nature genetics

Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data

Eran Segal[1,6], Michael Shapira[2], Aviv Regev[3,5,6], Dana Pe'er[4,6], David Botstein[2], Daphne Koller[1] & Nir Friedman[4]

Much of a cell's activity is organized as a network of interacting modules: sets of genes coregulated to respond to different conditions. We present a probabilistic method for identifying regulatory modules from gene expression data. Our procedure identifies modules of coregulated genes, their regulators and the conditions under which regulation occurs, generating testable hypotheses in the form 'regulator *X* regulates module *Y* under conditions *W*'. We applied the method to a *Saccharomyces cerevisiae* expression data set, showing its ability to identify functionally coherent modules and their correct regulators. We present microarray experiments supporting three novel predictions, suggesting regulatory roles for previously uncharacterized proteins.

The complex functions of a living cell are carried out through the concerted activity of many genes and gene products. This activity is often coordinated by the organization of the genome into regulatory modules, or sets of coregulated genes that share a common function. Such is the case for most of the metabolic pathways as well as for members of multiprotein complexes. Identifying this organization is crucial for understanding cellular responses to internal and external signals. Genome-wide expression profiles[1–3] provide important information about these cellular processes. Yet, the regulatory mechanisms of a cell are far from transparent in these data. Current approaches for analyzing gene expression data[4–8] allow the identification of groups of co-expressed genes. But the regulatory programs of these groups can be suggested only indirectly, for example, by finding common *cis*-regulatory binding sites in the upstream regions of genes in each group[2,9–12].

Here, we present the module networks procedure, a method based on probabilistic graphical models[13] for inferring regulatory modules from gene expression data. In our framework, a regulatory module is a set of genes that are regulated in concert by a shared regulation program that governs their behavior. A regulation program specifies the behavior of the genes in the module as a function of the expression level of a small set of regulators. Similar to previous methods for inferring regulatory networks from gene expression data[14–17], our approach relies on the assumption[18] that the regulators are themselves transcriptionally regulated, so that their expression profiles provide information about their activity level. Clearly, this assumption is sometimes violated, a common instance being transcription factors that are regulated post-translationally. In some cases, however, we can obtain additional evidence about regulation by considering the expression levels of those signaling molecules that may have an indirect transcriptional impact.

Our automated procedure (**Fig. 1**) takes as input a gene expression data set and a large precompiled set of candidate regulatory genes for the corresponding organism (not dependent on the data set), containing both known and putative transcription factors and signal transduction molecules. Given these inputs, the algorithm searches simultaneously for a partition of genes into modules and for a regulation program (**Fig. 2**) for each module that explains the expression behavior of genes in the module. The regulation program of a module specifies the set of regulatory genes that control the module and the mRNA expression profile of the genes in the module as a function of the expression of the module's regulators (**Fig. 2**). The procedure gives as output a list of modules and associated regulation programs. These identify groups of coregulated genes, their regulators, the behavior of the module as a function of the regulators' expression and the conditions under which regulation takes place.

We applied our method to a *S. cerevisiae* gene expression data set consisting of 2,355 genes and 173 arrays[3]. With few exceptions, each of the inferred modules (46 of 50) contained a functionally coherent set of genes. Together the modules spanned a wide variety of biological processes including metabolic pathways (for example, glycolysis), various stress responses (for example, oxidative stress), cell cycle–related processes, molecular functions (for example, protein folding) and cellular compartments (for example, nucleus). Most modules (30 of 50) included genes previously known to be regulated by the module's predicted regulators. Many modules (15 of 50) had a match between a predicted regulator and its known *cis*-regulatory binding motif (that is, a statistically significant number

[1]Computer Science Department, Stanford University, Stanford, California, 94305, USA. [2]Department of Genetics, Stanford University School of Medicine, Stanford, California, 94305, USA. [3]Department of Cell Research and Immunology, Tel Aviv University & Computer Science Department, Weizmann Institute, Israel. [4]School of Computer Science & Engineering, Hebrew University, Jerusalem, 91904, Israel. [5]Present address: Bauer Center for Genomics Research, Harvard University, Cambridge, Massachusetts, USA. [6]These authors contributed equally to this manuscript. Correspondence should be addressed to E.S. (eran@cs.stanford.edu) or D.K. (koller@cs.stanford.edu).

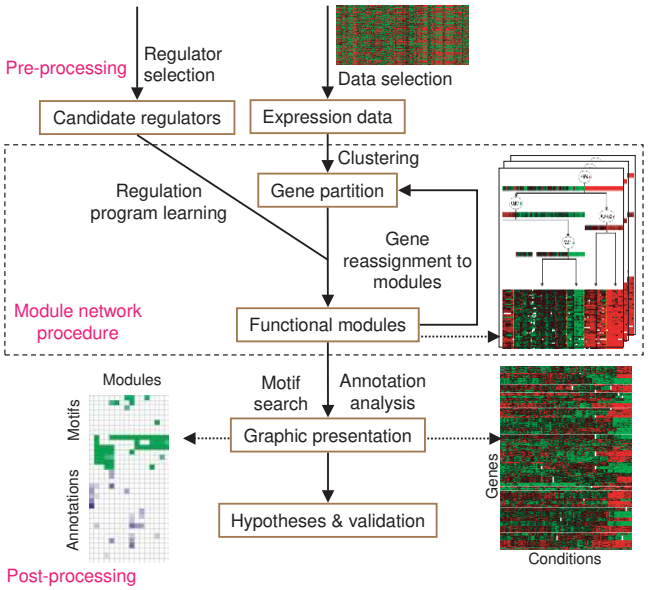


Figure 1 Overview of the module networks algorithm and evaluation procedure. The procedure takes as input a data set of gene expression profiles and a large precompiled set of candidate control genes. The method itself (dotted box) is an iterative procedure that determines both the partition of genes to modules and the regulation program (right icon in dotted box) for each module. In a post-processing phase, modules are tested for enrichment of gene annotations and *cis*-regulatory binding site motifs.

of the module's genes contained the known motif in their upstream regions). Overall, our results provide a global view of the yeast transcriptional network, including many instances in which our method identifies known functional modules and their correct regulators, showing its ability to derive regulation from expression.

A regulation program specifies that certain genes regulate certain processes under certain conditions. Our method thus generates detailed, testable hypotheses, suggesting specific roles for a regulator and the conditions under which it acts. We tested experimentally the computational predictions for three putative regulators with unknown functions (a transcription factor and two signaling molecules). Our method's results make specific predictions regarding the conditions under which these regulators operate. Using microarray analysis, we compared the transcriptional responses of the respective genetically disrupted strains with their congenic wild-types under these conditions. Deletion of each of the three regulators caused a marked impairment in the expression of a substantial fraction of their computationally predicted targets, supporting the method's predictions and giving important insight regarding the function of these uncharacterized regulators.

RESULTS

We compiled a list of 466 candidate regulators and applied our procedure to 2,355 genes in the 173 arrays of the yeast stress data set[3], resulting in automatic inference of 50 modules. We analyzed each of the resultant modules (**Fig. 1**) using a variety of external data sources, evaluating the functional coherence of its gene products and the validity of its regulatory program.

Sample modules

We first present in detail several of the inferred modules, selected to show the method's ability to reproduce diverse features of regulatory programs.

The respiration module (**Fig. 3**) is a clear example of a predicted module and of the validation process. It consists primarily of genes encoding respiration proteins (39 of 55) and glucose-metabolism regulators (6 of 55). The inferred regulatory program specifies the Hap4 transcription factor as the module's top (activating) regulator, primarily under stationary phase (a growth phase in which nutrients, primarily glucose, are depleted). This prediction is consistent with the known role of Hap4 in activation of respiration[1,19]. Indeed, our post-analysis detected a Hap4-binding DNA sequence motif (bound by the Hap2/3/4/5 complex) in the upstream region of 29 of 55 genes in the module ($P < 2 \times 10^{-13}$). This motif also appears in non-respiration genes (mitochondrial genes and glucose-metabolism regulators), which, together with their matching expression profiles, supports their inclusion as part of the module. When Hap4 is not induced, the

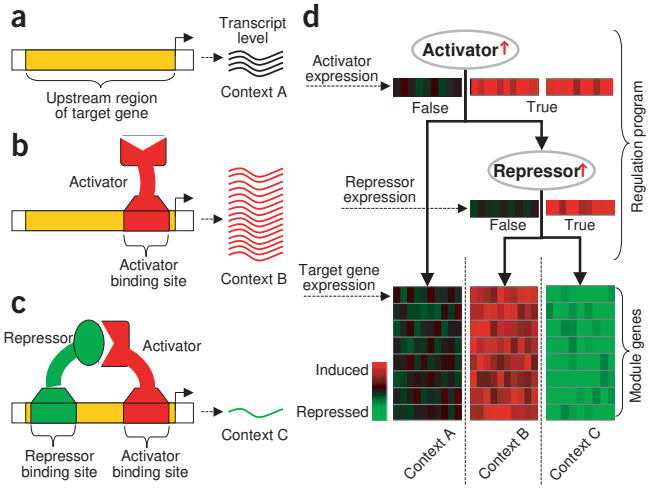


Figure 2 Regulation programs represent context-specific and combinatorial regulation. Shown is a scheme depicting three distinct modes of regulation for a group of genes. (**a**) Context A. Genes in the module are not under transcriptional regulation and are in their basal expression level. (**b**) Context B. An activator gene is upregulated and, as a result, binds the upstream regions of the module genes, thereby inducing their transcription. (**c**) Context C. A repressor gene is upregulated and, as a result, blocks transcription of the genes in the module, thereby reducing their expression levels. (**d**) A regulation tree or program can represent the different modes of regulation described above. Each node in the tree consists of a regulatory gene (for example, 'Activator') and a query on its qualitative value, in which an upward arrow (red) denotes the query "is gene upregulated?" and a downward arrow (green) denotes the query "is gene downregulated?". Right branches represent instances for which the answer to the query in the node is 'true'; left branches represent instances for which the answer is 'false'. The expression of the regulatory genes themselves is shown below their respective node. Each leaf of the regulation tree is a regulation context (bordered by black dotted lines) as defined by the queries leading to the leaf. The contexts partition the arrays into disjoint sets, where each context includes the arrays defined by the queries of the inputs that define the context. In context A, the activator is not upregulated and the genes in the module are in their basal expression level (left leaf). In contexts B and C, the activator is upregulated. In context C, the repressor is also upregulated and the module genes are repressed (right leaf). In context B, the repressor is not upregulated and the activator induces expression of the module genes (center leaf). This regulation program specifies combinatorial interaction; for example, in context B, the module genes are upregulated only when the activator is upregulated but the repressor is not.

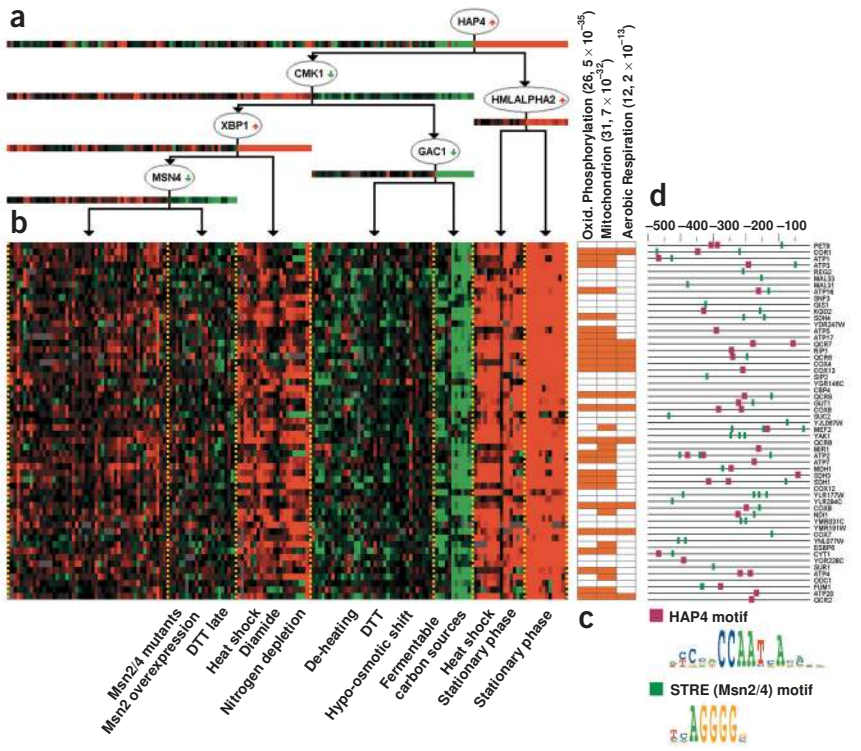


Figure 3 The respiration and carbon regulation module (55 genes). (**a**) Regulation tree/program. Each node in the tree represents a regulator (for example, Hap4) and a query of its qualitative value (for example, red upward arrow next to Hap4 for "is Hap4 upregulated?"). The expression of the regulators themselves is shown below their respective node. (**b**) Gene expression profiles. Genes, rows; arrays, columns. Arrays are arranged according to the regulation tree. For example, the rightmost leaf includes the arrays in which both Hap4 and HMLAlpha2 are upregulated. Contexts that consist primarily of one or two types of experimental conditions are labeled. (**c**) Significant annotations. Colored entries indicate genes with the respective annotation. The most significantly enriched annotations for this module were selected for display (the number of annotated genes and the calculated *P* value for the enrichment of each annotation are shown in parentheses). Note the enrichment of three annotations representing a biochemical process, cellular compartment and physiological process, respectively, all relating to cellular respiration. (**d**) Promoter analysis. Lines represent 500 bp of genomic sequence located upstream to the start codon of each of the genes; colored boxes represent the presence of *cis*-regulatory motifs located in these regions. Note the enrichment of both the HAP4 motif (purple) and the stress response element (STRE; green), recognized by Hap4 and Msn4, respectively, supporting their inclusion in the module's regulation program.

module is activated more mildly or is repressed. The method suggests that these changes are regulated by other regulators, such as the protein phosphatase type 1 regulatory subunit Gac1 and the transcription factor Msn4. Indeed, the stress response element (STRE), recognized by Msn4, appears in the upstream region of 32 of 55 genes in the module ($P < 10^{-3}$), as well as in those of many of the genes containing the Hap4 motif (17 of 29 genes; $P < 7 \times 10^{-10}$), supporting our placement of both regulators in one control program.

The nitrogen catabolite repression module (**Supplementary Fig. 1** online) shows the ability of our method to capture an entire cellular response whose genes participate in diverse metabolic pathways and cellular roles (12 of 29 in allantoin and urea metabolism, 5 of 29 in amino-acid metabolism and 6 of 29 in sulfur or methionine metabolism), all of which relate to the process by which the yeast uses the best available nitrogen source. Gat1 is suggested as the key (activating) regulator of this module, further supported by the presence of the GATA motif, the known binding sequence for Gat1, in the upstream region of 26 of 29 genes ($P < 10^{-17}$). This module also shows that the method can identify context-specific regulation, as the similarity in expression of genes in the module is mostly pronounced in stationary phase (17 of 22 experiments; $P < 10^{-4}$), amino-acid starvation (5 of 5; $P < 9 \times 10^{-5}$) and nitrogen depletion (10 of 10; $P < 8 \times 10^{-9}$), all of which are conditions in which using alternative nitrogen sources is crucial. Two additional known regulators involved in this response, Uga3 and Dal80, are suggested as members, rather than regulators, of the module.

The galactose metabolism module (**Supplementary Fig. 2** online) illustrates our method's ability to identify small expression signatures, as the module consisted of only four Gal4-regulated genes and predicted Gal4 as a regulator, with a predicted regulatory role that includes activation in galactose-containing medium.

The energy, osmolarity and cAMP signaling module (**Supplementary Fig. 3** online) shows that our method can identify regulation by proteins other than transcription factors, as the top predicted regulator was Tpk1, a catalytic subunit of the cAMP dependent protein kinase (PKA). This prediction is supported by a recent study[20] showing that the expression of several genes in the module (for example, Tps1) is strongly affected by Tpk1 activity in osmotic stress, which was among the conditions predicted by the method to be regulated by Tpk1. Further support is given by the presence of the STRE motif, known to be bound by transcription factors that are regulated by Tpk1 (ref. 20), in the upstream region of most genes in the module (50 of 64; $P < 3 \times 10^{-11}$), often in combination with other motifs bound by Tpk1-modulated transcription factors, such as Adr1 (37 of 64; $P < 6 \times 10^{-3}$) and Cat8 (26 of 64; $P < 2 \times 10^{-3}$). Our method suggests that Tpk1 is an activator of the module, however, in contrast to its known role as a repressor[21]. We discuss this discrepancy below.

Evaluation of module content and regulation programs

We evaluated all 50 modules to test whether the proteins encoded by genes in the same module had related functions. We scored the functional/biological coherence of each module (**Table 1**) according to the percentage of its genes covered by annotations significantly enriched in the module ($P < 0.01$). Most modules (31 of 50) had a coherence level above 50% and only 4 of 50 had gene coherence below 30%. The actual coherence levels may be considerably higher, as many genes are not annotated in current databases. Indeed, an in-depth inspection identified many cases in which genes known to be associated with the main process of the module were simply not annotated as such.

We obtained a global view of the modules and their function by compiling all gene annotations and motifs significantly enriched in

each module into a single matrix[5] (**Fig. 4a**). This presentation enables an automatic approach for deriving rich descriptions for modules. For example, the attributes for the respiration module (**Fig. 3**) are immediately apparent in this representation, including the Hap4 and Msn4 (STRE) binding sites and the ion transport, TCA cycle, aerobic respiration, and mitochondrion annotated genes (**Fig. 4a**). The matrix representation also gives further support to the inferred modules. For example, it justifies the division of amino-acid metabolic processes into four modules (**Fig. 4b**): whereas the modules share certain attributes (for example, amino acid metabolism), each is characterized by a unique combination of gene annotations (for example, only module 9 is also annotated as starvation response). Furthermore, all of the modules in this module group are associated with a common *cis*-regulatory motif (Gcn4), but each has a unique signature of *cis*-regulatory motifs.

To obtain a global perspective on the relationships between different modules, and the extent to which they group together, we compiled a graph of modules and *cis*-regulatory motifs, and we connected modules to their significantly enriched motifs (**Fig. 5**). In this view, sets of similar but distinct modules, such as amino acid metabolism modules (8–11), energy modules (1–3, 25, 33, 41) and DNA/RNA modules (13–15, 17, 18) form module groups, such that all modules share at least one motif. Different modules in a group are again characterized by partly overlapping but distinct combinations of motifs. We also searched for pairs of motifs that are significantly enriched (as a pair) in the upstream regions of module genes (**Supplementary Table 1** online). Although different modules were characterized by distinct motif pairs, there was overlap between the motif pairs of modules within a module group (see, for example, **Supplementary Table 1** online), providing further support for the combinatorial nature of the inferred regulation programs. When we examined the predicted regulators of modules (**Fig. 5**), we saw that modules belonging to the same module group appear to share some, but not all, of their regulators. These results suggest a higher level of modularity of the yeast transcriptional network, in which functionally related modules share some of their regulatory elements, yet each module is characterized by a unique regulatory program.

We next evaluated the inferred regulation programs. We compared the known function of the inferred regulators with the method's predictions, where known function is based on a compiled list of literature references (**Supplementary Table 2** online), in which direct experimental evidence exists for the role of the predicted regulators. In most modules (35 of 50), the regulators were predicted to have a role under the expected conditions (**Table 1**). Most modules (30 of 50) also included genes known to be regulated by at least one of the module's predicted regulators (**Table 1**). Many modules (15 of 50) also had an exact match between *cis*-regulatory motifs enriched ($P < 10^{-4}$) in upstream regions of the module's genes and the regulator known to bind to that motif (**Table 1**).

To identify the function of the regulators, we associated each regulator with biological processes, experimental conditions and possibly a binding motif. As a regulator X may regulate more than one module, its targets consist of the union of the genes in all modules predicted to be regulated by X. We tested the targets of each regulator for enrichment of the same motifs and gene annotations as above (**Fig. 4a**). In addition, we tested each regulator for experimental conditions that it significantly regulates by examining how conditions are split by each

Table 1 Summary of module analysis and validation

#	Module[a]	# G[b]	C (%)[c]	Reg.[d]	M	C	G	Reg.[d]	M	C	G	Reg.[d]	M	C	G	Reg.[d]	M	C	G	Reg.[d]	M	C	G	Reg.[d]	M	C	G
1	Respiration and carbon regulation	55	84	Hap4				HMLAlpha2				Cmk1				Gac1				Xbp1				Msn4			
2	Energy, osmolarity and cAMP signaling	64	64	Tpk1				Kin82				Yer184c				Cmk1				Ppt1				Kns1			
3	Energy and osmotic stress I	31	65	Xbp1				Kin82				Tpk1															
4	Energy and osmotic stress II	42	38	Ypl230w				Yap6				Gac1				Wsc4											
5	Glycolysis and folding	37	86	Gcn20				Ecm22				Bmh1				Bas1											
6	Galactose metabolism	4	100	Gal4				Gac1				Hir3				Ime4											
7	Snf kinase regulated processes	74	47	Ypl230w				Yap6				Tos8				Sip2											
8	Nitrogen catabolite repression	29	66	Gat1				Plp2																			
9	Amino acid metabolism I	39	95	Gat1				Ime4				Cdc20				Slt2											
10	Amino acid metabolism II	37	95	Xbp1				Hap4				Afr1				Uga3				Ppt1							
11	Amino acid and purine metabolism	53	92	Gat1				Ppz2				Rim11															
12	Nuclear	47	47	HMLAlpha2				Ino2																			
13	Mixed I	28	50	Pph3				Ras2				Tpk1															
14	Ribosomal and phosphate metabolism	32	81	Ppt1				Sip2				Cad1															
15	mRNA,rRNA and tRNA processing	43	40	Lsg1				Tpk2				Ppt1															
16	RNA processing and cell cycle	59	36	Ypl230w				Ime4				Ppt1				Tpk2				Rho2				Mcm1			
17	DNA and RNA processing	77	43	Tpk1				Gis1				Ppt1															
18	TFs and RNA processing	59	68	Gis1				Pph3				Tpk2				Lsg1											
19	TFs and nuclear transport	48	56	Ypl230w				Met18				Ppt1															
20	TFs I	53	92	Cdc14				Mcm1				Ksp1															
21	TFs II	50	54																								
22	TFs, cell wall and mating	39	59	Ptc3				Sps1																			
23	TFs and sporulation	43	60	Rcs1				Ypl133c																			
24	Sporulation and TFs	74	39	Gcn20				Gat1				Ste5															
25	Sporulation and cAMP pathway	59	37	Xbp1				Ypl230w				Sip2				Not3											
26	Sporulation and cell wall	78	40	Ypl230w				Yap6				Msn4															
27	Cell wall and transport I	23	48	Shp1				Bcy1				Gal80				Ime1				Yak1							
28	Cell wall and transport II	63	46	Ypl230w				Kin82				Msn4															
29	Cell differentiation	41	71	Ypl230w				Ypk1				Cna1															
30	Cell cycle (G2/M)	30	70	Cdc14				Clb1				Far1															
31	Cell cycle, TFs and DNA metabolism	71	85	Gis1				Ste5				Clb5															
32	Cell cycle and general TFs	64	72	Ime4				Ume1				Xbp1				Prr1				Cnb1				Arp9			
33	Mitochondrial and signalling	87	60	Tpk1				Cmk1				Yer184c				Gis1											
34	Mitochondrial and protein fate	37	78	Ypk1				Sds22				Rsc3															
35	Trafficking and mitochondrial	87	56	Tpk1				Sds22				Etr1															
36	ER and nuclear	79	86	Gcn20				Yjl103c				Not3				Tup1											
37	Proteasome and endocytosis	31	71	Ime4				Cup9				Bmh2				Hrt1											
38	Protein modification and trafficking	62	79	Ypl230w				Ptc3				Cdc42															
39	Protein folding	23	87	Bmh1				Bcy1				Ypl230w															
40	Oxidative stress I	15	80	Yap1				Sko1				Far1															
41	Oxidative stress II	15	73	Tos8				Flo8																			
42	Unkown (sub-telomeric)	82	45	Gcn20																							
43	Unknown genes I	36	42																								
44	Unknown genes II	29	14	Apg1				Pcl10																			
45	Unknown genes III	39	5	Xbp1				Kar4																			
46	Mixed II	52	42	Gcn20				Tos8				Sip2															
47	Mixed III	41	63	Gcn20				Ume1				Cnb1															
48	Mixed IV	35	29	Fkh1				Sho1																			
49	Ty ORFs	16	6																								
50	Missing values	64	39																								

Enrichment for motif known to participate in regulation by respective regulator — Partial evidence

Respective regulator known to have a role under the predicted condition — Partial evidence

Respective regulator known to regulate module genes or their implied process — Partial evidence

[a]Each module was assigned a name based on the largest one or two categories of genes in the module (combining gene annotations from SGD and the literature). These concise names are used to facilitate the presentation and may not convey the full content of some of the more heterogeneous modules (see modules and their significant annotations in **Fig. 4**). [b]Number of genes in module. [c]Functional/biological coherence of each module, measured as the percentage of genes in the module covered by significant gene annotations ($P < 0.01$). [d]Regulators predicted to regulate each module, along with three scores for each regulator compiled from the literature (for a list of all literature references used, see **Supplementary Table 2** online). Some modules (21, 43, 49, 50) did not have regulators, as none of the candidate regulators was predictive of the expression profile of their gene members.

Darker boxes indicate biological experiments supporting the prediction; lighter boxes indicate indirect or partial evidence. M, enrichment for a motif known to participate in regulation by the respective regulator in upstream regions of genes in the module; C, experimental evidence for contribution of the respective regulator to the transcriptional response under the predicted conditions; G, direct experimental evidence showing that at least one of the genes in the module, or a process significantly overrepresented in the module genes, is regulated by the respective regulator. TF, transcription factor.

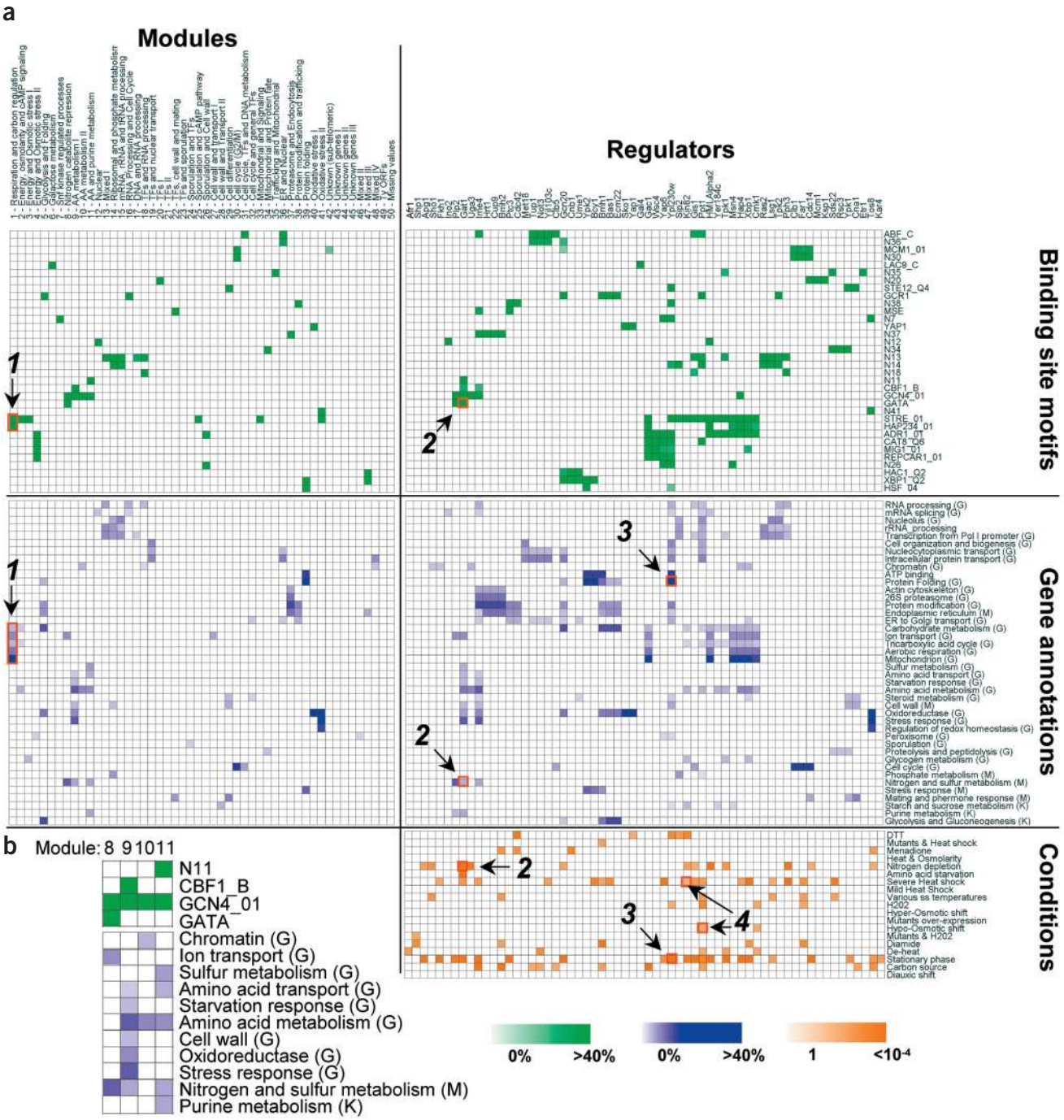
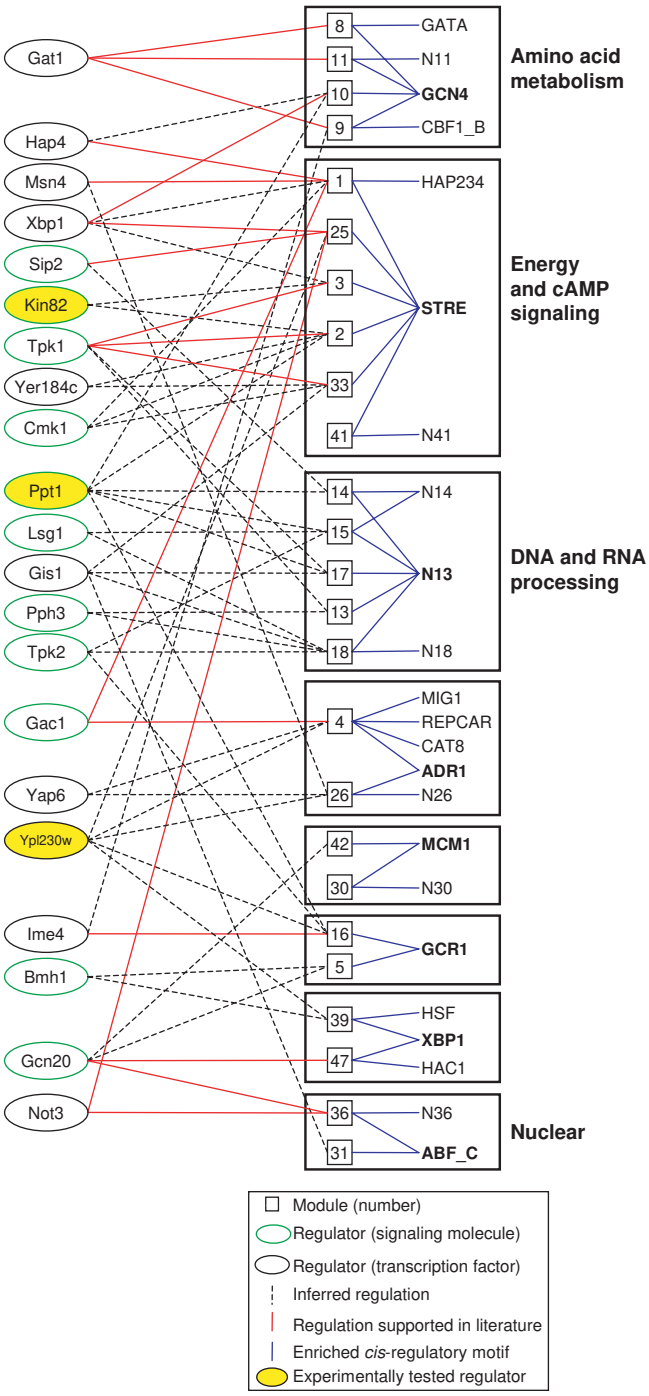


Figure 4 Enrichment of annotations and motif binding sites in modules and in predicted targets of regulators. (**a**) Shown are matrices whose entries represent the percentage of genes associated with annotations in each module (left) or in all predicted targets of each regulator (right). Only significantly enriched annotations ($P < 0.05$, Bonferroni-corrected) are shown. Binding sites (top blocks, green) include both known motifs (TRANSFAC[39]) and *de novo* motifs (identified by a motif-finding program[12] applied to the upstream regions of the genes in each module; *de novo* motifs are named with the character 'N' followed by the module number from which the motif was identified; *de novo* motifs are available in **Supplementary Table 5** online). Gene annotations (middle block, black) were compiled from GO[32] (G), MIPS[33] (M) and KEGG[34] (K). The bottom right matrix shows significant conditions regulated by each regulator, and the entries correspond to *P* values for each condition (mild heat shock, 27 °C to 33 °C; severe heat shock, 17, 21, 25 or 29 °C to 37 °C). A subset of all attributes and regulators was selected for their diversity. A full version of matrices with all attributes and all regulators is available on our website. Entries explicitly referenced in the text are highlighted with red rectangles and labeled with numbered arrows: 1, significant respiration module annotations; 2, selected significant annotations for Gat1 regulator; 3, selected significant annotations for putative regulator Ypl230w; 4, significant conditions for Ppt1 and Kin82 that were experimentally tested. (**b**) Submatrix of significant annotations for amino acid metabolism related modules (8–11). Each module is characterized by a different combination of motifs and gene annotations, supporting the partition into different modules.

relevant regulation tree. For example, in the respiration module (**Fig. 3**), Hap4 upregulation distinguishes stationary-phase conditions from the rest (right branch; Hap4 activates the module) and would thus be associated with regulation in stationary phase. Significant conditions for a particular regulator can thus be identified either by visual inspection of the regulation tree or by an automated statistical procedure. The results of this procedure are summarized in **Figure 4a**. As an example of the resulting associations, the matrix suggests that Gat1 regulates nitrogen and sulfur metabolism processes, binds to the GATA motif and works under conditions of nitrogen depletion (**Fig. 4a**).



When we consider uncharacterized regulators, the predicted regulator annotations provide focused hypotheses about the processes they regulate, the conditions under which they work and the *cis*-regulatory motifs through which their regulation is mediated. For example, we can predict that the putative transcription factor Ypl230w regulates genes important for protein folding during stationary phase (**Fig. 4a**). The ability to generate detailed hypotheses, in the form 'regulator *X* regulates process *Y* under conditions *W*', is among the most powerful features of the module networks procedure, as it also suggests the specific experiments that can validate these hypotheses.

Experimental tests

We selected three hypotheses suggested by the method, involving largely uncharacterized putative regulators, and obtained the relevant yeast deletion strains[22]. To test our ability to predict different types of regulatory mechanisms, we selected a putative zinc-finger transcription factor, Ypl230w, and two putative signaling molecules, the protein kinase Kin82 and the phosphatase Ppt1. Under normal growth conditions, all three deletion strains showed no apparent abnormalities.

As discussed above, each hypothesis generated by the method provides the significant conditions under which the regulator is active, and thereby specifies the experimental conditions under which the mutant should be tested. In concordance with the method's hypotheses (**Fig. 4a**), we tested ΔKin82 under severe heat shock conditions (25 °C to 37 °C), ΔPpt1 during hypo-osmotic shift and ΔYpl230w during the entry to stationary phase.

In each experiment, we used microarray analysis to compare the transcriptional response in the deletion strain to that of the wild-type strain under the same conditions. These genome-wide experiments enable a complete evaluation of the accuracy of our predictions for each regulator: whether it has a regulatory role in the predicted conditions, whether it regulates genes in modules that it was predicted to regulate and most importantly, whether it regulates processes that the method predicted it to regulate.

We used a paired *t*-test ($P < 0.05$) to identify the genes that were differentially expressed between wild-type and mutant strains under the tested conditions. The number of such genes was much higher than expected by chance (1,034 for ΔKin82, 1,334 for ΔPpt1 and 1,014 for ΔYpl230w), showing that all three regulators have a role in the predicted conditions. To focus on the most significant changes, we examined only genes with a significant relative change

Figure 5 Global view and higher order organization of modules. The graph depicts inferred modules (middle; numbered squares), their significantly enriched *cis*-regulatory motifs (right; significant motifs from **Fig. 4a**) and their associated regulators (left; ovals with black border for transcription factors or with green border for signal transduction molecules). Modules are connected to their significantly enriched motifs by solid blue lines. Module groups, consisting of sets of modules that share a common motif, and their associated motifs are enclosed in bold boxes. Only connected components that include two or more modules are shown. Motifs connected to all modules of their component are marked in bold. Modules are also connected to their predicted regulators. Red edges between a regulator and a module are supported in the literature: either the module contains genes that are known targets of the regulator (**Table 1**, G column) or upstream regions of genes in the module are enriched for the *cis*-regulatory motif known to be bound by the regulator (**Table 1**, M column). Regulators that we tested experimentally are marked in yellow. Module groups are defined as sets of modules that share a single significant *cis*-regulatory motif. Module groups whose modules are functionally related are labeled (right). Modules belonging to the same module group seem to share regulators and motifs, with individual modules having different combinations of these regulatory elements.

in expression between the wild-type and mutant expression profiles, a total of 281 genes for ΔKin82, 602 genes for ΔPpt1 and 341 genes for ΔYpl230w (**Fig. 6a** and **Supplementary Table 3** online). Under normal conditions, there are few differences between wild-type and mutant strains (data not shown).

To test whether our method correctly predicted the targets of each regulator, we examined the distribution of the differentially expressed genes among the modules. For each putative regulator X, we calculated a P value for the enrichment of differentially expressed genes in each module and then ranked the modules according to these P values. In all three cases, the highest ranking module was predicted to be regulated by X (**Fig. 6b**). In each case, 25% (ΔPpt1; $P < 9 \times 10^{-3}$), 26% (ΔKin82; $P < 10^{-4}$) and 30% (ΔYpl230w; $P < 10^{-4}$) of the genes in the highest ranking module were differentially expressed.

Finally, we tried to identify the process regulated by each regulator by searching for significantly enriched functional annotations in its set of differentially expressed genes. In two cases (ΔYpl230w and ΔPpt1), the annotations matched those predicted for the regulator (**Fig. 6c**), supporting the method's suggestions for the regulatory roles of the tested regulators: Ypl230w activates protein-folding, cell-wall and ATP-binding genes, and Ppt1 represses phosphate metabolism and rRNA processing.

Altogether, the experimental validations support the functions proposed by our method and show its ability to accurately predict functions for regulators, their targets and the experimental conditions under which this regulation occurs, providing insight into the roles of regulatory genes.

DISCUSSION

Discovering biological organization from gene expression data is a promising but challenging task. The module networks identification method presented here offers unique capabilities in extracting modularity and regulation from expression data.

Although other approaches identify modules of coregulated genes and their shared *cis*-regulatory motifs[6,10], they do not directly suggest the regulators themselves. In contrast, our method identifies both regulatory modules and their control programs, suggesting concrete regulators for each module, their effect and combinatorial interactions and the experimental conditions under which they are active. Our comprehensive evaluation using functional annotations, *cis*-regulatory motifs and the literature validates the coherence of the modules and the consistency of the regulation programs. On a global scale, it also suggests a higher order organization of combinatorial regulation in the yeast transcriptional network, in which distinct modules are characterized by partly overlapping combinations of *cis*-regulatory motifs and regulators.

Perhaps the most powerful feature of our method is its ability to generate detailed testable hypotheses concerning the role of specific regulators and the conditions under which this regulation takes place. We offer experimental results supporting three of our computationally generated hypotheses, suggesting regulatory roles for previously uncharacterized proteins. Our other hypotheses have not yet been tested.

A key question regarding the validity of our approach is explaining how regulatory events can be inferred from gene expression data. To identify a regulatory relation in expression data, both the regulator and its targets must be transcriptionally regulated[18], resulting in detectable changes in their expression. Recent large-scale analyses of the regulatory networks of *Escherichia coli*[23] and *S. cerevisiae*[24,25] found a prevalence of cases in which the regulators are themselves transcriptionally regulated, a process whose functional importance is supported both theoretically and experimentally[26,27]. Such concordant changes in the expression of both the regulators and their targets (for example, **Fig. 7b**,**d**) allow our automated procedure to detect statistical associations between them. Indeed, using recently published genome-wide *cis*-regulatory location data[24], we found that some of the inferred modules and their associated transcription factors are part of such regulatory structures (**Fig. 7a**,**f**). The location data allowed only a limited comparison, as it was obtained under normal growth conditions.

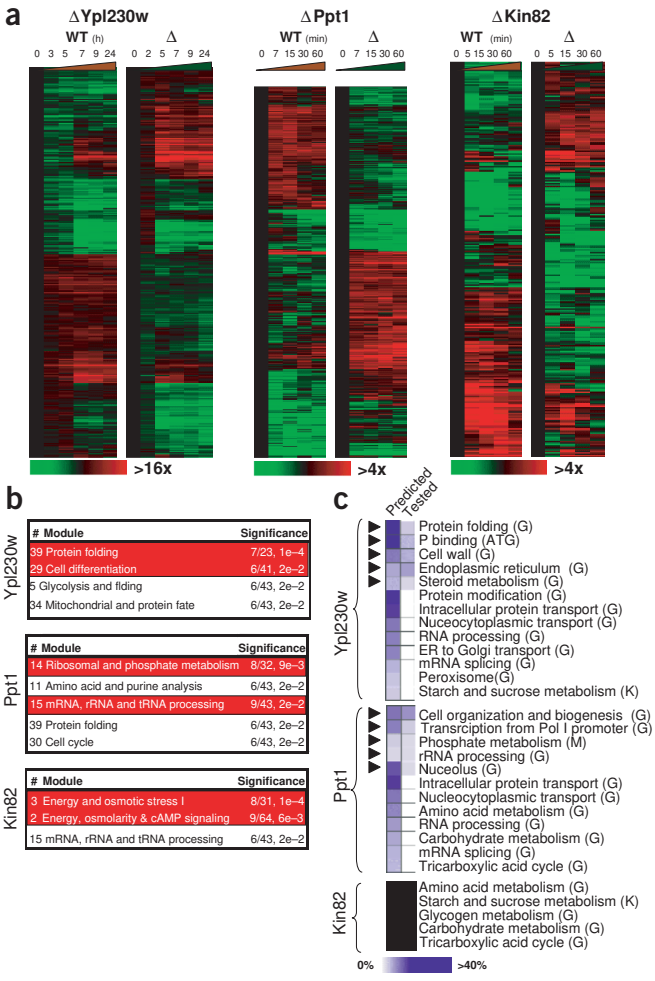


Figure 6 Microarray experiments testing functional predictions for putative regulators. (**a**) Expression data for the differentially expressed genes (extracted using paired *t*-test) for both the wild-type (WT) and mutant (Δ) time series in the following experiments: ΔYpl230w during stationary phase, ΔPpt1 during hypo-osmotic shift and ΔKin82 under heat shock. (**b**) Ranked modules table for each tested regulator X; ranking is based on P value calculated for enrichment of differentially expressed genes in each module. All modules significantly enriched for these genes ($P < 0.05$) are shown along with the number of differentially expressed genes out of the total number of genes in the module and the corresponding P value for the enrichment. Modules predicted to be regulated by the respective regulator X are highlighted in red. (**c**) Functional predictions for tested regulators. The left column (Predicted) for each regulator shows all annotations predicted by the method to be associated with that regulator (extracted from the corresponding column in **Fig. 4a**). The right column (Tested) shows which annotations were also significantly enriched in the set of differentially expressed genes of each regulator ($P < 0.05$; black triangles), where the intensity of each entry represents the fraction of genes with the annotation from the set of differentially expressed genes.

Our method is also able to identify correct regulatory roles for signal transduction molecules from expression data. We attribute this ability to the presence of positive and negative feedback loops[28] in which a signaling molecule regulates a transcription factor that, in turn, regulates the activity of the gene encoding the signaling molecule (**Fig. 7c**,**d**). We found evidence for the presence of such feedback loops in the predicted regulation programs for some modules (for example, **Fig. 7c**). Negative feedback loops also explain why Tpk1 is inferred as an activator rather than a repressor (**Fig. 7e**).

Overall, our results show that regulatory events, including post-transcriptional ones, have a detectable signature in the expression of genes encoding transcription factors and signal transduction molecules. Notably, our computational method will probably succeed in organisms other than yeast. For example, in *E. coli*, approximately 40% of transcription factors are autoregulated[26,27], potentially increasing the power of our approach. In addition, our method's ability to detect combinatorial regulation is crucial for its application to higher eukaryotes.

Despite the successes described above, our method fails to identify certain regulatory relations and may occasionally predict regulatory relationships that do not hold. In some cases, the regulator's expression does not change sufficiently to be detected. This behavior can occur when the regulator's activity is attributed mostly to post-transcriptional changes or when the regulatory event occurs primarily under specific conditions that are not included in our data set. We expect the latter relations to be identified by our method if given a more appropriate data set.

Our method may also fail to identify certain correct regulatory relations despite a detectable change in the regulator's expression pattern. If several regulators participate in the same regulatory event, our method typically identifies only one representative of this group, as the effect of the remaining regulators is generally indistinguishable based on expression data alone. In other cases, a gene belonging to the set of candidate regulators may be highly predictive of the module, often because it is a member of the module but occasionally simply by chance. Such a gene may be mistakenly selected by the procedure as the module's regulator, leading to the prediction of an incorrect regulatory relationship. In such cases, the true regulator may become redundant and is often assigned as a module member together with its targets. Finally, some regulatory relations may be specific to a regulator and its target and cannot be generalized to an entire module. As our method is designed to identify shared regulatory responses across a module, it will not detect such focused regulatory relations.

Our computational framework also has several important limitations. For example, we want to automatically select the appropriate number of modules for a given data set also, unlike other approaches (for example, that described in ref. 6), our modules are currently non-overlapping, so that a gene may belong only to a single module. It is possible to extend our framework to address these issues.

Overall, our method provides a clear global view of functional modules and their regulation and suggests concrete hypotheses concerning the role of specific regulators. As more diverse gene expression data sets become available, it is our belief that applying the

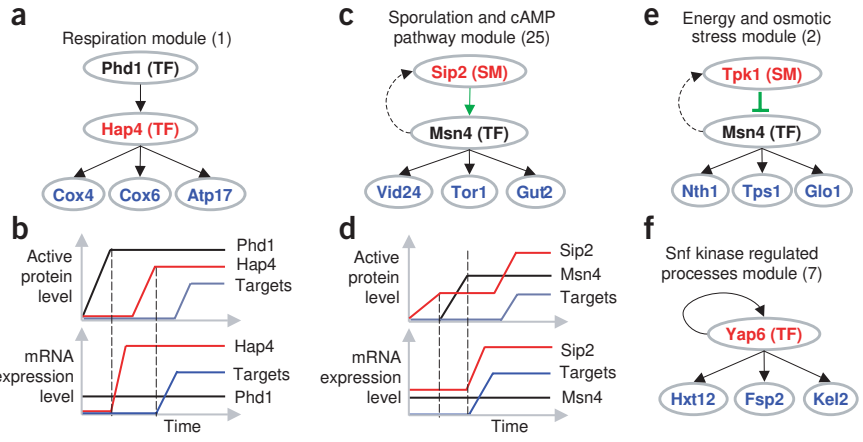


Figure 7 Regulatory components allowing inference of regulation from expression data. We used genome-wide location data[24] and the literature to show how the method identifies regulators from expression data. We present different types of regulatory components and an inferred module in which such a component is found. For each regulatory component, the relevant transcription factors (TFs), signal transduction molecules (SMs), target genes (blue) and their relations (directed arrows) are shown. *Cis*-regulation events using transcription factors are shown in black arrows, post-transcriptional events using signaling molecules in green arrows. The gene in red is that predicted as a regulator of the module. The targets in blue are those genes that are both predicted by our method to be regulated by the red gene and also bound by the transcription factor according to the location data ($P < 0.001$; ref. 24). Black regulators are not expected to be inferred from gene expression data, because, in general, their mRNA levels do not change. (**a**) Regulator chain. Transcription factor (Phd1) activates its targets that include an additional secondary transcription factor (Hap4). The secondary transcription factor activates the secondary targets (Cox4, Cox6, Atp17, Cox7, Cox8, Qcr2, Mir1, Qcr7, Cox12, Qcr9, Cox13, Cyt1, Atp1, Atp2, Atp3, Rip1, Atp5, Atp7, Atp20, Cor1 and Ylr294c). Only the secondary transcription factor is inferred as the module's regulator. (**b**) Dynamic behavior of regulator chain. Transcription factor (Phd1) activity (top graph, black) rises with no change in its mRNA level (bottom graph, black). Transcription factor induces expression of secondary transcription factor (Hap4; bottom graph, red), resulting in a greater amount of active secondary transcription factor (top graph, red). Finally, secondary transcription factor activation leads to a rise in the expression of its targets (bottom graph, blue). (**c**) Positive signaling loop. Signaling molecule (Sip2) activates transcription factor (Msn4; ref. 42). Transcription factor induces transcription of various targets (Vid24, Tor1 and Gut2), possibly including the signaling molecule. The coordinated expression changes in signaling molecule and targets allow signaling molecule (but not transcription factor) to be correctly inferred as a regulator. (**d**) Dynamic behavior of signaling multi-component loop. Signaling molecule (Sip2) protein (top graph, red) induces transcription factor (Msn4) activity (top graph, black). Transcription factor induces expression of signaling molecule (bottom graph, red) and target genes (bottom graph, blue). Transcription factor expression level is unchanged. (**e**) Negative signaling loop. Signaling molecule (Tpk1) inhibits activity of transcription factor (Msn4; ref. 43). Transcription factor (Msn4) induces transcription of the module's genes (for example, Nth1 (ref. 44), Tps1 (ref. 45) and Glo1 (ref. 46)) and possibly of signaling molecule (supported by presence of STRE, an Msn4 bound motif, in Tpk1's upstream region; Tpk1 is also part of the cAMP/PKA pathway and other components of the pathway were previously shown to be induced by Msn4; ref. 3). Signaling molecule changes concordantly with targets and is thus correctly inferred by the method as the module's regulator. But, because both signaling molecule and the targets are upregulated, the method predicts that signaling molecule activates the module, in contrast to its actual inhibitory role. (**f**) Single input module (auto-regulation). Transcription factor (Yap6) activates its own transcription and that of its target genes (Hxt12, Hxt15, Hxt16, Yil122w, Fsp2, Yol157c, Yil172c and Kel2).

module networks method to such data sets may result in important new insights in the ongoing endeavor to understand the complex web of biological regulation.

METHODS

Candidate regulators. We compiled a set of 466 candidate regulators whose annotations in the *Saccharomyces* Genome Database (SGD; ref. 29) or Yeast Proteome Database (YPD; ref. 30) suggest a potential regulatory role in the broad sense: both transcription factors and signaling proteins that may have transcriptional impact[18]. We also included genes described to be similar to such regulators. We excluded global regulators, whose regulation is not specific to a small set of genes or processes. A full candidate regulators list is available in **Supplementary Table 4** online. We binned the gene expression of the candidate regulators into three categories: downregulated, no change and upregulated, using k-means clustering[15,31].

DNA microarray data set. We used an *S. cerevisiae* gene expression data set consisting of 173 microarrays that measure responses to various stress conditions[3]. We downloaded these expression data, in log (base 2) ratio to control format, from the Stanford Microarray Database. We chose a subset of 2,355 genes that have a significant change in gene expression under the measured stress conditions, excluding members of the generic environmental stress response cluster[3]. Our gene set also included all genes chosen as candidate regulators; a full gene list can be found at our website.

Database annotations. We downloaded the Gene Ontology (GO; ref. 32) hierarchy associations from SGD[29] on 23 October 2002 (version 1.311), the Munich Information Center for Protein Sequence (MIPS; ref. 33) functional categories on 15 August 2002 and the Kyoto Encyclopedia of Genes and Genomes (KEGG; ref. 34) metabolic pathways on 2 July 2002.

Regulation programs. A regulation program specifies a set of contexts and the response of the module in each context. A context is a rule describing the qualitative behavior (upregulation, no change or downregulation) of a small set of genes that control the expression of the genes in the module. This set of rules is organized as a regression tree in which each path to a leaf in the tree defines a context using the tests on the path. A regression tree is composed of two basic building blocks: decision nodes and leaf nodes. Each decision node corresponds to one of the regulatory inputs (regulator expression values) and a query on its value (for example, "is Hap4 upregulated?"). Each decision node has two child nodes: the right child node is chosen when the answer to the corresponding query is true; the left node is chosen when not. For a given array, one begins at the root node and continues down the tree in a path according to the answers to the queries in that particular array. Thus, the context specifies a set of arrays: those in which the path down the tree reaches the corresponding leaf. The response in each context is modeled as a normal distribution over the expression values of the module's genes in these arrays; this distribution is encoded using a mean and variance stored at the corresponding leaf. The model semantics is that, given a gene g in the module and an array a in a context, the probability of observing some expression value for gene g in array a is governed by the normal distribution specified for that context. For each array, all genes in the same module follow the same normal distribution. For a context in which the genes are tightly coregulated, the distribution will have a small variance. For a context in which the genes are not tightly regulated, the distribution may have a large variance. Thus, a regression tree allows for expression profiles with different degrees of conservation of the mean behavior of the module.

Learning module networks. Our procedure is iterative; in each iteration, the procedure searches for a regulation program for each module and then reassigns each gene to the module whose program best predicts its behavior. These two steps are iterated until convergence in reached. The approach is model-based and integrates the ideas and rationale previously described[15,35]: we define a space of possible models and use a statistically based likelihood score called the Bayesian score[36] to evaluate a model's fit to the data. Our iterative learning procedure attempts to search for the model with the highest score by using the Expectation Maximization (EM) algorithm[37,38]. An important property of the EM algorithm is that each iteration is guaranteed to improve the likelihood of the model until convergence to a local maximum of the score is achieved. For clarity, we present a simplified version that captures the algorithm's essence (see the technical report at the website for exact details). Each iteration of the algorithm consists of two steps: an E-step and an M-step. In the M-step, the procedure is given a partition of the genes into modules and learns the best regulation program (regression tree) for each module. For computational efficiency, some M-steps optimize only the parameters of the normal distributions at the leaves of the regulation tree and leave the tree structure unchanged. An M-step that re-learns the regulation tree structure is used only after iterations of E-steps and parameter-optimizing M-steps converge. The regulation program is learned through a combinatorial search over the space of trees. The tree is grown from the root to its leaves. At any given node, the query that best partitions the gene expression into two distinct distributions is chosen until no such split exists. In the E-step, given the inferred regulation programs, we determine the module whose associated regulation program best predicts each gene's behavior. Each regulation program defines a probability distribution over the gene's expression levels in each array. We can test the probability of a gene's measured expression values in the data set under each regulatory program as follows. We evaluate, for each array, the probability of the associated expression measurement in the array's context, as specified by the regression tree. We then multiply the probabilities for the different arrays, obtaining an overall probability that this gene's expression profile was generated by this regulation program. We then select the module whose program gives the gene's expression profile the highest probability and re-assign the gene to this module. We take care not to assign a regulator gene to a module in which it is also a regulatory input, as it is not surprising that a gene can predict its own gene expression. Overall, each iteration of this procedure requires computation time that is linear in the size of the expression matrix (number of genes multiplied by number of experiments).

We initialized our module network procedure with 50 clusters (see our website for rationale on the choice of number of clusters) by using PCluster, a hierarchical agglomerative clustering (see technical report in our website) and creating one module from each of the resulting clusters. We then applied the EM algorithm to this starting point, refining both the gene partition and the regulatory programs. Our procedure converged after 23 iterations (four tree-structure-change iterations) to the 50 modules we analyzed, changing the initial module assignment of 49% of the genes (**Supplementary Figs. 4** and **5** online). We note that EM converges only to a local maximum and is sensitive to its initial starting point. We provide an extensive evaluation of the quality and sensitivity of our procedure on our website, showing that it leads to a high-quality local maximum.

Evaluating statistical significance of modules. All of the statistical evaluations were done and visualized in GeneXPress, a cluster analysis and visualization tool we developed for this purpose. The tool can evaluate the output of any clustering program for enrichment of gene annotations and motifs and is freely available for academic use.

Annotation enrichment. To analyze the biological relevance of a module, we associated each gene with the processes in which it participates. We removed all annotations associated with less than five genes from our gene set. This resulted in a list of 923 GO[32] categories, 208 MIPS[33] categories and 87 KEGG[34] pathways. For each module and for each annotation, we calculated the fraction of genes in the module associated with that annotation and used the hypergeometric distribution to calculate a P value for this fraction. We carried out a Bonferroni correction for multiple independent hypotheses and took values of $P < 0.05/n$ ($n = 923$, 208 and 87 for GO, MIPS and KEGG annotations, respectively) to be significant for **Table 1** and **Figure 4**.

Promoter analysis. We searched for motifs (represented as position-specific scoring matrices) within 500 bp upstream of each gene (sequences were retrieved from SGD[29] on 2 July 2002). We downloaded version 6.2 of TRANSFAC[39], containing 34 known fungi *cis*-regulatory motifs. We also used a discriminative motif finder[12] to search for novel motifs that differentiate each module from the other genes in the data set and identified 50 potentially novel motifs. We used the *S. cerevisiae* GC content as a background distribution, over which we computed the distribution of motif scores. We selected a threshold so that only 5% of the random sequences pass this threshold and considered the binding site to be present in the upstream region if it was scored above this threshold.

Motif combinations. We searched for statistically significant occurrences of motif pairs. We constructed a motif pair attribute, which assigns a 'true' value for each gene if and only if both motifs of the pair are found in the upstream region of that gene. For each module and for each motif pair attribute, we calculated the fraction of genes in the module associated with that attribute and used the hypergeometric distribution to calculate a *P* value for this fraction as above. We took values of $P < 0.05$ (Bonferroni-corrected) to be significant.

Regulator annotation. We associated regulators with annotations and binding sites in the same way that we associated these attributes with modules. Because a regulator may regulate more than one module, its targets consist of the union of the genes in all modules predicted to be regulated by that regulator. We tested the targets of each regulator for enrichment of the same motifs and gene annotations as above using the hypergeometric *P* value. We took values of $P < 0.05$ (Bonferroni-corrected as for module annotations) to be significant.

Regulator experimental conditions. We associated regulators with experimental conditions for which they are significantly predictive according to the inferred modules. Experimental conditions were extracted from the array labels[3]. For each occurrence of a regulator as a decision node in a regression tree, we computed the partition of each experimental condition between the right branch (the 'true' answer to the query on the regulator) and the left branch (the 'false' answer) and used the binomial distribution to compute a *P* value on this partition. We took values of $P < 0.05$ to be significant.

Strains and growth conditions. We used the following strains (from Invitrogen) in this study: DBY8778(BY4741) (genotype *MATa ura3Δ leu2Δ hisΔ1 met15Δ1*); DBY10058 (DBY8778 ypl230w:Kan^R); DBY10089 (DBY8778 kin82:Kan^R); DBY10090 (DBY8778 ppt1:Kan^R). Unless otherwise mentioned, cells were grown with shaking (295 r.p.m.) in rich medium (yeast extract/peptone/dextrose; YPD) at 30 °C (normal conditions).

Apart from the stationary phase experiment, we grew cultures to early log phase (A_{600} = 0.2–0.4). In all experiments, we used mutant and wild-type strains carrying the same genetic background and subjected them to the same procedures. For all experiments, we drew out aliquots of 25–35 ml for extracting total RNA, vacuum-collected cells onto a 45 μm filter (Osmonics), snap-froze them in liquid nitrogen and kept them at –80 °C until use.

Heat shock. We grew cells at 25 °C, collected them by centrifugation, resuspended them in an equal volume of 37 °C medium and returned them to 37 °C for growth. We collected samples 0, 5, 15, 30 and 60 min after transfer to 37 °C.

Stationary phase. We grew cultures to A_{600} of 0.27 (Ypl230w mutants) and 0.4 (congenic wild-type; DBY8778) and collected samples (0 h). We also collected samples at 2 (or 3), 5, 7, 9 and 24 h.

Hypo-osmotic shock. We grew cultures with 1 M sorbitol for ~20 h, collected cells by centrifugation and resuspended them in YPD. We collected samples 0, 7, 15, 30 and 60 min after transfer to YPD.

RNA preparation and hybridization. We isolated total RNA using the hot acid phenol method followed by ethanol precipitation[2]. We extracted poly(A)$^+$ mRNA, used for all cDNA microarray analyses, from total RNA using the Oligotex midi kit (Qiagen). We used 1–2 μg for each labeling reaction. We labeled cDNA probes using a 3′ anchored oligo-dT primer, essentially as described[2]. We used experimental samples to generate Cy5-labeled cDNA probes and used mRNA reference pools extracted from cultures of the respective strains grown to early log phase under normal conditions to generate Cy3-labeled cDNA probes. We hybridized Cy5- and Cy3-labeled probes together to microarrays printed with PCR-amplified fragments[1] representing 6,280 of the *S. cerevisiae* open reading frames.

Data acquisition and analysis. We acquired and analyzed images using the GenePix 4000 microarray scanner (Axon instruments) and GenePix Pro 3.0, respectively. Data were subjected to quality-control filters, normalized and stored in the Stanford Microarray Database.

Raw data files for each array containing all measured values and flags are also available on our website. In subsequent analyses, we used only those spots representing successfully amplified genes, with fluorescent intensity in both channels that was 1.2 times greater than the local background. We selected for analysis only those open reading frames for which more than 80% of their spots followed the above rule. We used log-transformed (base 2) ratios for subsequent analysis.

Verifying chromosomal integrity of deletion strains. For each deletion strain, we compared the reference RNA pool to that of the congenic wild-type strain to look for gross chromosomal rearrangements reflected in locus-associated gene expression biases[40]. None of the strains used in this study showed such a bias (data not shown).

Identifying differentially expressed genes in microarray experiments for putative regulators. We used a paired two-tailed *t*-test to compare the expression time series of mutant and wild-type strains. Genes with $P < 0.05$ from the *t*-test were considered differentially expressed. Each time series was zero-transformed to enable comparison of the response to the tested condition. For Kin82 and Ppt1, we gave all time points as input to the *t*-test (5, 15, 30 and 60 min for Kin82; 7, 15, 30 and 60 min for Ppt1). For the Ypl230w experiment, measuring response during stationary phase, we used only the late time points (7, 9 and 24 h) as the response to this growth condition starts at this time. To ensure that only genes with large differences are included, we also required expression to differ by a factor of at least 2 in at least half the time points compared. The only exception was Ppt1, in which we required expression to differ by a factor of at least 1.3, because the overall signal in these arrays was weaker.

URLs. More details of our results, together with the full raw expression data obtained from the three different microarray experiments, can be found on our website (http://dags.stanford.edu/module_nets/). GeneXPress is freely available for academic use at http://GeneXPress.stanford.edu/. The main yeast stress data set that we analyzed in the paper can be downloaded from http://genome-www.stanford.edu/yeast_stress/. A technical report describing the probabilistic clustering used to initially partition the genes to modules is available at http://dags.stanford.edu/module_nets/tech.html. A technical report describing the full details of the module networks method is available at http://dags.stanford.edu/module_nets/tech.html. All experimental data is available at the Stanford Microarray Database: http://genome-www5.stanford.edu/MicroArray/SMD/.

Accession numbers. Expression data for the three different microarray experiments is available at Gene Expression Omnibus[41] (accession numbers: GSE406 for Kin82, GSE407 for Ppt1, GSE408 for Ypl230w).

Note: Supplementary information is available on the Nature Genetics website.

ACKNOWLEDGMENTS

We thank L. Garwin, M. Scott, G. Simchen and L. Stryer for their useful comments on earlier versions of this manuscript and A. Kaushal, T. Pham, A. Tanay and R. Yelensky for technical help with software and visualization. E.S., D.K. and N.F. were supported by a National Science Foundation grant under the Information Technology Research program. E.S. was also supported by a Stanford Graduate Fellowship. M.S was supported by the Stanford University School of Medicine Dean's Fellowship. A.R. was supported by the Colton Foundation. D.P. was supported by an Eshkol Fellowship. N.F. was also supported by an Alon Fellowship, by the Harry & Abe Sherman Senior Lectureship in Computer Science and by the Israeli Ministry of Science.

COMPETING INTERESTS STATEMENT

The authors declare that they have no competing financial interests.

Received 28 January; accepted 17 April 2003
Published online 11 May 2003; doi:10.1038/ng1165

1. DeRisi, J.L., Iyer, V.R. & Brown, P.O. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* **278**, 680–686 (1997).
2. Spellman, P.T. *et al.* Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell* **9**, 3273–3297 (1998).
3. Gasch, A.P. *et al.* Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell* **11**, 4241–4257 (2000).

4. Eisen, M.B., Spellman, P.T., Brown, P.O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* **95**, 14863–14868 (1998).
5. Wu, L.F. *et al.* Large-scale prediction of *Saccharomyces cerevisiae* gene function using overlapping transcriptional clusters. *Nat. Genet.* **31**, 255–265 (2002).
6. Ihmels, J. *et al.* Revealing modular organization in the yeast transcriptional network. *Nat. Genet.* **31**, 370–377 (2002).
7. Halfon, M.S., Grad, Y., Church, G.M. & Michelson, A.M. Computation-based discovery of related transcriptional regulatory modules and motifs using an experimentally validated combinatorial model. *Genome Res.* **12**, 1019–1028 (2002).
8. Tanay, A., Sharan, R. & Shamir, R. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **18** Suppl 1, S136–S144 (2002).
9. Roth, F.P., Hughes, J.D., Estep, P.W. & Church, G.M. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.* **16**, 939–945 (1998).
10. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J. & Church, G.M. Systematic determination of genetic network architecture. *Nat. Genet.* **22**, 281–285 (1999).
11. Pilpel, Y., Sudarsanam, P. & Church, G.M. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat. Genet.* **29**, 153–159 (2001).
12. Segal, E., Barash Y., Simon I., Friedman N. & Koller D. From Promoter Sequence to Expression: A Probabilistic Framework. in *Proceedings of the 6th International Conference on Research in Computational Molecular Biology (RECOMB)* 263–272 (Washington, DC, 2002).
13. Pearl, J. *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufmann, Palo Alto, 1988).
14. Dhaseleer, P., Liang, S. & Somogoyi, R. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* **16**, 707–726 (2000).
15. Pe'er, D., Regev, A., Elidan, G. & Friedman, N. Inferring subnetworks from perturbed expression profiles. *Bioinformatics* **17** Suppl 1, S215–S224 (2001).
16. Hartemink, A.J., Gifford, D.K., Jaakkola, T.S. & Young, R.A. Combining Location and Expression Data for Principled Discovery of Genetic Regulatory Networks. in *Pacific Symposium on Biocomputing* (Kauai, 2002).
17. Tanay, A. & Shamir, R. Computational expansion of genetic networks. *Bioinformatics* **17** Suppl 1, S270–S278 (2001).
18. Pe'er, D., Regev, A. & Tanay, A. Minreg: inferring an active regulator set. *Bioinformatics* **18** Suppl 1, S258–S267 (2002).
19. Forsburg, S.L. & Guarente, L. Identification and characterization of HAP4: a third component of the CCAAT-bound HAP2/HAP3 heteromer. *Genes Dev.* **3**, 1166–1178 (1989).
20. Norbeck, J. & Blomberg, A. The level of cAMP-dependent protein kinase A activity strongly affects osmotolerance and osmo-instigated gene expression changes in *Saccharomyces cerevisiae*. *Yeast* **16**, 121–137 (2000).
21. Lenssen, E., Oberholzer, U., Labarre, J., De Virgilio, C. & Collart, M.A. *Saccharomyces cerevisiae* Ccr4-not complex contributes to the control of Msn2p-dependent transcription by the Ras/cAMP pathway. *Mol. Microbiol.* **43**, 1023–1037 (2002).
22. Winzeler, E.A. *et al.* Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis. *Science* **285**, 901–906 (1999).
23. Shen-Orr, S.S., Milo, R., Mangan, S. & Alon, U. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.* **31**, 64–68 (2002).
24. Lee, T.I. *et al.* Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* **298**, 799–804 (2002).
25. Milo, R. *et al.* Network motifs: simple building blocks of complex networks. *Science* **298**, 824–827 (2002).
26. Hlavacek, W.S. & Savageau, M.A. Rules for coupled expression of regulator and effector genes in inducible circuits. *J. Mol. Biol.* **255**, 121–139 (1996).
27. Rosenfeld, N., Elowitz, M.B. & Alon, U. Negative autoregulation speeds the response times of transcription networks. *J. Mol. Biol.* **323**, 785–793 (2002).
28. Roberts, C.J. *et al.* Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles. *Science* **287**, 873–880 (2000).
29. Cherry, J.M. *et al.* SGD: Saccharomyces Genome Database. *Nucleic Acids Res.* **26**, 73–79 (1998).
30. Hodges, P.E., McKee, A.H., Davis, B.P., Payne, W.E. & Garrels, J.I. The Yeast Proteome Database (YPD): a model for the organization and presentation of genome-wide functional data. *Nucleic Acids Res.* **27**, 69–73 (1999).
31. Duda, R.O. & Hart, P.E. *Pattern classification and scene analysis* (John Wiley & Sons, New York, 1973).
32. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
33. Mewes, H.W., Albermann, K., Heumann, K., Liebl, S. & Pfeiffer, F. MIPS: a database for protein sequences, homology data and yeast genome information. *Nucleic Acids Res.* **25**, 28–30 (1997).
34. Kanehisa, M., Goto, S., Kawashima, S. & Nakaya, A. The KEGG databases at GenomeNet. *Nucleic Acids Res.* **30**, 42–46 (2002).
35. Segal, E., Taskar, B., Gasch, A., Friedman, N. & Koller, D. Rich probabilistic models for gene expression. *Bioinformatics* **17** Suppl 1, S243–S252 (2001).
36. Heckerman, D. A tutorial on learning with Bayesian networks. in *Learning in Graphical Models* (ed. Jordan, M.I.) 301–354 (MIT Press, Cambridge, Massachusetts 1998).
37. Dempster, A.P., Laird, N.M. & Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B* **39**, 1–39 (1977).
38. Friedman, N. The Bayesian structural EM algorithm. in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI) 129–138 (1998).
39. Wingender, E. *et al.* The TRANSFAC system on gene expression regulation. *Nucleic Acids Res.* **29**, 281–283 (2001).
40. Hughes, T.R. *et al.* Functional discovery via a compendium of expression profiles. *Cell* **102**, 109–126 (2000).
41. Edgar, R., Domrachev, M. & Lash, A.E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**, 207–210 (2002).
42. Mayordomo, I., Estruch, F. & Sanz, P. Convergence of the target of rapamycin and the Snf1 protein kinase pathways in the regulation of the subcellular localization of Msn2, a transcriptional activator of STRE (Stress Response Element)-regulated genes. *J. Biol. Chem.* **277**, 35650–35656 (2002).
43. Gorner, W. *et al.* Nuclear localization of the C2H2 zinc finger protein Msn2p is regulated by stress and protein kinase A activity. *Genes Dev.* **12**, 586–597 (1998).
44. Zahringer, H., Thevelein, J.M. & Nwaka, S. Induction of neutral trehalase Nth1 by heat and osmotic stress is controlled by STRE elements and Msn2/Msn4 transcription factors: variations of PKA effect during stress and growth. *Mol. Microbiol.* **35**, 397–406 (2000).
45. Boy-Marcotte, E., Perrot, M., Bussereau, F., Boucherie, H. & Jacquet, M. Msn2p and Msn4p control a large number of genes induced at the diauxic transition which are repressed by cyclic AMP in *Saccharomyces cerevisiae*. *J. Bacteriol.* **180**, 1044–1052 (1998).
46. Inoue, Y., Tsujimoto, Y. & Kimura, A. Expression of the glyoxalase I gene of *Saccharomyces cerevisiae* is regulated by high osmolarity glycerol mitogen-activated protein kinase pathway in osmotic stress response. *J. Biol. Chem.* **273**, 2977–2983 (1998).