Revealing modular organization in the yeast transcriptional network

Jan Ihmels, Gilgi Friedlander, Sven Bergmann, Ofer Sarig, Yaniv Ziv & Naama Barkai

Published online: 22 July 2002, doi:10.1038/ng941

Standard clustering methods can classify genes successfully when applied to relatively small data sets, but have limited use in the analysis of large-scale expression data, mainly owing to their assignment of a gene to a single cluster. Here we propose an alternative method for the global analysis of genome-wide expression data. Our approach assigns genes to context-dependent and potentially overlapping 'transcription modules', thus overcoming the main limitations of traditional clustering methods. We use our method to elucidate regulatory properties of cellular pathways and to characterize *cis*-regulatory elements. By applying our algorithm systematically to all of the available expression data on *Saccharomyces cerevisiae*, we identify a comprehensive set of overlapping transcriptional modules. Our results provide functional predictions for numerous genes, identify relations between modules and present a global view on the transcriptional network.

Introduction

With the advent of the DNA microarray technology, it is now possible to study the transcriptional response of a complete genome to different experimental conditions. Gene classification is an essential task in studying the global structure of the transcriptional network. But although standard clustering methods classify genes successfully when applied to relatively small data sets, their use in the analysis of large-scale expression data is limited by two well-recognized drawbacks[1–3]. First, commonly used algorithms assign each gene to a single cluster, whereas in fact genes may participate in several functions and should thus be included in several clusters[4–6]. Second, these algorithms classify genes on the basis of their expression under all experimental conditions, whereas cellular processes are generally affected only by a small subset of these conditions. In the analysis of a particular cellular process, therefore, most conditions do not contribute information but instead increase the amount of background noise.

To study transcriptional regulation, both the co-regulated genes and the experimental conditions that trigger this co-regulation must be identified. We refer to such a combined group of genes and conditions as a 'transcription module'. The naive approach of searching for such modules by considering all possible subsets of genes and conditions is computationally infeasible even for a moderately sized data set. Therefore, more sophisticated methods[2,3] are required.

We have devised a method for identifying transcription modules. Our approach relies on the observation that a set of randomly selected genes is unlikely to be identical to the genes of any transcription module, because the number of such modules is limited. Yet many such sets do have some overlap (that is, a fraction of common genes) with a specific transcription module. In particular, sets of genes that are compiled according to existing knowledge of their functional or (regulatory) sequence similarity may have a significant overlap with a transcription module. Our method is based on an algorithm that receives a gene set that partially overlaps a transcription module and then provides the complete module as output. We refer to this algorithm as the 'signature algorithm'.

Here we present the details of the signature algorithm and establish a reliability measure for its output. We have applied our method to a large data set of over 1,000 expression profiles in the yeast *S. cerevisiae* to characterize the genes and experimental conditions associated with cellular pathways, to identify *cis*-regulatory elements and to carry out a global analysis of the yeast transcriptional network. We used computer-generated expression data to assess the classification capabilities of our method in a controlled setting. In addition, our analysis generated functional links for almost 1,000 genes of unknown function. We verified experimentally the computational predictions for two of these genes, confirming their involvement in the processing of precursor rRNA.

Results

The signature algorithm

The algorithm receives a set of genes as input and proceeds in two stages (Fig. 1*a*). In the first stage, we identify the experimental conditions under which the input genes are co-regulated most tightly. To this end, we calculate the average change in the expression of the input genes for each condition. We refer to these average values as the 'condition scores'. Only conditions with a large (absolute) score are selected. In the second stage, the algorithm selects from the whole genome those genes that show a significant and consistent change in expression under the conditions selected in the first stage. For each gene, we calculate the weighted average change in expression over these conditions, using the condition scores as weights. We refer to these average values as the 'gene scores'. Only genes with a large score are selected (Methods).

Departments of Molecular Genetics and Physics of Complex Systems, Weizmann Institute of Science, Rehovot, 76100, Israel. Correspondence should be addressed to N.B. (e-mail: naama.barkai@weizmann.ac.il).

To evaluate the performance of the signature algorithm, we carried out the following numerical experiment. First, we applied the algorithm to a set of N_{core} genes that were known to be co-regulated. Second, we applied the algorithm to a set that included both those co-regulated genes and N_{rand} randomly selected genes. The addition of many random genes leaves the output of the signature algorithm essentially unchanged (Fig. 2). For example, up to 1,000 random genes can be added to 73 co-regulated ribosomal genes without significantly altering the output of the signature algorithm. In general, the proper identification of the transcription module is achieved as long as N_{rand} is below a critical number N_{rand}^{crit}. This number is proportional to the square of the number of co-regulated genes (that is, $N_{rand}^{crit} \sim N_{core}^2$). An analytical explanation and further numerical confirmation of this scaling law are given, respectively, in Web Note A and Web Fig. A online.

Recurrence as a measure of reliability

A general issue for any classification algorithm is how to evaluate the reliability of its output. For example, applying the signature algorithm to an input set that does not contain a subset of co-regulated genes usually yields an output set, even though this output does not correspond to any transcription module. To distinguish a transcription module from meaningless output, we exploit the ability of the algorithm to filter out non-relevant genes. Specifically, for a given set of genes, we create a new set that contains both genes of this set and genes randomly selected from the whole genome. The signature algorithm is then applied to both the original set and the one derived from it. If the original set includes a subset of co-regulated genes, then the set derived from it also contains those genes. Consequently, the two outputs essentially coincide and are likely to represent a transcription module. By contrast, very different outputs are obtained when the original set is composed of randomly chosen genes that are not co-regulated (Fig. 2*b*).

We thus established the following measure of reliability: a transcription module is considered to be reliable if it is obtained from several distinct input sets (Fig. 1*b*). This recurrence property vanishes completely when randomized expression data, obtained by shuffling the components of the gene expression matrix, are used (see Web Fig. B online). The recurrence measure is defined mathematically in Methods.

Pathway analysis

Although many cellular pathways are known, only in a few cases have all of the participating genes and their regulatory relationships been characterized. The signature algorithm can be used to extend and refine partial knowledge about a pathway using the available expression data. Specifically, by applying the signature algorithm to a given set of genes that are thought to participate in a particular cellular function, it is possible to (i) reject genes that are mistakenly included, (ii) retrieve additional genes that are also likely to be involved in the pathway and (iii) identify the experimental conditions under which these genes are co-regulated. Notably, a reliability measure for these results is given by the recurrence property.

To verify the efficiency of this approach, we considered the well-studied tricarboxylic acid (TCA) cycle in *S. cerevisiae*. We applied our algorithm to a set of 37 yeast genes that are homologous to the known genes of the TCA cycle in *Escherichia coli*. The resulting transcription module included most of the genes that are known to be involved in this pathway (Fig. 3*a*). In particular, all misclassified genes in the input set did not appear in the output module. The yeast *SDH4* gene, which was not identified by homology but functions in the TCA cycle, was identified correctly. In addition, using different combinations of the TCA cycle genes as input sets, we identified two subparts of the cycle that are autonomously co-regulated in different cellular contexts (Fig. 3*b*,*c*).

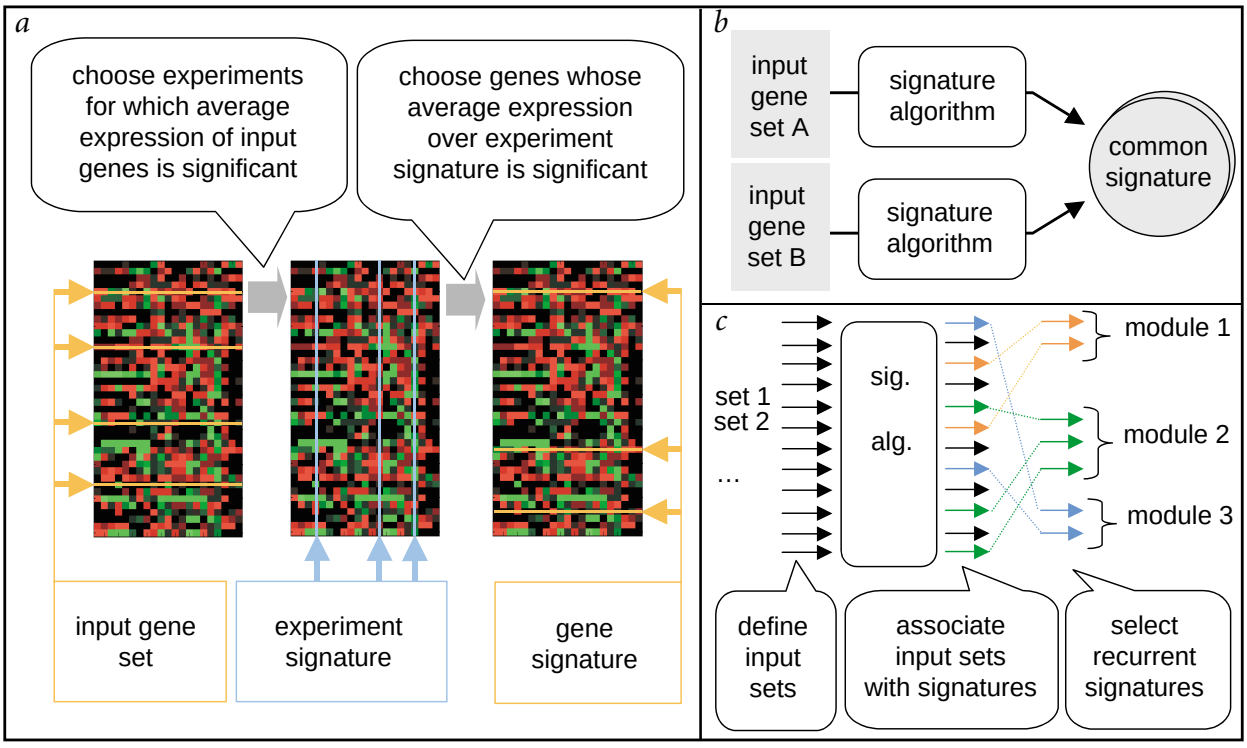

Fig. 1 The recurrence signature method. ***a***, The signature algorithm. ***b***, Recurrence as a reliability measure. The signature algorithm is applied to distinct input sets containing different subsets of the postulated transcription module. If the different input sets give rise to the same module, it is considered reliable. ***c***, General application of the recurrent signature method.

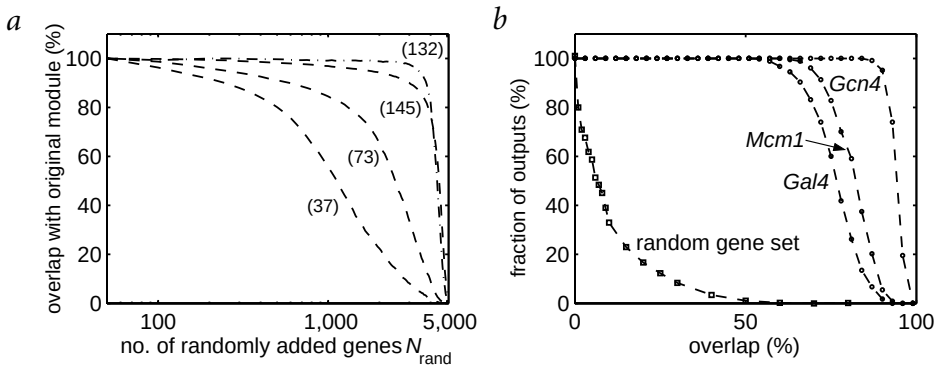

Fig. 2 The recurrence criteria. ***a***, A reference set of N_{core} co-regulated genes was composed of genes encoding either ribosomal proteins (dashed lines) or proteins involved in amino-acid biosynthesis (dashed/dotted line). (Other groups of co-regulated genes yielded similar results.) The recurrent signature method was applied to this set as follows. First, a collection of input sets was derived by randomly adding genes to the reference set. Second, the signature algorithm was applied to the reference set and to the derived sets; this generates a reference signature and a collection of perturbed signatures, respectively. Last, the overlaps between the reference signature and the perturbed signatures were calculated. Shown is the average overlap as a function of the number of genes added to the reference set. The different lines correspond to different choices of N_{core}, shown in parentheses. ***b***, The recurrent signature method was applied to three sequence-related references sets. These sets include all of the genes that contain the binding sequences $CGGN_{11}CCG$ (for Gal4), TGACTC (for Gcn4) or TTN_9GGAAA (for Mcm1) in a region of 600 bp upstream. Shown is the fraction of perturbed signatures whose overlap with the reference signature is greater than some threshold, as a function of this threshold. Note the large number of highly overlapping outputs for all three references sets. By contrast, the profile corresponding to a random sequence is distinctly different, with no large overlaps. Thus, the 'recurrence profile' gives a clear indication of whether a given sequence functions as a regulatory control element.

Identification of *cis*-regulatory elements

Although many *cis*-regulatory elements have been characterized, a motif sequence by itself is not sufficient to identify the genes that are regulated by the motif. In fact, most motif sequences that are found in the upstream region of genes have no regulatory function. The signature algorithm can be used to single out those genes that are co-regulated by the transcription factor associated with a particular regulatory motif.

For this task, we create an input set by collecting all of the genes that contain the relevant sequence in their upstream region and apply the algorithm to this set. The output of our algorithm provides both the co-regulated genes and the experimental conditions that induce this co-regulation. The reliability of this output is determined by the recurrence criterion described above (Fig. 2*b*). High recurrence indicates that the output genes are regulated by the corresponding transcription factor and that this factor is activated under the output conditions. In fact, this method can be applied not only to determine the exact regulatory context of known *cis*-regulatory elements, but also to assess whether any given DNA sequence is a *cis*-regulatory element.

We tested this approach by applying it to the known Gal4, Gcn4 and Mcm1 transcription-factor binding sites in *S. cerevisiae*. Specifically, we first composed three input sets from genes whose promoter regions contain the associated motifs ($CGGN_{11}CCG$, TGACTC and TTN_9GGAAA, respectively). A control set of genes corresponding to a random sequence was also assembled. We then applied the recurrent signature method to each of the four sets. The recurrence measure clearly distinguished the three known regulatory motifs from the random sequence (Fig. 2*b*). The transcription modules associated with the three sequences were biologically meaningful. For example, although the binding site for Gal4 appears in the upstream region of 213 genes, most of which are not connected to galactose use, only 15 genes were assigned to the associated transcription module. The top-scoring

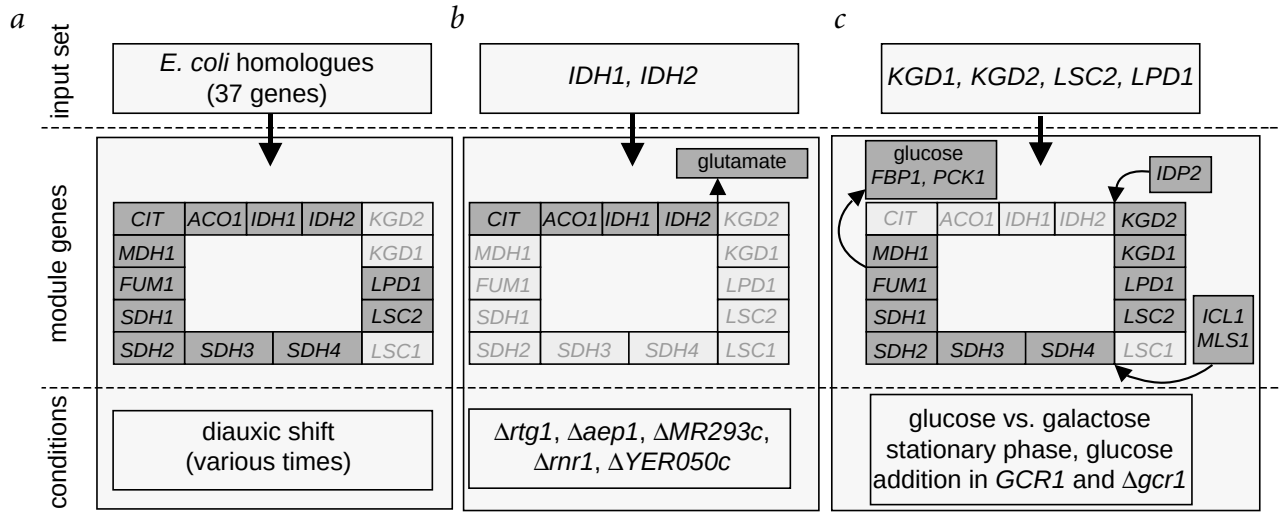

Fig. 3 Co-regulation of TCA cycle genes. ***a***, A standard BLAST search was carried out to find yeast homologs of the *E. coli* genes of the TCA cycle. Applying the recurrent signature method to the input set comprising these homologs yields only genes that are involved in the yeast TCA cycle. The TCA cycle genes are shown; those that are assigned to the modules are highlighted (dark background). ***b,c,*** Two subsets of the TCA cycle are found to be independently co-regulated. *b,* Genes upstream of α-ketoglutarate (α-KG), a primary precursor of glutamate, are found to be co-regulated under experimental conditions of deletion of *RTG1* and deletions of genes that affect mitochondrial function, *YMR293c*, *AEP1*, *YER050c* and *RNR1* (ref. 12). In fact, it has been reported that the expression of these genes becomes Rtg1 dependent when mitochondrial respiration capacity is compensated[13]. *c,* Under a different set of conditions, the genes upstream of α-KG are co-expressed with genes whose expression is dependent on the transcriptional activator Cat8, which suggests that they are involved in gluconeogenesis[14].

genes included the known targets *GAL10*, *GAL1*, *GAL7*, *GAL3*, *GAL2*, *GAL80* and *GCY1* and a few newly described targets. The shift of yeast cells from use of glucose to galactose and related changes in sugar metabolism were the top-scoring experimental conditions assigned to this module. Notably, several experimental conditions related to the cell cycle were found to belong this module, which suggests that it also has a regulatory association.

The above two applications of our approach to extend previous biological knowledge are available on our website, where researchers can submit their own candidate groups of genes, as well as sequences suspected to function as regulatory elements.

Global study of the yeast transcription modules

The capability of the signature algorithm to retrieve a transcription module from any set of genes that shares a sufficiently large number of genes with this module allows us to use this method in the global decomposition of the yeast genome into transcription modules (Methods). We therefore applied the signature algorithm to a diverse collection of input sets derived in three different ways. First, sequence-related input sets were assembled from all of the genes that contain a particular sequence in their upstream region. All possible combinations of six, seven and eight nucleotides were considered, resulting in a total of $4^6 + 4^7 + 4^8 \approx 86{,}000$ input sets. Second, function-related input sets were defined according to the classification in the Munich Information Center for Protein Sequences (MIPS) database[7]. Third, cluster-related input sets were constructed from the output of a hierarchical cluster algorithm (which clusters the full expression data)[5]. The signature algorithm was applied to all of these input sets. Only the recurrent output sets were used to identify the transcription modules (Fig. 1*c*).

A comprehensive description of the transcription modules that we obtained is given in Web Table A online and on our website, and we highlight only a few of the results of our global analysis here. We identified 86 overlapping transcription modules, comprising a total of 2,241 genes. The function of 927 of these genes is unknown, according to the Yeast Proteome Database (YPD)[8,9]. We found that the genes of most modules participate in a module-specific cellular process (Fig. 4). Thus, functional links can be assigned reliably to the genes of unknown function in each module.

Experimental validation of functional assignment

To examine the predictive power of our results, we checked experimentally the involvement of two genes, *SDA1* and *LTV1*, in rRNA processing. Both genes were assigned to module 10, which includes 48 genes. Although the functions of most genes in this module are unknown, many observations indicate that it includes genes involved in rRNA processing (Fig. 4*a*). First, most of the functionally annotated genes are involved in rRNA processing. Second, all of the genes with known cellular localization are found in the nucleus, with most located in the nucleolus. Third, most of the genes are necessary for viability.

We examined *SDA1* and *LTV1* because of the availability of temperature-sensitive phenotypes. Temperature-sensitive mutants of *SDA1* (*sda1-2*) show severe actin depolymerization and cell-cycle arrest[10,11], and a strain lacking *LTV1* (*ltv1*) shows lethality at low temperatures (see YPD[8,9]). Before our study, neither of these genes had been associated with rRNA processing. We carried out northern-blot analysis to detect various pre-rRNA intermediates (Fig. 5). Several pre-rRNA intermediates accumulated on transfer of either mutant strain to the restrictive temperature, indicating that *LTV1* and *SDA1* are involved in rRNA processing.

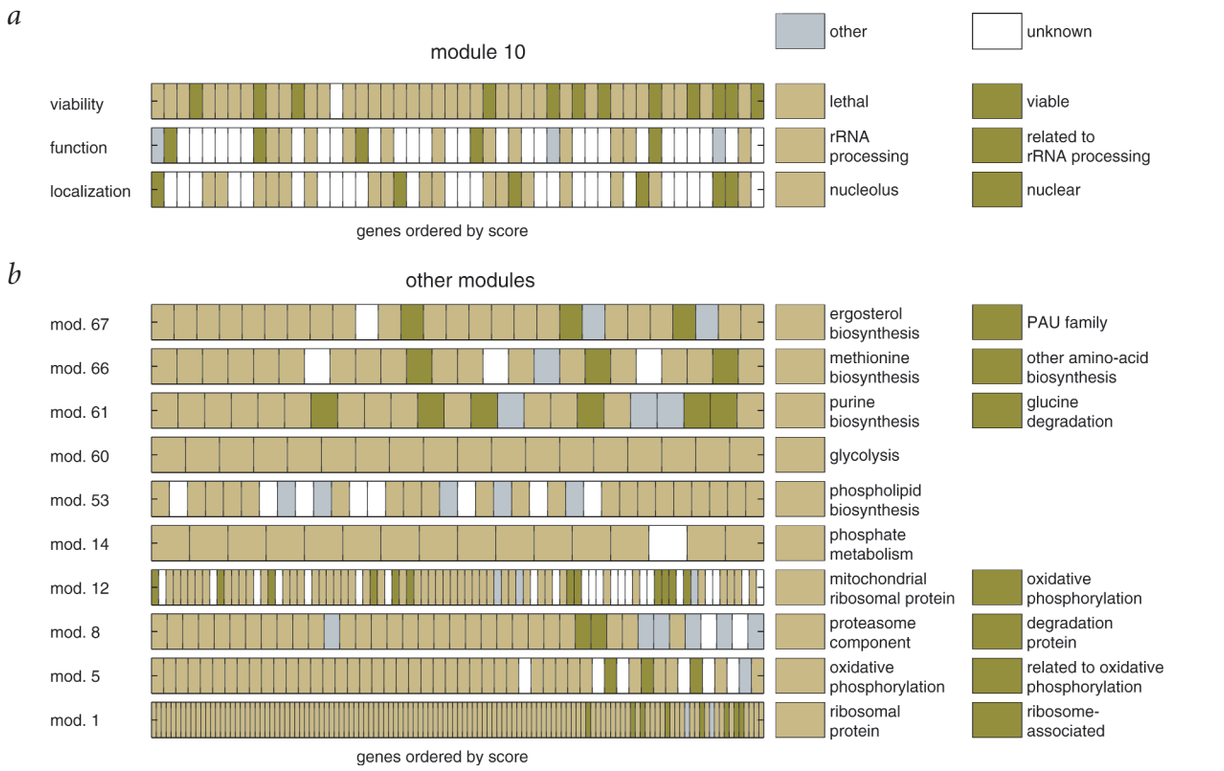

Fig. 4 Functional consistency of transcription modules. Each row represents a module; boxes refer to the genes, ordered from left to right according to score. The gene properties are color coded using information in the YPD and other sources. ***a***, Viability, function and localization of the genes in module 10. Characterization (if known) is consistent, except for a few genes. We verified experimentally the involvement of the first gene (*SDA1*, annotated here as unrelated) in pre-rRNA processing (Fig. 5). The involvement of three other genes in rRNA processing (*YGR103w*, *YOR206w* and *YLR002c*, annotated here as unknown) has been shown[15,16]. ***b***, Functional consistency for a selection of modules. Similar consistency was also observed for most of the other modules.

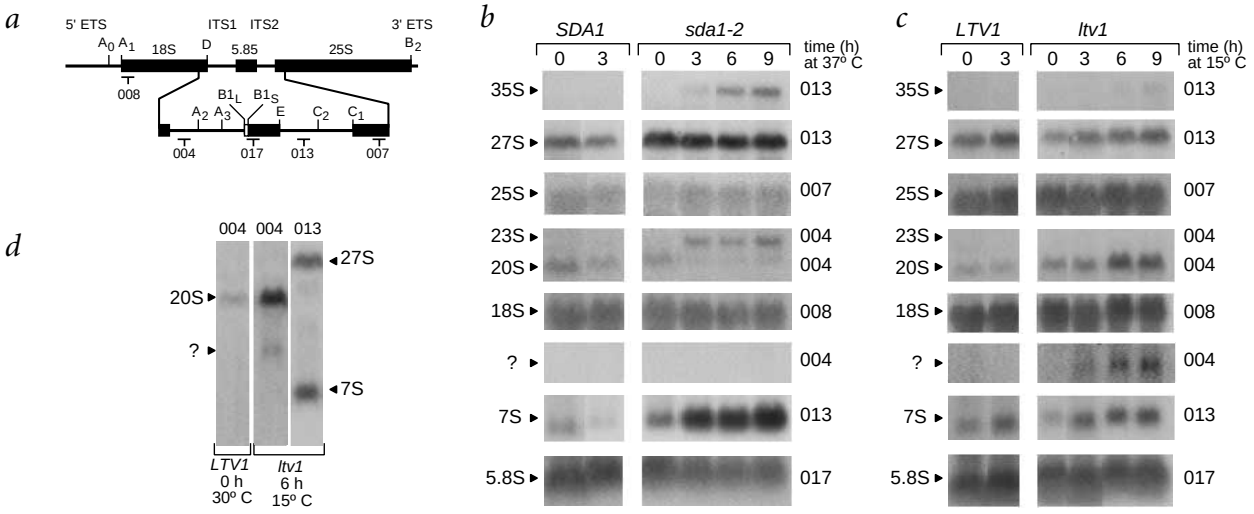

Fig. 5 Experimental verification of the involvement of Sda1 and Ltv1 in rRNA processing. ***a***, Structure and processing sites of the 35S pre-rRNA, which contains sequences for the mature 18S, 5.8S and 25S rRNAs separated by two internal transcribed spacers (ITS1 and ITS2) and flanked by the two external transcribed spacers (5′ ETS and 3′ ETS). Oligonucleotide probes were chosen according to ref. 15, and their position is indicated. ***b–c***, Northern-blot analysis of pre-rRNA. Cultures of *ltv1*, *sda1-2* and the associated wild-type strains were grown to early log phase and either maintained at the permissive temperature or shifted to restrictive temperatures, 15 °C (*ltv1*) or 37 °C (*sda1-2*). Total RNA was isolated 3, 6 and 9 h after the temperature shift. The positions of the mature rRNAs and pre-rRNAs are indicated on the left, and the oligonucleotides used are on the right. *b*, Transfer of the *sda1-2* strain to the restrictive temperature leads to an accumulation of 35S pre-rRNA, the appearance of 23S pre-rRNA, and a marked reduction in 20S pre-rRNA, which indicates a delay in cleavage at sites A_0, A_1 and A_2. Accumulation of 7S pre-rRNA and a parallel reduction in the 5.8S rRNA are also seen. *c*, Transfer of the *ltv1* strain to the restrictive temperature leads to the accumulation of 20S pre-rRNA. An additional RNA intermediate, which to our knowledge has not been reported previously, seems to result from improper cleavage of the 20S pre-rRNA. ***d***, The position of the new cleavage product is shown relative to that of known intermediates. The oligonucleotides used are indicated on the top.

Higher-order relations in the transcription program

In addition to the co-regulated genes, each transcription module also includes the experimental conditions that regulate those genes. This 'experimental signature' provides valuable information about the function of the module. In addition, it can be used to reveal higher-order relations between different modules (Fig. 6; an annotated version is available on our website). For example, the experimental signatures of module 10 (which is associated with rRNA processing) and module 20 (which is related to stress response) are composed essentially of the same experiments, albeit with experimental scores that have opposite signs.

This strong inverse correlation indicates that rRNA processing is repressed under most stress conditions. Similarly, most conditions that induce the mating genes (module 6) repress the genes that are involved in the G1/S transition during the cell cycle (module 13), reflecting the G1 arrest that accompanies the mating response. By contrast, the genes associated with the mitochondrial ribosomal proteins (module 12) are not affected by most conditions that regulate the ribosomal proteins (module 1).

In silico evaluation of the signature algorithm

Because the transcriptional networks that underlie measured expression data are not known, it is difficult to compare the results of our analyses systematically with those obtained by other methods. We therefore used computer-generated expression data for this task. The important aspect of the *in silico* generation of the expression data was the modeling of overlapping transcription modules. We therefore assigned a regulatory logic to each gene in the genome that determines which combination of the transcription factors regulates its expression. We generated the expression data by collecting the 'response' of all of the genes for many different combinations of the active transcription factors under different experimental conditions. In this controlled setup, each transcription module corresponds to a single transcription factor and includes the genes that are regulated by this factor and the conditions in which this factor is active. The number of transcription factors that regulate each gene determines the degree of overlap between the modules (Methods).

We applied our recurrent signature algorithm to the computer-generated expression matrix that resulted from the above model. Using a sufficiently diverse collection of random input sets, essentially all of the transcription modules could be successfully identified, even in the case of highly overlapping modules. By contrast, applying hierarchical clustering algorithms to the same expression matrix captured only small, incomplete fractions of these modules, owing to the fact that these methods do not allow the multiple assignment of a gene to different modules (Fig. 7).

Discussion

Although the potential of the information contained in large and diverse databases of genome-wide expression profiles is well recognized, the extraction of meaningful biological knowledge from such data remains a challenging task. Our recurrent signature approach described here offers several advantages over commonly used methods of gene classification.

First, genes are classified on the basis of co-regulation under a subset of the experimental conditions, rather than on the basis of co-regulation under all the experimental condition. Thus, each transcription module specifies not only the co-regulated genes but also their regulatory context. Second, genes may be assigned to several overlapping modules—a property that is essential for capturing the biologically relevant combinatorial regulation. Third, our method provides a simple and intuitive means of integrating additional biological information, such as functional annotation or sequence information, with the analysis of gene expression data. Last, the computation time of our algorithm depends linearly on the size of the data set (S.B., J.I. and N.B., unpublished data). This computational efficiency will be crucial to our ability to deal with rapidly growing data sets and to extend the analysis to higher eukaryotes.

The recurrent signature method can be used to advance two complementary aspects of genetic research. First, for studying specific cellular functions, we offer practical applications for the refinement and extension of existing knowledge about cellular pathways and for the identification of *cis*-regulatory elements. Although we have verified both applications by focusing on well-studied examples, our website allows them to be used for any problem of interest. Those applications provide researchers with a simple way to use our compilation of almost all of the available genome-wide yeast expression data, which includes more than 1,000 experimental conditions.

The second complementary application is for studying the global structure of the transcription program. By systematically integrating genomic sequence information and functional annotation with the available yeast expression data, we identified many modules and characterized their relationships. The assignment of functionally annotated genes to these modules is highly consistent. Thus, the database that summarizes our results provides functional links to almost 1,000 yeast genes of unknown function. Notably, most of the identified modules are associated with known *cis*-regulatory elements—a fact that reflects the large amount of knowledge for the model organism *S. cerevisiae* and underlines the efficiency of our method.

We evaluated our approach systematically in a controlled *in silico* setting. In such a setting, the desired output of the analysis is known, and thus we could rigorously evaluate the performance of different methods. Using artificially created expression data, we verified that our algorithm is well suited for identifying overlapping groups of co-regulated genes and clearly outperforms standard clustering. Further support for our method was provided by experiments that confirmed our functional prediction for two previously uncharacterized genes.

Finally, an alternative strategy to reveal the modular structure is to assemble a large number of random gene sets and to apply the signature algorithm iteratively, by using the output gene signatures as new input sets. This approach requires no existing information. Therefore, it is particularly applicable in the absence of good annotation, as is generally the case for higher eukaryotes. Preliminary investigations show that such an iterative procedure converges on most of the yeast modules identified by the global analysis presented here. Further exploration of the relationship between the fixed points of this iterative scheme should provide insight into the modular structure of the gene expression data and will be pursued elsewhere.

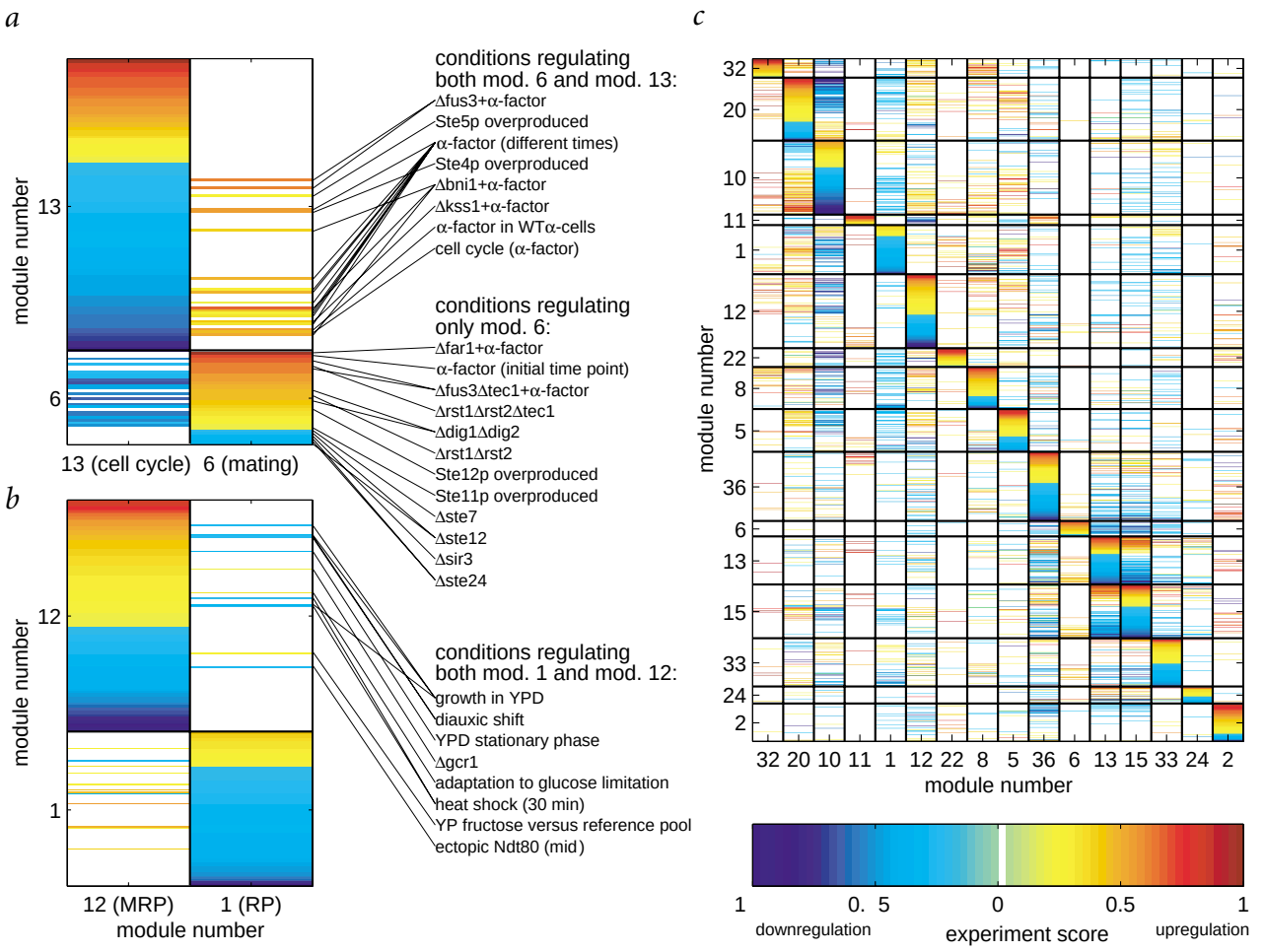

Fig. 6 Correlation between the regulatory contexts of different modules. ***a***,***b***, The conditions assigned to two different modules are compared using the matrix display shown. The four boxes of the matrix are composed of discrete lines, which represent all of the conditions assigned to the module on the vertical axis. The lines are color coded according to the score assigned to those experiments in the module on the horizontal axis. Thus, whereas the diagonal boxes are filled, fewer lines are seen in the off-diagonal blocks, which correspond to the experimental conditions that are shared by the two modules. Comparison of the color of the lines between the diagonal and off-diagonal boxes reveals whether the regulation of the two modules is negatively or positively correlated. In this example, module 1 contains ribosomal proteins (RP), module 6 contains genes involved in mating, module 12 contains mitochondrial ribosomal proteins (MRP) and module 13 is associated with the G1/S transition during the cell cycle. ***c***, Global visualization of the correlation between many modules. The individual experiments can be viewed on our website.

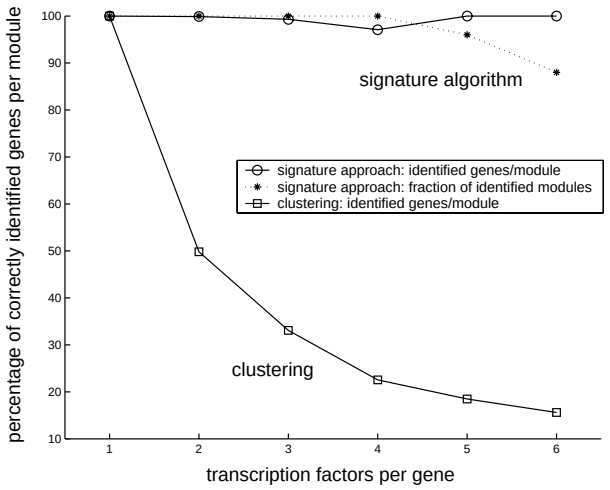

Fig. 7 Comparison between the signature approach and clustering using *in silico* expression data (Methods). The expression level of 1,050 genes was regulated by 25 transcription factors. On average, 20% of the transcription factors was active at each of the 1,000 conditions. The graph shows the average fraction of correctly identified genes per module for the signature approach (circles) and per cluster for hierarchical clustering (squares) as a function of the number of transcription factors regulating each gene. The number of clusters was fixed to the number of modules (25). Note that the percentage of modules correctly identified by the signature algorithm (indicated by asterisks) was less than 100% only for high combinatorial regulation (five or more transcription factors regulating each gene). This fraction depends on the number of input sets as well as the number of experimental conditions.

Methods

The signature algorithm. The element E^{gc} of the gene expression matrix contains the log-expression change of gene $g \in G = \{1, ..., N_G\}$ at the experimental condition $c \in C = \{1, ..., N_C\}$, where N_G and N_C denote the total number of genes and conditions, respectively. We introduce two normalized expression matrices E_G^{gc} and E_C^{gc}, which have zero mean and unit variance with respect to genes and conditions, respectively:

$$\left\langle E_G^{gc} \right\rangle_{g\in G} = 0,\ \left\langle (E_G^{gc})^2 \right\rangle_{g\in G} = 1,$$

and

$$\left\langle E_C^{gc} \right\rangle_{c\in C} = 0,\ \left\langle (E_C^{gc})^2 \right\rangle_{c\in C} = 1,$$

where $\langle ... \rangle_x$ denotes the average with respect to x. The input set consists of N_I genes:

$$G_I = \{g_1, \ldots, g_{N_I}\} \subset G$$

In the first step of the signature algorithm, we score each experimental condition by the average expression change over the genes of the input set. The condition score is

$$s_c = \left\langle E_G^{gc} \right\rangle_{g\in G_I}$$

The experiment signature S_C contains those conditions whose absolute score is statistically significant:

$$S_C = \left\{ c \in C : |s_c - \left\langle s_c \right\rangle_{c\in C}| > t_C \sigma_C \right\}$$

In our analysis, we used $t_C = 2.0$ as the condition threshold level and the standard deviation expected for random fluctuations of

$$\sigma_C = 1/\sqrt{N_I}$$

In the second step, we score all genes by the weighted average change in the expression within the experimental signature. The gene score is

$$s_g = \left\langle s_c E_C^{gc} \right\rangle_{c\in S_c}$$

The gene signature S_G contains those genes whose score is statistically significant:

$$S_G = \left\{ g \in G : s_g - \left\langle s_g \right\rangle_{g\in G} > t_G \sigma_G \right\}$$

We used $t_G = 3.0$ as gene threshold and the measured standard deviation σ_G.

Fusion of signatures in the analysis of pathways and sequence elements. We apply the signature algorithm to a reference input set G_I^{ref} and to a set of input sets $\{G_I^{(i)}\}$ that are obtained from G_I^{ref}. Each set contains a fraction of the genes in $G_I^{(i)}$ and some unrelated genes that were selected at random. The result is a reference signature S_{ref} and a collection of modified signatures $\{S_i\}$. The overlap between any of these signatures and the reference signature is defined as

$$OL_i^{\mathrm{ref}} = \frac{|S_i \cap S_{\mathrm{ref}}|}{\sqrt{|S_i| \cdot |S_{\mathrm{ref}}|}},$$

where $|\ldots|$ refers to the size of a set and $\cap$ denotes intersection. All signatures S_i whose overlap with the reference signature exceeds a certain threshold are included in the set of recurrent signatures

$$R = \left\{ S_i : OL_i^{\mathrm{ref}} > t_R \right\}$$

The threshold t_R must be chosen to be large enough to discriminate against random fluctuations, but small enough to include a significant fraction of signatures. In general, we used $t_R = 70\%$, but our results have been robust with respect to the exact value chosen. Finally, a module is obtained by selecting only those genes that appear in at least 80% of all signatures in R. All genes within a module are assigned a score according to the average of their gene scores in all the signatures in R. The module conditions are defined correspondingly.

Fusion of signatures in the global analysis. The procedure that we used to generate modules from recurrent signatures resembles agglomerative clustering, albeit for signatures rather than genes. We considered pairs of recurrent signatures $\{S_i, S_j\}$ obtained from the three classes of input sets used in the global analysis. For sequence-related signatures, we searched for pairs of overlapping signatures that were associated with sequences differing by a single nucleotide or that were the inverse complements of each other. Because the two input sets associated with each of those pairs are essentially distinct, a large overlap between the corresponding signatures indicates that both sequences bind to the same transcription factor. This overlap requirement is important to distinguish sequences involved in the regulation of a module from those that are merely overrepresented. We also searched for coinciding pairs of function-related or cluster-related signatures. Here we considered all the pairs and selected those with the highest overlap.

The pairs of recurrent signatures were fused into transcription modules as follows. For each pair, we computed the intersect $P_{ij} = S_i \cap S_j$ of genes appearing in both signatures as well as the overlap

$$OL_{ij} = |P_{ij}| / \sqrt{|S_i| \cdot |S_j|}$$

We selected the pair signature P_{ref} with the largest associated overlap OL_{ref} as the 'seed' of a new module. We then assigned all pair signatures P_{ij} whose overlap with P_{ref} exceeded a certain fraction t_R of OL_{ref} to the set of recurrent signatures R—that is,

$$R = \left\{ P_{ij} : OL(P_{ij}, P_{\text{ref}}) > t_R \, OL_{\text{ref}} \right\}$$

The gene content and scores of the associated module were obtained from R as described above. Subsequently, we removed the pairs that had been assigned to R from the total 'pool' of pair signatures $\{P_{ij}\}$. To avoid the identification of more, less-coherent realizations of the same module, we also removed those pairs from $\{P_{ij}\}$ that would have been assigned to R for a somewhat lower value of the threshold t_R, unless they had a significant (~75%) overlap with any other pair signature. This process was iterated until all sets were assigned.

Analysis of computer-generated expression data. We generated the *in silico* expression data as follows. The regulation of each gene was encoded by a 'promoter matrix' whose elements $P_{tg} \in \{0,1\}$ specify whether the transcription factor $t \in \{1, ..., N_T\}$ activates (1) or does not affect (0) gene $g \in \{1, ..., N_G\}$. In our analysis, we considered a total of $N_G = 1{,}050$ genes regulated by $N_T = 25$ transcription factors. The log expression of gene g at condition c was defined as

$$E^{gc} = \sum_{t=1}^{N_t} P_{tg} A_{tc},$$

where $A_{tc} \in \{0,1\}$ specifies the activity of transcription factor t at condition c. Five randomly chosen transcription factors were active at each condition, on average, and we considered an expression matrix generated from 1,000 conditions. The recurrent signature algorithm was applied to 2,000 initial random input sets. The resulting output sets were reused as input sets and this procedure was repeated three times (details of this iterative scheme will be published elsewhere). The fusion of the resulting signatures to transcription modules was carried out precisely, as in the global analysis of the yeast expression data described above. We used $t_C = 1.5$ and $t_G = 1.0, \ldots, 2.5$ as condition and gene thresholds, respectively. Modules were considered reliable if they were recovered at least three times with an overlap of 80%. A gene was assigned to the final module if it appeared in at least two of such overlapping modules. For the hierarchical clustering, we used the standard Matlab clustering functions and a previously described algorithm[5].

Strains and microbiological techniques. Standard techniques were used to grow and handle the yeast strains. We used the following strains: ZZ28-a (*sda1-2*) and its parental strain dk186-a (*lue2-3,112*, *trp1-1*, *his3-11*, *ura3-52*, *ade2-1*, *can1-100*, *GAL+*, *Δbar1*), which were provided by D.R. Kellogg; and BY4742-ltv1Δ-α (*ltv1Δ*) and its parental strain BY4742-α (*lue2-Δ0*, *his3-Δ1*, *ura3-Δ0*), which were purchased from EUROSCARF.

Northern-blot analysis. Equal amounts of total RNA were separated by agarose gel electrophoresis, blotted and hybridized with labeled oligonucleotide probes complementary to different regions of the pre-rRNA transcript. The oligonucleotides were chosen according to ref. 15 (sequences available on request).

URLs. More details of our results and the applications of our method, together with the list of all expression profiles used, can be found on our website (http://www.weizmann.ac.il/home/barkai).

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

Acknowledgments

We thank D.R. Kellogg for the sda2-1 strain; U. Alon, M. Dolev, E. Domany, A. Eldar, O. Gileadi, Y. Kafri, B.-Z. Shilo and S. Shnider for discussions and comments on the manuscript; G. Jona and O. Gileadi for experimental help. This work was supported by the US National Institutes of Health, the Israeli Science Ministry and the Benoziyo center. S.B. is a Koshland fellow. N.B. is the incumbent of the Soretta and Henry Shapiro career development chair.

Competing interests statement

The authors declare that they have no competing financial interests.

Received 26 April; accepted 24 June 2002.

1. Bittner, M., Meltzer, P. & Trent, J. Data analysis and integration: of steps and arrows. *Nature Genet.* **22,** 213–215 (1999).
2. Cheng, Y. & Church, G.M. Biclustering of expression data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8,** 93–103 (2000).
3. Getz, G., Levine, E. & Domany, E. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl Acad. Sci. USA* **97,** 12079–12084 (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. Alon, U. *et al.* Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA* **96,** 6745–6750 (1999).
6. Tamayo, P. *et al.* Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA* **96,** 2907–2912 (1999).
7. Mewes, H. W. *et al.* MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.* **30,** 31–34 (2002).
8. Costanzo, M. C. *et al.* The yeast proteome database (YPD) and *Caenorhabditis elegans* proteome database (WormPD): comprehensive resources for the organization and comparison of model organism protein information. *Nucleic Acids Res.* **28,** 73–76 (2000).
9. Costanzo, M.C. *et al.* YPD, PombePD & WormPD: model organism volumes of the BioKnowledge library, an integrated resource for protein information. *Nucleic Acids Res.* **29,** 75–79 (2001).
10. Zimmerman, Z.A. & Kellogg, D.R. The Sda1 protein is required for passage through start. *Mol. Biol. Cell* **12,** 201–219 (2001).
11. Buscemi, G., Saracino, F., Masnada, D. & Carbone, M.L. The *Saccharomyces cerevisiae SDA1* gene is required for actin cytoskeleton organization and cell cycle progression. *J. Cell Sci.* **113,** 1199–1211 (2000).
12. Fikus, M. U. *et al.* The product of the DNA damage-inducible gene of *Saccharomyces cerevisiae*, *DIN7*, specifically functions in mitochondria. *Genetics* **154,** 73–81 (2000).
13. Liu, Z. & Butow, R. A. A transcriptional switch in the expression of yeast tricarboxylic acid cycle genes in response to a reduction or loss of respiratory function. *Mol. Cell. Biol.* **19,** 6720–6728 (1999).
14. Bojunga, N. & Entian, K. D. Cat8p, the activator of gluconeogenic genes in *Saccharomyces cerevisiae*, regulates carbon source-dependent expression of NADP-dependent cytosolic isocitrate dehydrogenase (Idp2p) and lactate permease (Jen1p). *Mol. Gen. Genet.* **262,** 869–875 (1999).
15. Milkereit, P. *et al.* Maturation and intranuclear transport of pre-ribosomes requires Noc proteins. *Cell* **105,** 499–509 (2001).
16. Harnpicharnchai, P. *et al.* Composition and functional characterization of yeast 66S ribosome assembly intermediates. *Mol. Cell* **8,** 505–515 (2001).