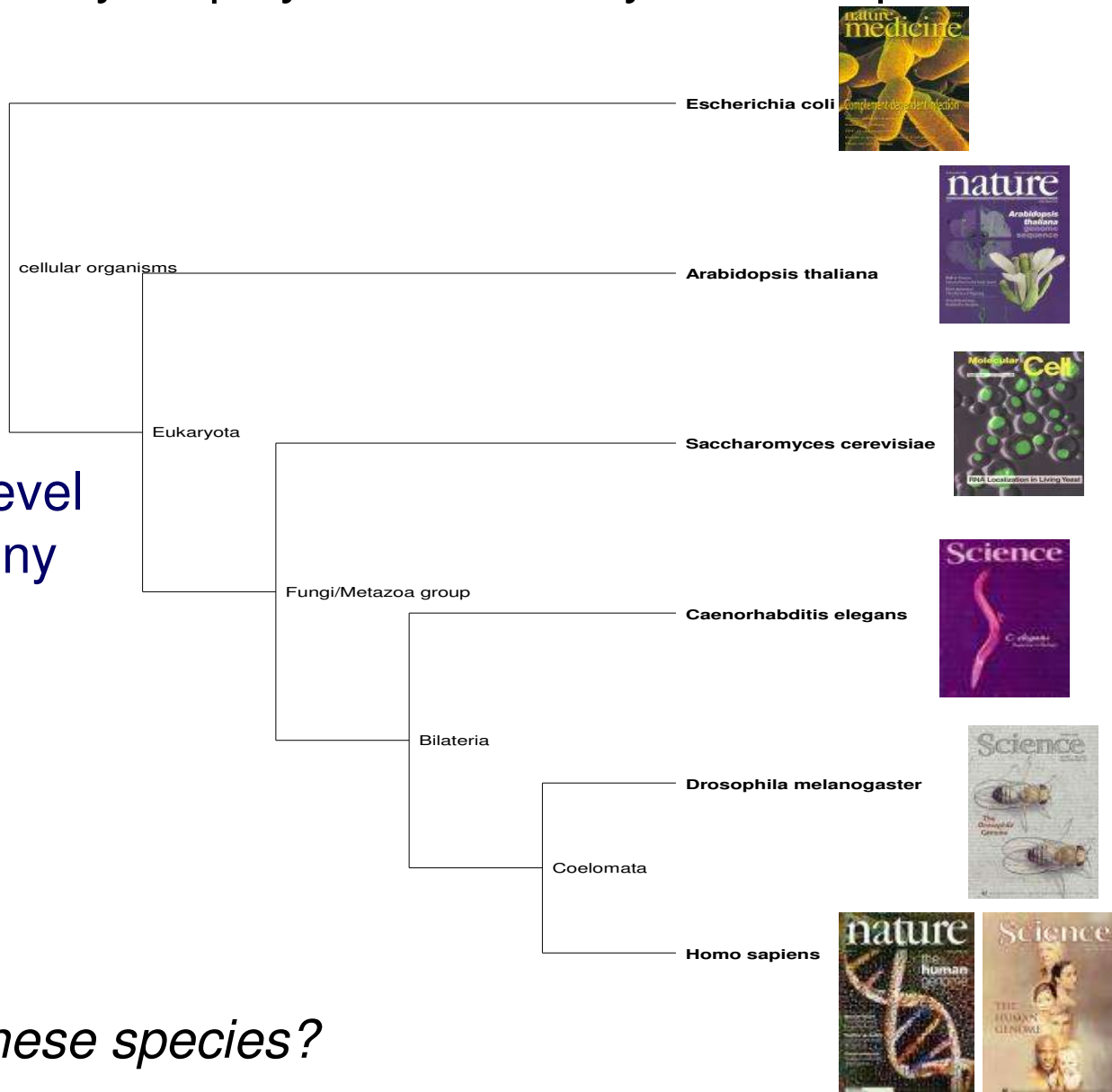


Organism form and
function

Current version of the Tree of Life

They employ 6 evolutionary distant species in their study

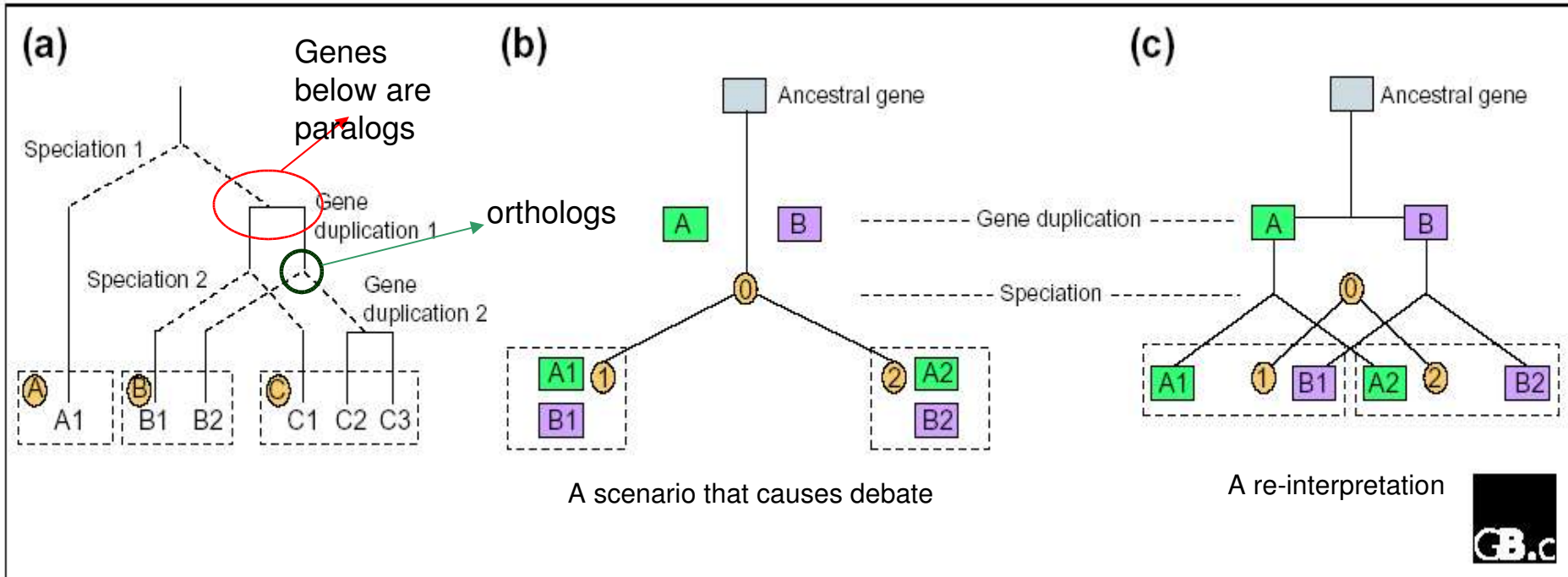
Deep Level phylogeny



Why these species?

A brief note about terminology

Homologs, orthologs, and paralogs, Oh my!



Orthologs and Paralogs are the result of divergence from a common ancestor. *Neither implies a functional relationship.*

Goal

- Comparative study of large datasets of expression profiles from six evolutionarily distant organisms:



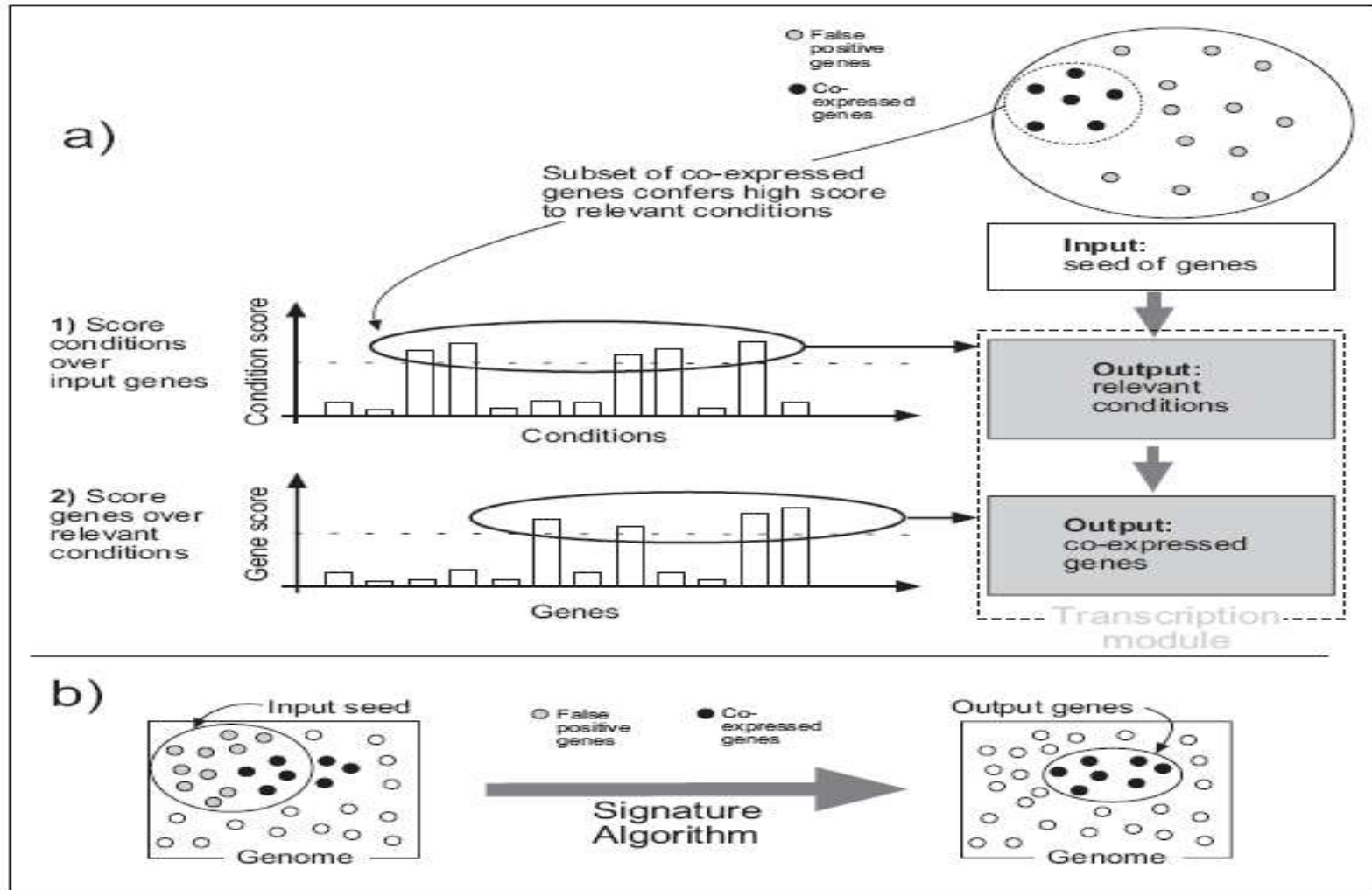
Organism	Genes	Conditions
<i>S. cerevisiae</i>	6,206	1,011
<i>E. coli</i>	4,009	83
<i>A. thaliana</i>	5,095	131
<i>C. elegans</i>	18,372	547
<i>D. melanogaster</i>	4,040	75
<i>H. sapiens</i>	6,184	153



Signature Algorithm

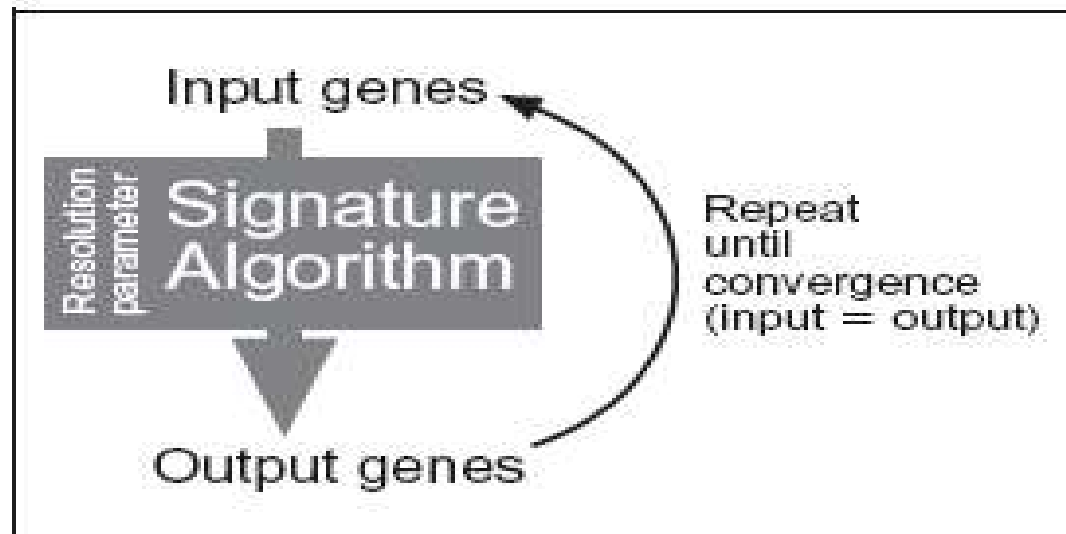
- Designed to identify co-regulated genes with experimental conditions
- Start with a set of input genes that partially overlap with a TM
- Within this set, the algorithm identifies genes that are co-expressed under experimental conditions
- Reveals additional genes that display similar expression patterns
- Input genes are chosen according to some common feature

Signature Algorithm

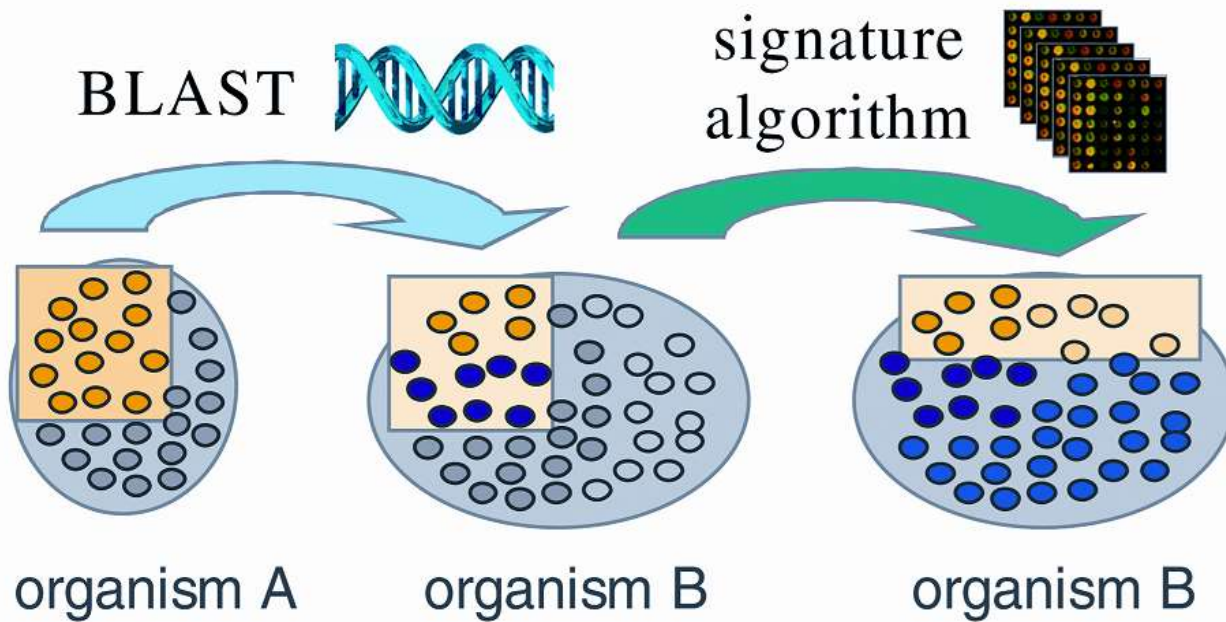


The Iterative Signature Algorithm

- Designed to reveal the hierarchies of co-regulatory units
- Output genes are re-used as input until convergence occurs



Signature Algorithm



Genomic properties of different organisms

attributed to differences in gene expression data

Use large datasets of expression profiles from 6 evolutionarily distant organisms obtained from different sources

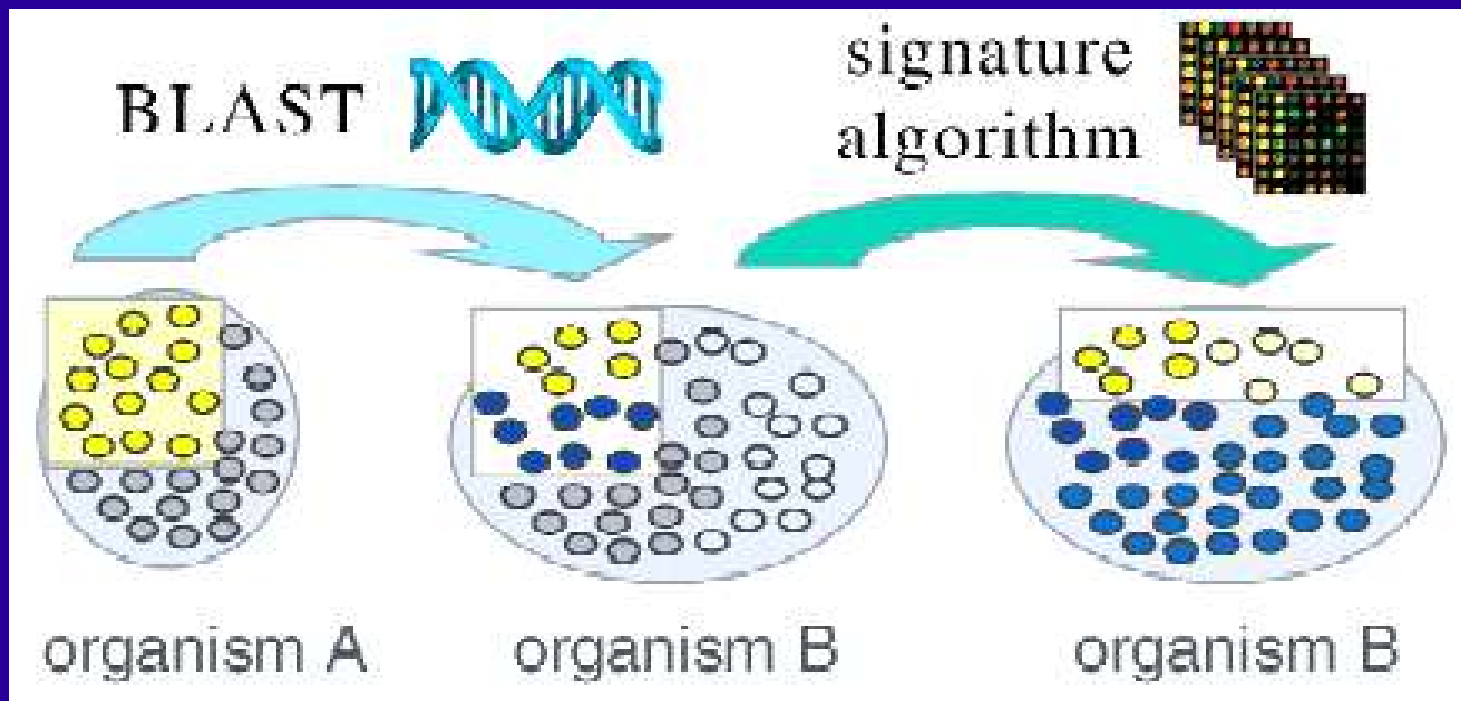
Table 1. Large-Scale Expression Data Used in This Study

Organism	Genes	Conditions
<i>S. cerevisiae</i>	6,206	1,011
<i>E. coli</i>	4,009	83
<i>A. thaliana</i>	5,095	131
<i>C. elegans</i>	18,372	547
<i>D. melanogaster</i>	4,040	75
<i>H. sapiens</i>	6,184	153

Interested in finding homologous genes between organisms

biology a part or organ that has the same evolutionary origin as another but differs in function, for example, a bird's wing in relation to the fin of a fish

Here we construct 'homologue modules' – which contain the respective homologs in the other organism

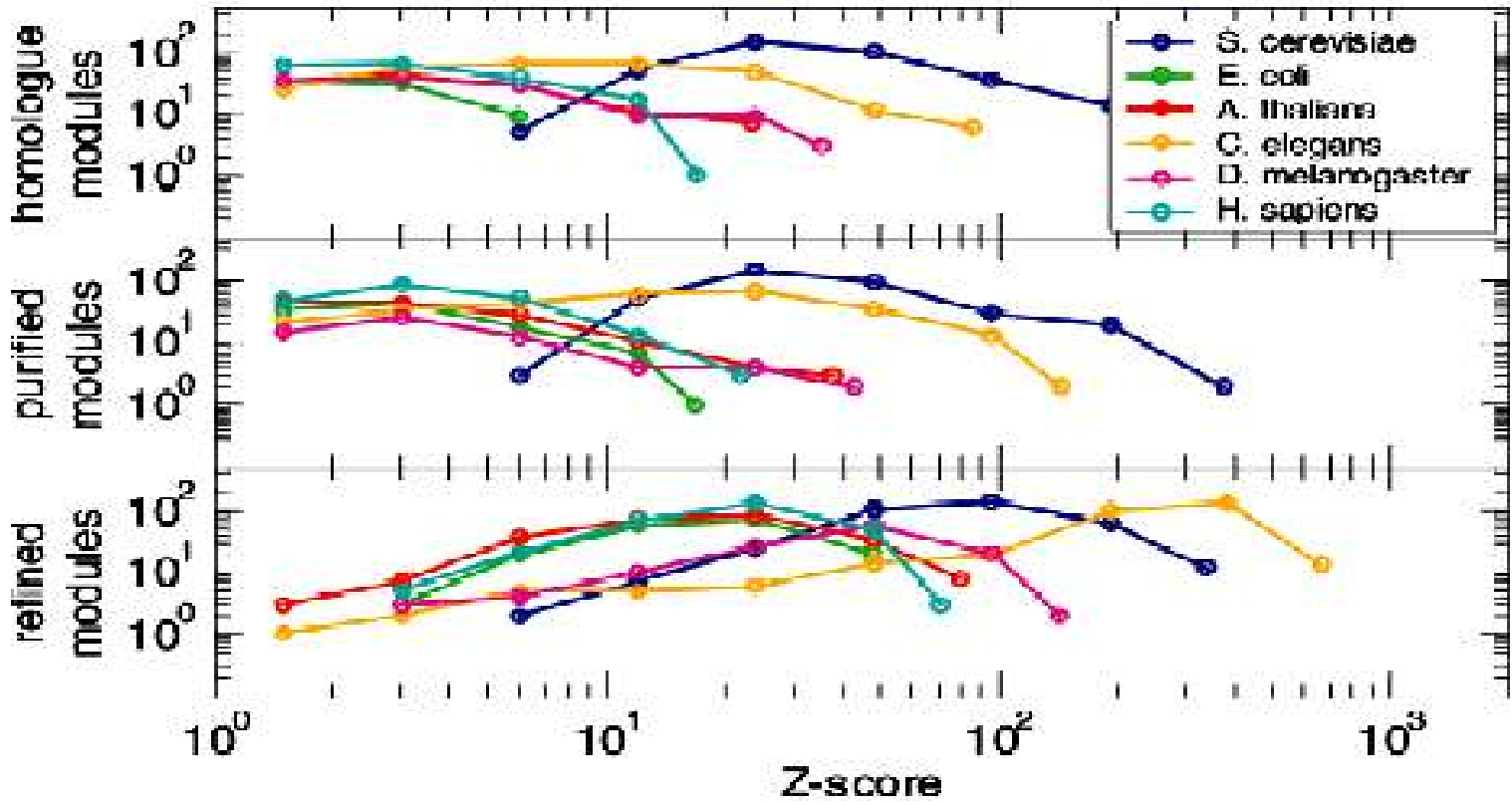


Here focus is placed on

1. Organism A – group of functionally related genes
2. Organism B – simultaneously identify respective homologues
3. Then examine which homologues are coexpressed

Here ISA is used to examine conservation of coexpression across groups of genes that are associated with same cellular function

Using Expression Data to identify and Refine Sequence-Based Functional Assignments



‘Purified module’ – contain only ‘homologues modules’ that are coexpressed or homologues that are not coexpressed are rejected

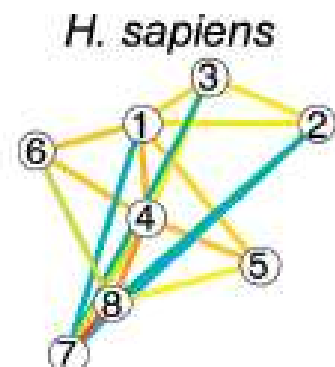
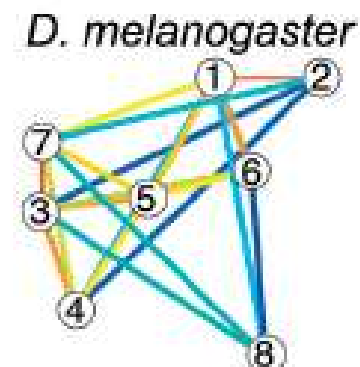
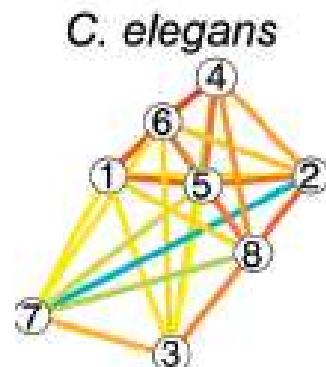
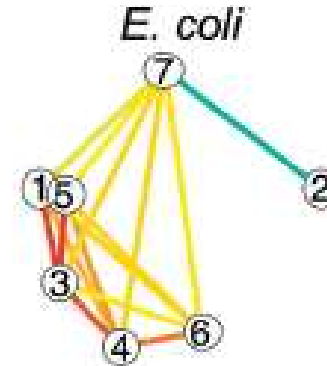
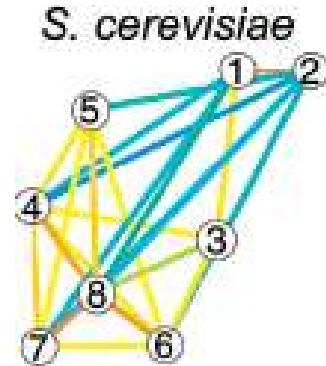
‘Refined modules’ – added further coexpression

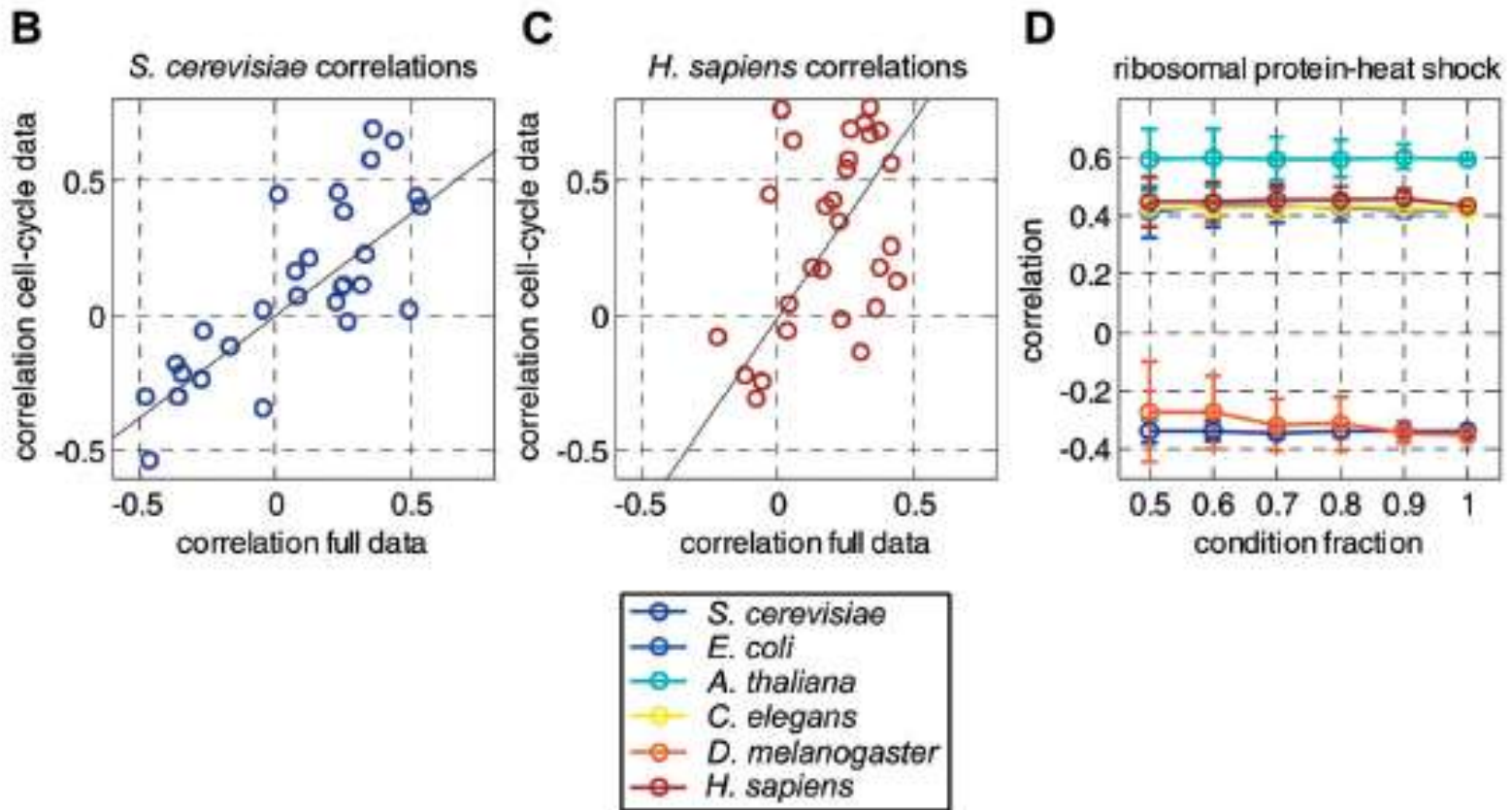
Regulatory Relations between Modules

Selection of 8 transcription modules whose function is known in yeast

Used to generate the 'refined homologue modules' in other 5 organisms. Each module associated with a 'condition profile' created by ISA

- ① = ribosomal protein
- ② = rRNA processing
- ③ = glycolysis
- ④ = heat shock
- ⑤ = MRP
- ⑥ = secreted protein
- ⑦ = peroxide
- ⑧ = proteasome





Despite sparseness of the data

correlation of modules are not effected to changes in the gene content of most refined modules

Observations from transcription modules

1. Several functional groups were repeatedly identified as coexpressed include modules related to core biological functions

Further comparing of Global Features of Gene Expression Networks

Using **Topological properties** of expression data – represent data by an undirected ‘expression network’

2. Genes associated with only one module have connectivity

Reflecting size of module – larger clustering coefficient

3. Genes belonging to several modules are correlated with a larger number of genes

Reflecting size of module – smaller clustering coefficient

Can the variations in the regulatory relations among organisms arise from the use of unrelated sets of experimental conditions?

1. Restrict both human and yeast expression data to the cell cycle experiments

Found correlations between modules did not change qualitatively due to this restriction

2. Examine sensitivity of results of number of conditions used

Removal of up to 50% of all conditions did not considerably change the gene content of refined modules