nature biotechnology

Computational discovery of gene modules and regulatory networks

Ziv Bar-Joseph[1,4], Georg K Gerber[1,4], Tong Ihn Lee[2,4], Nicola J Rinaldi[2,3], Jane Y Yoo[2], François Robert[2], D Benjamin Gordon[2], Ernest Fraenkel[2], Tommi S Jaakkola[1], Richard A Young[2,3] & David K Gifford[1]

We describe an algorithm for discovering regulatory networks of gene modules, GRAM (Genetic Regulatory Modules), that combines information from genome-wide location and expression data sets. A gene module is defined as a set of coexpressed genes to which the same set of transcription factors binds. Unlike previous approaches[1–5] that relied primarily on functional information from expression data, the GRAM algorithm explicitly links genes to the factors that regulate them by incorporating DNA binding data, which provide direct physical evidence of regulatory interactions. We use the GRAM algorithm to describe a genome-wide regulatory network in *Saccharomyces cerevisiae* using binding information for 106 transcription factors profiled in rich medium conditions data from over 500 expression experiments. We also present a genome-wide location analysis data set for regulators in yeast cells treated with rapamycin, and use the GRAM algorithm to provide biological insights into this regulatory network.

High-throughput biological data sources hold the promise of revolutionizing molecular biology by providing large-scale views of genetic regulatory networks. Many genome-wide expression data sets are now readily available, and typical computational analyses have applied clustering algorithms to expression data to find sets of coexpressed and potentially coregulated genes[1]. Recent approaches have used more sophisticated algorithms; one group of researchers constructed a probabilistic model that uses expression data to link regulators to regulated genes[2]. Their method relies on the assumption that the expression levels of regulated genes will depend on the expression levels of regulators, which is a limitation in cases in which the expression level of the regulator does not change appropriately (*e.g.*, cases of post-transcriptional modification). Other approaches have combined expression data with additional information, such as shared DNA binding motifs or Munich Information Center for Protein Sequences (MIPS) categories[3–5], but the use of these data sources provides essentially only functional or indirect evidence of genetic regulatory interactions. These methods cannot reliably distinguish among genes that have similar expression patterns but are under the control of different regulatory networks (see **Supplementary Note** online for further details).

Large-scale, genome-wide location analysis for DNA-binding regulators offers a second means for identifying regulatory relationships[6]. Location analysis identifies physical interactions between regulators and DNA regions, providing strong direct evidence for genetic regulation. Although helpful, the usefulness of binding information is also limited, as the presence of the regulator at a promoter region indicates binding but not function. The regulator may act positively, negatively or not at all. In addition, as with all microarray-based data sources, location analysis data contain substantial experimental noise. Because expression and location analysis data provide complementary information, our goal was to develop an efficient computational method for integrating these data sources. We expected that such an algorithm could assign groups of genes to regulators more accurately than methods based on either data source alone.

The GRAM algorithm begins by performing an efficient, exhaustive search over all possible combinations of transcriptional regulators indicated by the DNA-binding data with a stringent criterion for determining binding. Once a set of genes to which a common set of transcriptional regulators binds is found, the algorithm identifies a subset of these genes with highly correlated expression, which serves as a 'seed' for a gene module. The algorithm then revisits the binding data and, using a relaxed binding criterion, seeks to add additional genes to the module that are similarly expressed and to which the same set of transcriptional regulators binds. Our algorithm allows genes to belong to more than one module. (See the Methods section for a complete description of the GRAM algorithm.)

The GRAM algorithm was applied to genome-wide location data for 106 transcription factors and over 500 expression experiments (details on the data used are available in **Supplementary Table 1** online). We identified 106 gene modules, containing 655 distinct genes and regulated by 68 of the transcription factors. **Figure 1** presents a visualization of these results as a graph with edges between gene modules and regulators.

The gene modules abstraction allowed us to label regulator-module edges in the graph to indicate whether there is significant evidence ($P < 0.05$) that regulators may be functioning as activators. Because a gene module provides a link between a set of regulators and the common expression pattern of a set of genes to which the regulators bind, we can use the relationship between a regulator's expression pattern

[1]MIT Computer Science and Artificial Intelligence Laboratory, 200 Technology Square, Cambridge, Massachusetts 02139, USA. [2]Whitehead Institute for Biomedical Research, Nine Cambridge Center, Cambridge, Massachusetts 02142, USA. [3]MIT Department of Biology, 31 Ames Street, Room 68-132, Cambridge, Massachusetts 02139, USA. [4]These authors contributed equally to this work. Correspondence should be addressed to D.G. (gifford@mit.edu).

Published online 12 October 2003; doi:10.1038/nbt890

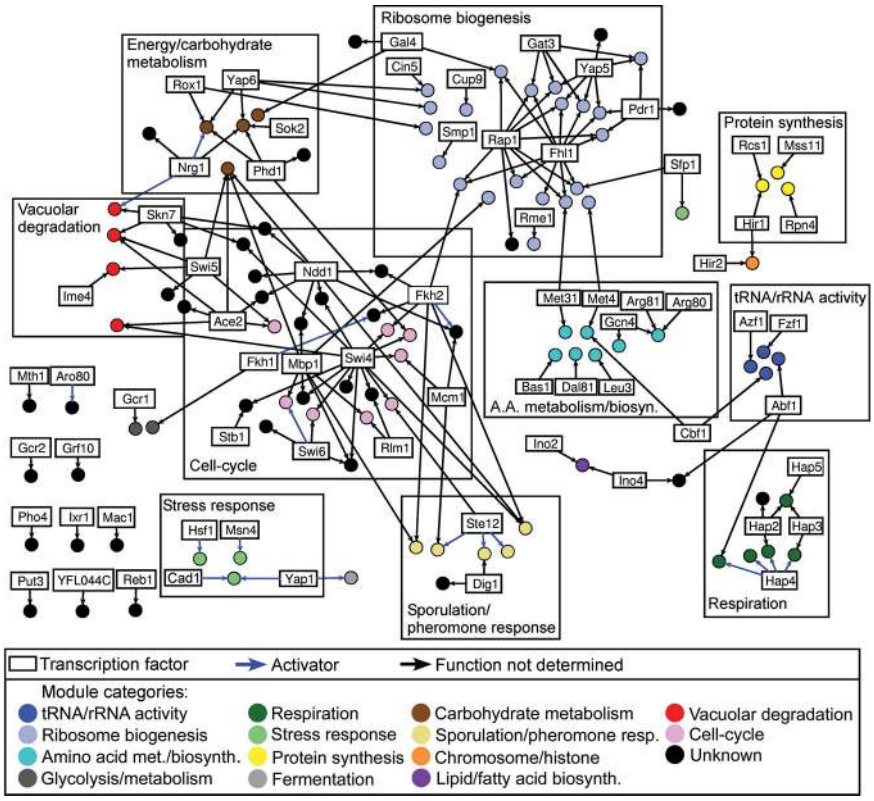

Figure 1 Rich medium gene modules network. Visualization of the transcriptional regulatory network discovered by the GRAM algorithm as a graph with edges between gene modules and regulators shows that there are many groups of connected gene modules and regulators involved in similar biological processes. The network consists of 106 modules containing 655 distinct genes regulated by 68 transcription factors. In most cases in which a gene module is controlled by one or more regulators, there was previous evidence suggesting that these regulators interact physically or functionally (see **Supplementary Table 3** online for details). The directed arrows point from transcription factors to the gene modules that they regulate. Blue arrows indicate discovered activator regulatory relationships (see **Supplementary Table 2** online and the text for details). Gene modules are colored according to the MIPS category to which a significant number of genes belong (significance test using the hypergeometric distribution $P < 0.005$). Modules containing many genes with unknown function or an insignificant number belonging to the same MIPS category are colored black. When the gene modules discovered by the GRAM algorithm were compared to results generated using location data alone, the GRAM algorithm yielded almost three times as many modules significantly enriched for genes in the same MIPS category.

and the common expression pattern of genes in a module to infer whether a regulator acts as an activator. In contrast, the use of genomic location data alone allows us only to infer the presence of regulators at promoters, but not to determine the type of interaction. We searched for activator relationships by examining regulators with expression profiles that are positively correlated with the expression profiles of genes in the corresponding modules. Positive correlation indicates that higher levels of regulator expression correlate with higher levels of expression of genes in the module and suggests that the transcription factor positively regulates the expression of genes in the module. We determined the statistical significance of the activator relationships by computing correlation coefficients between all transcriptional regulators studied and all gene modules and taking the 5% positive tail of the distribution of correlation coefficients. **Supplementary Table 2** online presents the 11 activators identified using the method described above. Ten of these were previously identified in the literature, suggesting that this analysis produces biologically meaningful results.

Several findings obtained by analysis of the discovered gene modules suggest that the algorithm identifies biologically relevant groupings of genes. First, we found that gene modules generally identify groups of genes that function in a similar biological pathway as defined by the MIPS functional categorization[7] (see **Fig. 1** and **Supplementary Table 3** online for details). Second, we found the gene modules to be generally accurate in assigning regulators to sets of genes whose functions are consistent with the regulators' known roles. As an example, Gcr1 is a well-characterized regulator of glucose metabolism[8,9]; six of the seven genes identified in the Gcr1 module are enzymes involved in glycolysis and gluconeogenesis. Additionally, we found that in most cases in which a gene module is controlled by one or more regulators, there was previous evidence suggesting that these regulators interact physically or functionally (see **Supplementary Table 4** online). For example, gene modules identify Hap2-Hap3-Hap4-Hap5, Hap4-Abf1, Ino2-Ino4, Hir1-Hir2, Mbp1-Swi6 and Swi4-Swi6 interactions. Taken together, these results provide evidence that the GRAM algorithm identifies not only biologically related sets of genes, but also relevant factors that are interacting to control the genes.

Although genome-wide location data alone are potentially useful for deriving transcriptional regulatory networks, a key feature of the GRAM algorithm is its ability to compensate for technical limitations in the location data through the integration of expression data. To determine binding events in location data, researchers have previously used a statistical model and chosen a relatively stringent *P*-value threshold (0.001) with the intention of reducing false positives at the expense of false negatives[6]. The GRAM algorithm presents a useful alternative to using a single *P*-value threshold to predict binding events, because our method allows the *P*-value cutoff to be relaxed if there is sufficient supporting evidence from expression data. As an example, consider Hap4, a well-characterized regulator of genes involved in oxidative phosphorylation and respiration[10]. The Hap4 modules contain 28 genes that are involved in respiration and show a high degree of coregulation over the collected expression data sets (**Fig. 2**). Six of these genes (*PET9, ATP16, KGD2, QCR6, SDH1* and *NDI1*) would not have been identified as Hap4 targets using the stringent 0.001 *P*-value threshold (*P*-values range from 0.0011 to 0.0036). Overall, 627 of 1,560 unique regulator-gene interactions (40%) in the rich medium network discovered by the GRAM algorithm would not have been detected using only location data and the stringent *P*-value cutoff.

To further verify the ability of the GRAM algorithm to lower the rate of false negatives without substantially increasing the rate of false positives, we performed gene-specific chromatin-immunoprecipitation (IP) experiments for the factor Stb1 and 36 genes. The profiled genes were picked randomly from the full set of yeast genes, with representatives selected from four *P*-value ranges. In these experiments, we found that Stb1 bound to three additional genes that had *P*-values between

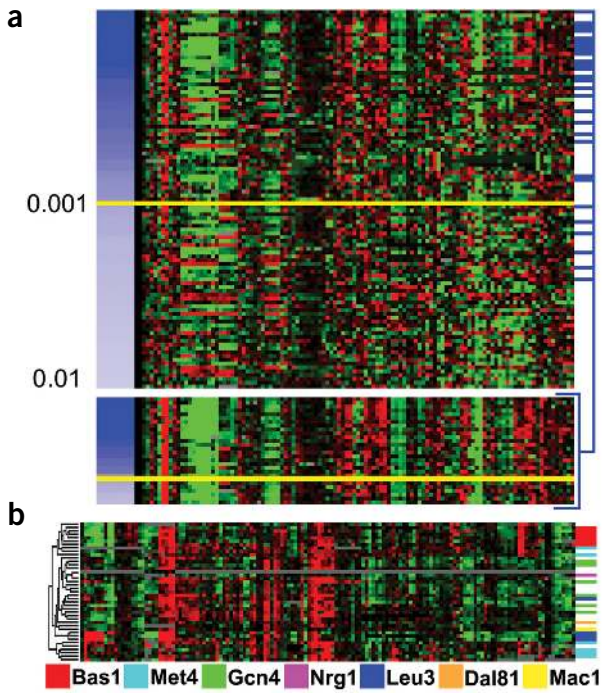

Figure 2 The GRAM algorithm integrates genome-wide binding and expression data and improves on either data source alone. (**a**) Binding data: the GRAM algorithm can improve the quality of DNA-binding information because it uses expression data to avoid a strict statistical significance threshold. Shown is DNA-binding and expression information for the 99 genes bound by the regulator Hap4 with a *P* value < 0.01 using an earlier statistical model[6]. The blue-white column on the left indicates binding *P* values, and the horizontal yellow line denotes the strict significance threshold of 0.001. As can be seen, the *P* values form a continuum and a strict threshold is unlikely to produce good results. The blue horizontal lines on the right indicate the 28 genes that were selected for modules by the GRAM algorithm. As can be seen, 22 (79%) have a *P* value < 0.001, but 6 (21%) have *P* values above this threshold. The lower portion of the figure shows together the 28 genes selected by the GRAM algorithm, and it can be seen that they exhibit coherent expression. Further, all the selected genes are involved in respiration. Six of these genes (*PET9*, *ATP16*, *KGD2*, *QCR6*, *SDH1* and *NDI1*) would not have been identified as Hap4 targets using the stringent 0.001 *P*-value threshold (*P* values range from 0.0011 to 0.0036). (**b**) Expression data: the GRAM algorithm can assign different regulators to genes with similar expression patterns that cannot be distinguished reliably using expression clustering methods alone. Hierarchical clustering of expression data was used to obtain the subtree on the left. On the right, the regulators assigned to genes by the GRAM algorithm are color coded. As can be seen, many genes with very similar expression patterns are regulated by different transcription factors.

0.001 and 0.01 in the genomic location experiments and had thus been excluded under the stringent cutoff. The GRAM algorithm identified all three as genes to which Stb1 binds without adding any additional genes that were not detected in the gene-specific chromatin-IP experiments (see **Supplementary Table 5** and **Supplementary Methods** online for full details).

We also expected that the gene modules derived by the GRAM algorithm would improve on the biological relevance of gene groupings that could be inferred from location data only. Because genes that participate in the same biological pathway often have similar expression patterns, and genes in a module share not only a common set of transcription factors but also similar expression patterns, we expected that genes in modules would be more likely to be functionally related than sets of genes identified by location data alone. Indeed, we found that gene modules derived using the GRAM algorithm were almost three times more likely to show enrichment for genes in the same MIPS functional category than were sets of genes derived solely from location data.

Similarly, we expected that genes in modules derived by the GRAM algorithm would be more likely to show independent evidence of coregulation by the regulators assigned to the module than would sets of genes obtained using location data alone. One line of evidence for such an improvement would be enrichment for specific DNA sequence motifs. We identified 34 transcriptional regulators that bind to genes in at least one module and have well-characterized DNA binding motifs in the Transcription Factor (TRANSFAC) database[11]. For each of these 34 transcriptional regulators, we constructed two lists of genes, the first using modules to which the regulator binds (generated by the GRAM algorithm) and the second using location data alone (stringent *P*-value cutoff of 0.001). We then computed from each list the percentage of genes that contained the appropriate known motif in the upstream region of DNA. We found that in most cases the percentage of genes containing the correct motif was higher when we used modules generated using the GRAM algorithm than when we used sets of genes generated from location data alone (see **Fig. 3** and **Supplementary Table 6**).

The use of a very large set of genome-wide location and expression data allowed us to validate the results of the GRAM algorithm comprehensively for the gene modules discussed above through literature searches, independent chromatin-IP experiments, and analysis for enrichment for genes in the same MIPS category and for known DNA-binding motifs. The results of this large-scale validation gave us confidence that the GRAM algorithm would be useful in analyzing new data sources. Because biological insights are often gained by examining responses to specialized treatments or environmental conditions, we were interested in exploring the performance of the GRAM algorithm on a data set that was smaller and more biologically targeted than the rich medium data. So, we chose to examine a transcriptional regulatory subnetwork involved in the response to Tor kinase signaling.

The Tor proteins are highly conserved and function as critical regulators in the response to nutrient stress[12–15]. Tor kinase signaling can be inhibited by the addition of the small macrolide rapamycin, which mimics nutrient starvation and results in a wide range of physiological responses including cytoskeleton reorganization, decreased translation initiation, decreased ribosome biogenesis, amino acid permease regulation and autophagy[16–19]. Expression analysis indicates that Tor signaling also controls transcriptional regulation of metabolic pathways involving nitrogen metabolism, glycolysis and the tricarboxylic acid (TCA) cycle[15–17].

The rapamycin response presented an ideal opportunity for applying the GRAM algorithm to the analysis of a novel transcriptional regulatory subnetwork. Previous studies suggest a specific set of regulators that are likely to function in the transcriptional response to rapamycin[15,16]. Also, several publicly available genome-wide expression data sets measuring response after rapamycin treatment are available[15,16]. More importantly, the fact that there is little information available about the transcriptional regulatory network involved and how this transcriptional network may contribute to the overall response to rapamycin treatment presented an opportunity for new biological insights.

We selected 14 transcriptional regulators that seemed likely to function in the rapamycin response in *S. cerevisiae* based on evidence from

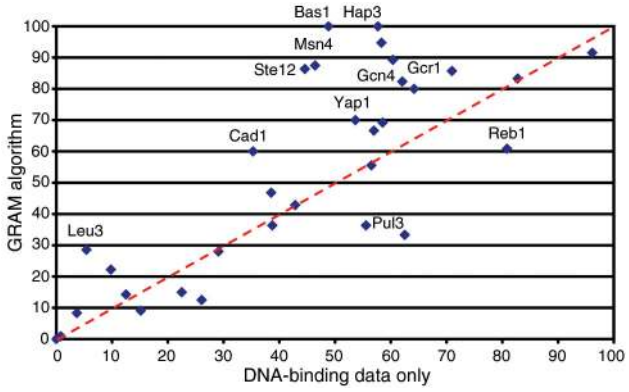

Figure 3 Motif enrichment. Genes in modules discovered by the GRAM algorithm are more likely to show independent evidence of coregulation by the regulators assigned to the module when compared to sets of genes obtained using genomic location analysis data alone, as demonstrated by an enrichment for the presence of known DNA-binding motifs. We identified 34 transcriptional regulators that bind to genes in at least one module and have well-characterized DNA binding motifs in the TRANSFAC database[11]. For each of these 34 transcriptional regulators, we generated a list of genes in modules bound by the regulator and a second list of genes bound by the regulator using location analysis data alone (stringent *P* value cutoff of 0.001). We then computed the percentage of genes from each list that contained the appropriate known motif in the upstream region of DNA. In most cases, the percentage of genes containing the correct motif was higher when we used modules generated by the GRAM algorithm than when we used sets of genes generated by location analysis data alone. See **Supplementary Table 6** online for a complete list of transcription factors analyzed.

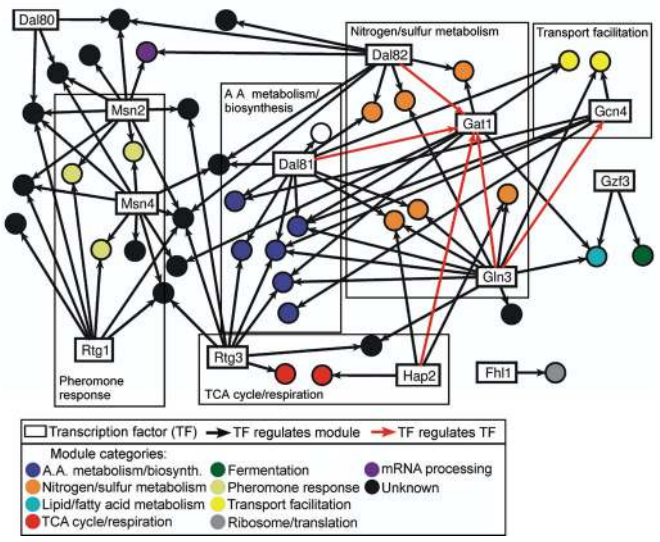

Figure 4 Rapamycin gene modules network. Analysis of the rapamycin transcriptional regulatory subnetwork revealed a number of novel biological insights, including evidence that some transcriptional regulators may control genes involved in biological pathways different from those generally associated with these regulators. Further, analysis of the network suggested more complex regulatory interactions in which there is communication among modules. Such complicated network topologies may be important for facilitating rapid and flexible responses to changing environmental conditions. See the text for further details. Thirty-nine modules containing 317 unique genes and regulated by 13 transcription factors were discovered. Red arrows between transcriptional regulators indicate that the source transcription factor binds at least one module containing the target transcription factor. Modules are colored according to the MIPS category to which a significant number of genes belong (significance test using the hypergeometric distribution $P < 0.05$).

the literature, and performed genome-wide location analysis experiments (see Methods and **Supplementary Table 7** online for full details). We ran the GRAM algorithm using the location data for the 14 transcription factors in rapamycin and 22 previously published expression experiments relevant to rapamycin conditions. We discovered 39 gene modules containing 317 unique genes and regulated by 13 transcription factors (see **Fig. 4** and **Supplementary Table 8** online). The GRAM algorithm added 192 pairs of gene-regulator interactions that would not have been identified with a strict *P* value (0.001) in the location analysis experiments. Because genome-wide binding experiments for the rapamycin regulatory network have not been performed before, it was not possible to verify these interactions comprehensively using literature searches.

As with the rich medium gene modules network, the rapamycin regulatory network discovered by the GRAM algorithm had many features that were consistent with expectations from the literature. Twenty-three of the gene modules were found to contain a significant number of genes ($P < 0.05$) belonging to a single MIPS category. There were a total of nine categories, all corresponding to biological responses associated with rapamycin treatment[12–14]. We also found that, in general, regulators were assigned to genes that reflect functions described in previously published results.

In addition to identifying established regulatory interactions, analysis of the rapamycin gene modules suggested several unexpected interactions in which regulators typically assigned to a particular biological response also appear to bind genes acting in different biological pathways. Below we give several examples of such regulatory interactions. These findings suggest models of transcriptional regulation of the rapamycin response that can be validated in further, more directed studies. A first example of an unexpected regulatory interaction involves the factors Msn2 and Msn4, which are generally regarded as stress response factors and have been well studied as activators of stress-related responses[18–21]. Unexpectedly, there were three gene modules in which Msn2 and Msn4 bound to a significant number of genes involved in the mating pheromone response pathway ($P < 0.006$). A second example involves the factor Rtg3, which is generally thought to regulate directly genes of the TCA cycle and indirectly contribute to nitrogen metabolism[22–25] (products of the TCA cycle are shunted to nitrogen metabolism pathways in low- or poor-nitrogen conditions). The gene modules network suggests that Rtg3 may directly regulate genes involved in amino acid metabolism, and more specifically in nitrogen metabolism.

A third example of an unexpected regulatory interaction involves Hap2, a part of the Hap2-Hap3-Hap4-Hap5 complex that has been well characterized as a regulator of genes involved in respiration[22,26]. Indeed, in the rich medium gene modules network, members of the Hap complex are unique among the 106 regulators profiled as the only regulators controlling modules that are significantly enriched for genes involved in respiration ($P < 0.005$). As expected, Hap2 regulates a module of respiration genes under rapamycin conditions. Unexpectedly, Hap2 was also found to regulate two modules containing genes involved in nitrogen metabolism. There is some genetic evidence for such cross-pathway regulation, as Hap2 was previously implicated as a regulator of two nitrogen metabolism genes[27,28]. Our results indicate that Hap2 participates in cross-pathway regulation more extensively than previously reported.

In addition to suggesting that some transcriptional regulators may control genes involved in biological pathways different from those

generally associated with these regulators, analysis of the gene modules network suggests more complex regulatory interactions in which there is communication among gene modules. Such complicated network topologies may be important for facilitating rapid and flexible responses to changing environmental conditions. As an example, we found that several transcriptional regulators may be involved in a feed-forward regulatory loop in which the gene encoding a regulator is bound by another regulator and both regulators bind to a set of common genes[6,29]. The regulator Gat1 has been previously identified as a general activator of nitrogen-responsive genes[30]. We found that Gat1 is itself contained in several modules along with genes involved in nitrogen metabolism. The transcriptional regulators Dal81, Dal82, Gln3 and Hap2 bind to these gene modules. Interestingly, Gat1 also binds to several gene modules along with Dal81, Dal82 and Gln3 (see **Fig. 4**). Feed-forward mechanisms may be important in regulatory responses (such as the response to rapamycin) by modulating regulatory sensitivity to sustained rather than transient inputs, providing temporal control or amplifying the transcriptional response[29]. These findings can be validated in further directed experimental studies.

The above analyses indicate that the GRAM algorithm can be useful for studying transcriptional regulatory networks using genome-wide location and expression data sources. We have made a Java implementation of the algorithm publicly available (see **Supplementary Methods** online), and believe that as new genome-wide location data become increasingly available, other researchers will find the algorithm helpful. As demonstrated, the algorithm can integrate sources of genome-wide location and expression data to help compensate for technical limitations in the data. Further, the inferred gene modules networks can give a clearer view of regulation than can either location or expression data sources alone. We have found that the algorithm is particularly useful for uncovering how certain regulators may act in multiple biological pathways. Overall, the GRAM algorithm facilitates a genome-wide approach to analysis of transcriptional regulatory networks that can suggest specific novel regulatory models, which can then be validated in more directed experimental studies.

METHODS

The GRAM (Genetic Regulatory Modules) algorithm. Below we describe the operation of the algorithm. Some details are omitted owing to space constraints; see the **Supplementary Methods** online for complete information as well as a Java implementation of the algorithm.

Let e_i denote an expression vector and b_i a vector of binding P values for gene i, where there are n_g genes. Let $B(i,t)$ denote the set of all transcription factors that bind to gene i with a P value less than t, that is, the list of indices j such that $b_{ij} < t$. Let $F \subseteq B(i,t)$ denote a subset of the transcription factors that bind to i. Let $G(F,t)$ be the set of all genes i such that for any gene $i \in G(F,t)$, $F \subseteq B(i,t)$, that is, genes to which all the factors in F bind with a given significance threshold. The algorithm begins by going over all genes, and assigning each gene i to all possible sets $G(F,t)$, where t_1 is a high-stringency binding threshold and F ranges over all subsets of $B(i,t)$.

For every set of transcription factors F, the genes in $G(F,t_1)$ serve as candidates for a module regulated by F. For each such set $G(F,t_1)$ with a sufficient number n of genes (*e.g.*, $n \geq 5$), the algorithm attempts to find a 'core' expression profile. That is, we are seeking a point c in expression space such that for an expression similarity threshold s_n, the ball centered at c of radius s_n contains as many genes in $G(F,t_1)$ as possible. Denote by $C(F,t_1,c)$ the 'core' set of genes such that $C(F,t_1,c) \subseteq G(F,t_1)$ and for each gene $i \in C(F,t_1,c)$, $d(e_i,c) < s_n$, where d is the Euclidian distance between two points. The threshold s_n is determined by using all genes, and randomly sampling subsets of size n to determine the distribution of expression distances from a subset to all genes. The problem of finding a point c for a set of expression vectors is nontrivial, and cannot be optimally solved in a reasonable time given the dimensionality of the expression space (>500). Thus, we use a theoretically motivated approximation algorithm that looks for the central point in all triplets of genes in $G(F,t_1)$ (see **Supplementary Methods** online for more details).

The genes in $C(F,t_1,c)$ are used to initialize a module $M(F)$. Conceptually, we would like to expand this module by relaxing our criteria for binding if a gene's expression profile is sufficiently similar to those in the 'core.' To do so, the algorithm calculates a combined P value p_i for each gene i that belongs to the expanded set $C(F,t_2,c)$ and does not belong to $C(F,t_1,c)$, where $t_2 > t_1$. The P value p_i is arrived at by computing independent P values for gene i and each transcription factor in F and then combining the P values using the Fisher method. A gene i from $C(F,t_2,c)$ is then included in $M(F)$ if $p_i < t_1$. This module initialization and expansion is completed for each feasible F, starting with the sets containing the largest number of factors and proceeding to the smallest. If a gene is included in a module $M(F)$, it is masked out (not considered) when forming modules with factor subsets, $M(F')$ where $F' \subseteq F$. That is, the algorithm will seek to explain a gene's expression using the most specific regulatory patterns. The thresholds $t_1 = 0.001$ and $t_2 = 0.01$ were chosen based on experiments[6] that suggested very low false positive rates for a significance threshold of 0.001. Further, the rate of false negatives was found to be relatively high for P values between 0.01 and 0.001, but decreased markedly (to <3%) thereafter.

Strains. Epitope-tagged strains were generated as described[6]. Briefly, regulators were tagged at the C terminus by using homologous recombination to insert multiple copies of the Myc epitope coding sequence into the normal chromosomal loci of these genes. Insertion of the epitope coding sequence was confirmed by PCR and expression of the epitope-tagged protein was confirmed by western blotting analysis.

Growth conditions. Strains containing epitope-tagged regulators were grown in 50 ml YPD broth (yeast extract, peptone, dextrose) at 30 °C. Cells were grown to an OD_{600} of 0.7–0.8 and rapamycin was then added to a final concentration of 100 nM. Cells were grown for 20 min at 30 °C in the presence of rapamycin.

Genome-wide location analysis. Genome-wide location analysis was done as previously described[6]. Briefly, cells containing an epitope-tagged regulator were fixed with formaldehyde (1% final concentration) and then harvested by centrifugation. Cells were lysed and then sonicated to shear DNA. DNA fragments representing chromosomal regions crosslinked to a protein of interest were enriched by immunoprecipitation with an anti-epitope antibody. After reversal of crosslinking, enriched DNA was purified. The ends of DNA fragments were then blunted using T4 DNA polymerase and ligated to previously prepared linkers. The enriched DNA was then amplified and labeled with a fluorescent dye by ligation-mediated PCR. A sample of control DNA was similarly processed and labeled with a different fluorophore. Both IP-enriched and control DNA were then hybridized to a single DNA microarray. For each factor, three independently grown cell cultures were processed and scanned to generate binding information as previously described (see **Supplementary Materials** online for complete binding data for the rapamycin experiments).

URL. The latest version of the Java implementation of the GRAM algorithm may be obtained from the authors' website at http://psrg.lcs.mit.edu/GRAM/Index.html.

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

ACKNOWLEDGMENTS

Z.B-J. is supported by the Program in Mathematics and Molecular Biology at Florida State University through the Burroughs Wellcome Fund Interfaces Program. G.G. is supported by a National Defense Engineering and Science graduate fellowship. This work was partially funded by a US National Institutes of Health grant.

COMPETING INTERESTS STATEMENT

The authors declare that they have no competing financial interests.

Received 9 June; accepted 5 August 2003
Published online at http://www.nature.com/naturebiotechnology/

1. 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).

2. Segal, E. *et al.* Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.* **34**, 166–176 (2003).
3. Ihmels, J. *et al.* Revealing modular organization in the yeast transcriptional network. *Nat. Genet.* **31**, 370–377 (2002).
4. Pilpel, Y., Sudarsanam, P. & Church, G.M. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat. Genet.* **29**, 153–159 (2001).
5. Berman, B.P. *et al.* Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the *Drosophila* genome. *Proc. Natl. Acad. Sci. USA* **99**, 757–762 (2002).
6. Lee, T.I. *et al.* Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* **298**, 799–804 (2002).
7. Mewes, H.W. *et al.* MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.* **30**, 31–34 (2002).
8. Holland, M.J., Yokoi, T., Holland, J.P., Myambo, K. & Innis, M.A. The GCR1 gene encodes a positive transcriptional regulator of the enolase and glyceraldehyde-3-phosphate dehydrogenase gene families in *Saccharomyces cerevisiae*. *Mol. Cell Biol.* **7**, 813–820 (1987).
9. Baker, H.V. Glycolytic gene expression in *Saccharomyces cerevisiae*: nucleotide sequence of GCR1, null mutants, and evidence for expression. *Mol. Cell Biol.* **6**, 3774–3784 (1986).
10. 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).
11. Matys, V. *et al.* TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res.* **31**, 374–378 (2003).
12. Jacinto, E. & Hall, M.N. Tor signalling in bugs, brain and brawn. *Nat. Rev. Mol. Cell Biol.* **4**, 117–126 (2003).
13. Crespo, J.L. & Hall, M.N. Elucidating TOR signaling and rapamycin action: lessons from *Saccharomyces cerevisiae*. *Microbiol. Mol. Biol. Rev.* **66**, 579–591 (2002).
14. Raught, B., Gingras, A.C. & Sonenberg, N. The target of rapamycin (TOR) proteins. *Proc. Natl. Acad. Sci. USA* **98**, 7037–7044 (2001).
15. Hardwick, J.S., Kuruvilla, F.G., Tong, J.K., Shamji, A.F. & Schreiber, S.L. Rapamycin-modulated transcription defines the subset of nutrient-sensitive signaling pathways directly controlled by the Tor proteins. *Proc. Natl. Acad. Sci. USA* **96**, 14866–14870 (1999).
16. Shamji, A.F., Kuruvilla, F.G. & Schreiber, S.L. Partitioning the transcriptional program induced by rapamycin among the effectors of the Tor proteins. *Curr. Biol.* **10**, 1574–1581 (2000).
17. Cardenas, M.E., Cutler, N.S., Lorenz, M.C., Di Como, C.J. & Heitman, J. The TOR signaling cascade regulates gene expression in response to nutrients. *Genes Dev.* **13**, 3271–3279 (1999).
18. Hasan, R. *et al.* The control of the yeast H_2O_2 response by the Msn2/4 transcription factors. *Mol. Microbiol.* **45**, 233–241 (2002).
19. Rep, M., Krantz, M., Thevelein, J.M. & Hohmann, S. The transcriptional response of *Saccharomyces cerevisiae* to osmotic shock. Hot1p and Msn2p/Msn4p are required for the induction of subsets of high osmolarity glycerol pathway-dependent genes. *J. Biol. Chem.* **275**, 8290–8300 (2000).
20. 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).
21. Martinez-Pastor, M.T. *et al.* The *Saccharomyces cerevisiae* zinc finger proteins Msn2p and Msn4p are required for transcriptional induction through the stress response element (STRE). *EMBO J.* **15**, 2227–2235 (1996).
22. Schuller, H.J. Transcriptional control of nonfermentative metabolism in the yeast *Saccharomyces cerevisiae*. *Curr. Genet.* **43**, 139–160 (2003).
23. Crespo, J.L., Powers, T., Fowler, B. & Hall, M.N. The TOR-controlled transcription activators GLN3, RTG1, and RTG3 are regulated in response to intracellular levels of glutamine. *Proc. Natl. Acad. Sci. USA* **99**, 6784–6789 (2002).
24. Komeili, A., Wedaman, K.P., O'Shea, E.K. & Powers, T. Mechanism of metabolic control: target of rapamycin signaling links nitrogen quality to the activity of the Rtg1 and Rtg3 transcription factors. *J. Cell Biol.* **151**, 863–878 (2000).
25. Liao, X. & Butow, R.A. RTG1 and RTG2: two yeast genes required for a novel path of communication from mitochondria to the nucleus. *Cell* **72**, 61–71 (1993).
26. Pinkham, J.L. & Guarente, L. Cloning and molecular analysis of the HAP2 locus: a global regulator of respiratory genes in *Saccharomyces cerevisiae*. *Mol. Cell Biol.* **5**, 3410–3416 (1985).
27. Dang, V.D., Bohn, C., Bolotin-Fukuhara, M. & Daignan-Fornier, B. The CCAAT box-binding factor stimulates ammonium assimilation in *Saccharomyces cerevisiae*, defining a new cross-pathway regulation between nitrogen and carbon metabolisms. *J. Bacteriol.* **178**, 1842–1849 (1996).
28. Dang, V.D., Valens, M., Bolotin-Fukuhara, M. & Daignan-Fornier, B. Cloning of the ASN1 and ASN2 genes encoding asparagine synthetases in *Saccharomyces cerevisiae*: differential regulation by the CCAAT-box-binding factor. *Mol. Microbiol.* **22**, 681–692 (1996).
29. 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).
30. Coffman, J.A., Rai, R., Cunningham, T., Svetlov, V. & Cooper, T.G. Gat1p, a GATA family protein whose production is sensitive to nitrogen catabolite repression, participates in transcriptional activation of nitrogen-catabolic genes in *Saccharomyces cerevisiae*. *Mol. Cell Biol.* **16**, 847–858 (1996).

Computational discovery of gene modules and regulatory networks

Ziv Bar-Joseph, Georg K Gerber, Tong Ihn Lee, Nicola J Rinaldi, Jane Y Yoo, François Robert, D Benjamin Gordon, Ernest Fraenkel, Tommi S Jaakkola, Richard A Young & David K Gifford
***Nature Biotechnology*; doi:10.1038/nbt890**

In the version of this article initially published online, the word "and" was omitted from the fourth sentence of the abstract, altering the meaning. The sentence should read: "We use the GRAM algorithm to describe a genome-wide regulatory network in *Saccharomyces cerevisiae* using binding information for 106 transcription factors profiled in rich medium conditions and data from over 500 expression experiments." This mistake has been corrected for the HTML and print versions of the article.