

Unsupervised learning of transcriptional regulatory networks via latent tree graphical models

Anthony Gitter^{1,2,†,*}, Furong Huang^{3,*}, Ragupathyraj Valluvan³, Ernest Fraenkel^{2,+}, Animashree Anandkumar^{3,+}

¹Microsoft Research, Cambridge, MA, USA

²Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

³Department of Electrical Engineering and Computer Science, University of California Irvine, Irvine, CA, USA

[†]Current address: Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, Madison, WI, USA and Morgridge Institute for Research, Madison, WI, USA

Email: Anthony Gitter - gitter@biostat.wisc.edu; Furong Huang - furongh@uci.edu; Ragupathyraj Valluvan - rvalluva@uci.edu; Ernest Fraenkel - fraenkel-admin@mit.edu; Animashree Anandkumar - a.anandkumar@uci.edu;

* Joint first author

+ Joint corresponding author

Abstract

Gene expression is a readily-observed quantification of transcriptional activity and cellular state that enables the recovery of the relationships between regulators and their target genes. Reconstructing transcriptional regulatory networks from gene expression data is a problem that has attracted much attention, but previous work often makes the simplifying (but unrealistic) assumption that regulator activity is represented by mRNA levels. We use a latent tree graphical model to analyze gene expression without relying on transcription factor expression as a proxy for regulator activity. The latent tree model is a type of Markov random field that includes both observed gene variables and latent (hidden) variables, which factorize on a Markov tree. Through efficient unsupervised learning approaches, we determine which groups of genes are co-regulated by hidden regulators and the activity levels of those regulators. Post-processing annotates many of these discovered latent variables as specific transcription factors or groups of transcription factors. Other latent variables do not necessarily represent physical regulators but instead reveal hidden structure in the gene expression such as shared biological function. We apply the latent tree graphical model to a yeast stress response dataset. In addition to novel predictions, such as condition-specific binding of the transcription factor Msn4, our model recovers many known aspects of the yeast regulatory

network. These include groups of co-regulated genes, condition-specific regulator activity, and combinatorial regulation among transcription factors. The latent tree graphical model is a general approach for analyzing gene expression data that requires no prior knowledge of which possible regulators exist, regulator activity, or where transcription factors physically bind. Consequently, it is promising for studying expression datasets in species and conditions where these types of information are not available or not reliable.

Introduction

Genome-wide studies of gene expression continue to be a widely-used technique for investigating biological processes and systems. Microarrays have enabled the collection of large gene expression datasets for well over a decade [1], and steady advances in experimental technologies, most notably RNA sequencing, have brought corresponding improvements in the quality of such datasets. Exploring which genes are activated or repressed in specific cell types or biological conditions can serve as a starting point for understanding gene function [2]. Beyond individual genes, groups of genes that behave similarly across diverse conditions can lead to the discovery of common transcriptional regulatory mechanisms, which provides further insight into the cellular reaction to changing conditions [3]. Transcription factors (TFs) are central regulatory proteins that bind the promoter regions of their target genes and control the expression of those genes. Genes regulated by the same TF have expression patterns that are correlated with the regulatory activity of that TF. TFs are themselves regulated, both transcriptionally and by several additional mechanisms. Post-transcriptional regulation, such as microRNA binding [4], decouples mRNA and protein expression levels. Even after a protein is translated, post-translational modifications can activate or deactivate a TF, and TFs must be localized to the nucleus in order to regulate their target genes. Consequently, a TF can be active in a particular condition without being differentially expressed and vice versa [5].

Computational strategies for recovering transcriptional regulatory networks have a long, rich history and have been extensively reviewed [6–9]. Most existing methods for inferring regulatory networks or modules require a map of TF-gene interactions [5, 10–13], depend on gene perturbations [14–16], or assume that the mRNA expression levels of the gene that encodes a TF are representative of the TF’s regulatory activity [3, 17–20]. Although this expression assumption is expedient, it is not accurate.

A preferable approach is to include the many other regulatory processes that act upon TFs in the

computational model as hidden, latent effects. We propose learning a latent tree probabilistic graphical model as an efficient approach for recovering the transcriptional regulators from gene co-expression data without relying on TF expression. Probabilistic graphical models represent probability distributions that factor according to a certain graph, termed the Markov graph [21]. Latent tree graphical models involve observed and hidden variables that factorize according to a tree model. In scenarios where hidden factors affect observed phenomena, latent tree models are capable of recovering intrinsic relationships between the observed phenomena and the latent factors. In addition, they provide a flexible approach for modeling hierarchical dependencies found in gene regulation. Although general graphical models with cycles are NP-hard to learn [22], there exist efficient guaranteed approaches for learning latent tree models. In addition, inferring the values of the latent variables given the gene expression levels is NP-hard in general but becomes computationally tractable on a tree. Moreover, biological datasets suffer from high dimensionality. There are far fewer observed samples than unknown parameters in general graphical models, and learning general models is thus ill-posed in the high-dimensional regime. Latent tree models, on the other hand, can be learned efficiently using far fewer samples than the number of nodes in the model.

In this paper, we employ the approach of Choi et al. [23] for learning a latent tree graphical model. This approach is guaranteed to recover the correct underlying tree when samples are drawn from a latent tree graphical model. Moreover, the algorithm is unsupervised and does not require knowledge of the tree structure or the number and location of the hidden variables. In addition, the approach has the flexibility to trade off the number of latent variables discovered (i.e., model complexity) with fidelity to the observed data. This latent tree learning algorithm has been successfully employed for automatic categorization of financial and text data [23], contextual object recognition [24], human pose estimation [25], and tracking dynamic social networks [26]. Alternative latent variable models such as latent Bayesian networks [27] or topic admixture models [28] are in general challenging to learn. Although some papers have shown promise [28,29], their applicability to the biological domain, which is highly noisy and data poor (in terms of the number of experimental samples per gene), is unclear and left for future investigation.

In our biological setting, we use the latent tree graphical model to represent the relationships among gene expression (represented as observed variables), expression regulators (represented as hidden variables), and other unobserved factors, where the Markov tree structure represents the hierarchical relationships. Our goals include learning which regulators and other hidden factors may control specific genes in different conditions, detecting groups of co-regulated genes (modules), and inferring the functional activity of regulators. We do

not assume that regulators are known *a priori* or that regulator activity is observed in the gene expression data. We instead recover the hidden relationships between gene expression levels, find factors that regulate different subsets of the genes, and interpret these latent variables.

We find that most latent variables correspond to specific transcriptional regulators or groups of regulators that drive gene co-expression. TFs can be mapped to latent variables as a post-processing step and *de novo* motif discovery can provide information about potential physical regulators when TF-gene interactions are unknown. Inference in the graphical model recovers the activity levels of these TFs in the various biological conditions that were surveyed. Latent variables that do not match TFs may represent additional unobserved influences on gene expression including environmental factors and higher-order biological processes. These unobserved nodes are akin to industrial sectors discovered when analyzing financial data [23], article topics found by analysis of text [23], or subjects in image analysis (Figure 1). Moreover, the neighbors of a given variable in the Markov tree structure are conditionally independent given the variable. In the biological context, this can be interpreted as the conditional independence of the expression levels of a group of genes that are neighbors of (i.e., controlled by) a common regulator or group of regulators. These edges between a latent variable and a set of genes are potentially more informative than gene-gene edges for understanding regulatory processes because they guide the search for explanations of why groups of genes are correlated.

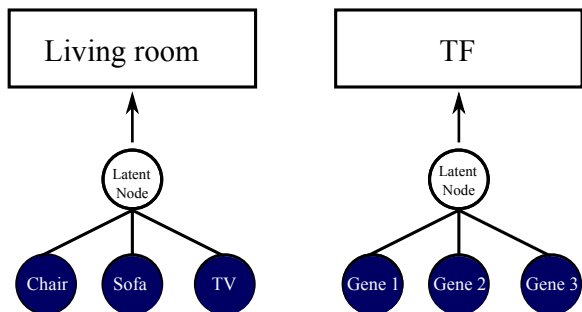


Figure 1: The latent tree algorithm we use to construct transcriptional regulatory networks is analogous to the latent tree approach for discovering objects in images. In image analysis, co-occurrence of objects in images can be explained by their common dependence on an unobserved ‘meta object’ that is not explicitly labeled in training images. For instance, the labeled objects ‘Chair’, ‘Sofa’, and ‘TV’ are related to the common subject ‘Living room’, which is latent. Similarly, a latent tree can be used to find groups of genes that are co-expressed across biological conditions due to co-regulation by an unobserved regulator. The latent tree does not directly provide the regulator-gene interactions, but we demonstrate how external information can be used to reveal the identities of these regulators as specific transcription factors or groups of transcription factors.

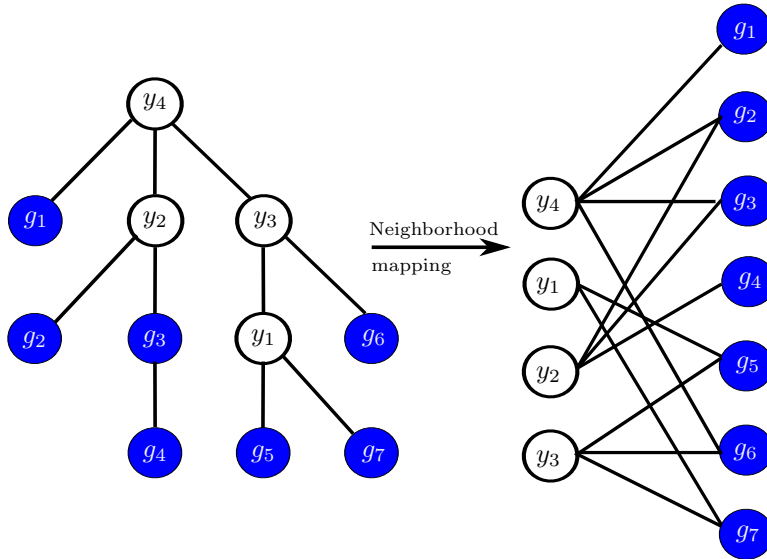
We applied our latent tree graphical model to a compendium of yeast gene expression data covering

various stress and non-stress conditions [30]. By studying yeast we can leverage comprehensive TF binding data [31] to annotate the latent nodes (LNs) and demonstrate how the latent tree recovers well-known yeast stress response mechanisms. For other organisms in which TF binding is poorly characterized, we show how *de novo* motif discovery can be used to identify specific regulators that correspond to the latent nodes. In addition, the latent tree predicts pairs of TFs that exhibit combinatorial regulation in specific stress conditions, and these predictions are supported by independent data. To highlight the advantages of not assuming gene expression is a reliable proxy for TF activity, we compared our results with ARACNE [17,32]. Due to its dependence on TF expression, ARACNE recovers only a small subset of the important regulators in the biological conditions we study. Because the latent tree approach does not require any input data besides gene expression, it is quite general and can be used to better understand regulatory relationships in many biological settings.

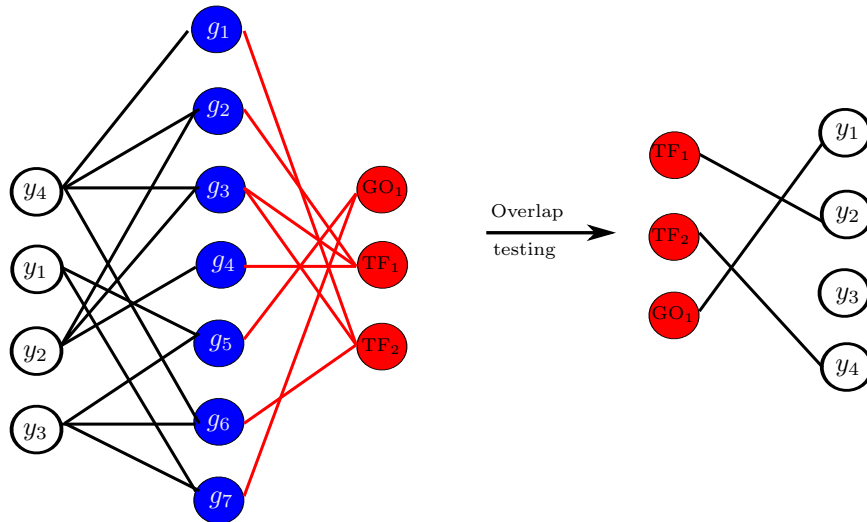
Results

We used the latent tree approach to model yeast gene expression levels. An example of a latent tree is shown in Figure 2a. A variable in the latent tree is conditionally independent of all other variables when conditioned on its immediate neighbors. Thus, the latent tree model provides useful conditional independence relationships between genes' observed expression levels and the discovered latent variables.

Our conjecture is that the latent effects that are captured by the latent variables primarily represent the activities of transcription factors. Latent nodes are introduced naturally by our unsupervised learning method without any knowledge of the TFs. Figure 2b illustrates how we can post-process the latent tree to determine which latent nodes represent known TFs and which are potentially novel regulators. Specifically, after learning the latent tree, we conduct Fisher's exact test (controlling for multiple hypothesis testing) to find statistically significant relationships between the genes bound by transcription factors and the genes that are correlated with latent nodes. This general framework allows us to interpret the latent nodes in terms of their relationships with TFs. We also demonstrate how to annotate latent nodes when TF-gene interactions are unavailable using motif discovery. Furthermore, we use the TF-latent node mappings to detect potential combinatorial regulation among TFs that are predicted to control the same group of genes. Latent nodes that do not correspond to specific transcriptional regulators are shown to instead represent biological processes in some cases.



(a) Latent tree



(b) TF and GO category annotation

Figure 2: An overview of the latent tree approach. (a) We learn a latent tree (left) from the gene expression data, which introduces latent nodes. g_i denotes an observed gene variable and y_j denotes a latent node. For each latent node, we define an extended neighborhood of influence that includes all genes that are highly correlated with the latent node activity (right). The latent node neighborhood may include both direct neighbors in the tree and more distant gene variables, and genes may be included in multiple neighborhoods. For example, gene g_5 is not directly connected to latent node y_3 in the latent tree structure (left), but it is influenced by both y_1 and y_3 (right). (b) We evaluate the groups of genes influenced by each latent node by annotating the LNs that likely represent specific TFs or Gene Ontology (GO) terms. We determine which latent nodes may correspond to TFs by assessing the overlap of the LN neighborhoods, the black edges, and the gene targets that are bound by a TF, the red edges (left). Likewise, we search for LN neighborhoods that are enriched for particular GO terms. Finally, we obtain a bipartite graph representing statistically significant LN-TF pairs and LN-GO term pairs (right). Note that we do not use TF binding interactions or GO annotations when learning the latent tree structure in (a).

Inference on the latent tree reveals the activity levels (conditional means) of the latent nodes without relying on external TF-gene binding information. TF activity cannot be directly observed from gene expression data so the inferred values provide a powerful way to detect TFs' context-specific regulatory behaviors.

Modeling Yeast Stress Response

We applied our latent tree algorithm to a compendium of *Saccharomyces cerevisiae* microarray experiments composed of many stress conditions (hyperosmotic stress, heat shock, DNA damage, amino acid starvation, etc.) and normal growth conditions [30]. Transcriptional regulation in yeast stress response has been studied extensively, revealing the primary TFs that drive transcriptional changes, which allows us to confirm many of our predictions. Groups of genes that exhibit similar expression profiles across different stress conditions are in many cases controlled by common TFs, and latent tree analysis of these co-expressed genes can guide the search for novel stress-specific TF activity.

Latent Tree

To model unobserved transcriptional processes, we first learn a latent tree network and then determine which genes are likely to be controlled by the hidden regulators in the tree. We construct a latent tree using 1035 genes that exhibit substantial expression changes in the yeast stress response data as the observed variables. Specifically, we include all genes that have high covariance with at least one other gene. The latent tree algorithm automatically determines the number of latent nodes using the Bayesian Information Criterion. However, to improve the biological interpretability of the model and minimize redundant latent nodes, we include a post-processing contraction step that merges latent variables that have a small information distance to an observed gene variable. Each latent node should reflect a unique biological activity signature, which may be similar to the signatures of other latent nodes but should not be nearly identical. In our yeast study, we set the contraction parameter using prior knowledge (Methods) to obtain a latent tree with 90 latent nodes (Figure 3a and Supplementary Table 1).

Supplementary Figure 1 shows the degree distribution of the LNs. Fifty LNs have only three direct neighbors, the minimum possible degree of a LN in the latent tree model (Methods). One hub LN (node 85) has a degree of 23 and is connected to many sporulation genes. The average LN degree is 4.2, but the

influence of the LNs extends beyond their immediate neighbors. When a group of genes are highly correlated with a LN, only a subset of them are selected by our algorithm as the direct neighbors of that LN due to the tree constraint on the network structure. Nevertheless, the LN may control all of these genes and influence their co-expression (Figure 2a).

In order to guide our biological interpretation of the latent tree model, we define an extended neighborhood of influence for each LN (also referred to as the LN neighborhood), which extends beyond its immediate neighbors in the tree (Supplementary Table 2). The LN neighborhood includes all genes in the latent tree network that are highly correlated with the LN, and each gene may belong to multiple LN neighborhoods. For instance, latent node 37, which is associated with osmotic stress response and the TFs Msn2 and Hsf1 (as determined below), is directly connected to only three genes in the latent tree (Figure 3b) but is likely to regulate all 64 genes in its extended neighborhood of influence (Figure 4).

Recall that the latent tree approach does not use TF-gene binding interactions because our learning approach is unsupervised. However, in order to evaluate our model, we can leverage existing TF-gene binding interactions and annotate which LNs may represent specific TFs or groups of TFs. We employ a high-confidence yeast TF binding dataset [31] to carry out this annotation. If a LN neighborhood is significantly enriched for genes bound by a specific TF, then it is likely that the LN signature represents the activity levels of that TF across the biological conditions. For each LN-TF pair, we compare the LN neighborhood with the set of genes bound by the TF and apply Fisher’s exact test to assess the overlap (Figure 2b). Raw p -values are calculated for each null hypothesis $\mathcal{H}_{0,ij}$ that the pair LN_i and TF_j are statistically independent. We use the false discovery rate (FDR) to control for multiple hypothesis testing (Supplementary Methods) and consider a TF and LN to be associated if the FDR is less than 0.05 (Supplementary Table 3). The associated TF-LN pairs can be represented as a bipartite graph (Figure 5). The significant pairs provide a many-to-many mapping between the TFs and LNs and reveal how specific TFs may induce the correlation structure observed in the gene expression data.

In our experiments, we find that the majority of the LNs (51 out of 90) are annotated with at least one TF. On the other hand, only 28 TFs match at least one LN. Many of the samples in the expression dataset are from stress conditions, and we filter genes that do not vary substantially across the conditions. These genes perform other biological functions, and the TFs that primarily bind the removed genes are not expected to appear as predicted regulators in our latent tree analysis. TFs that predominantly bind the stress response

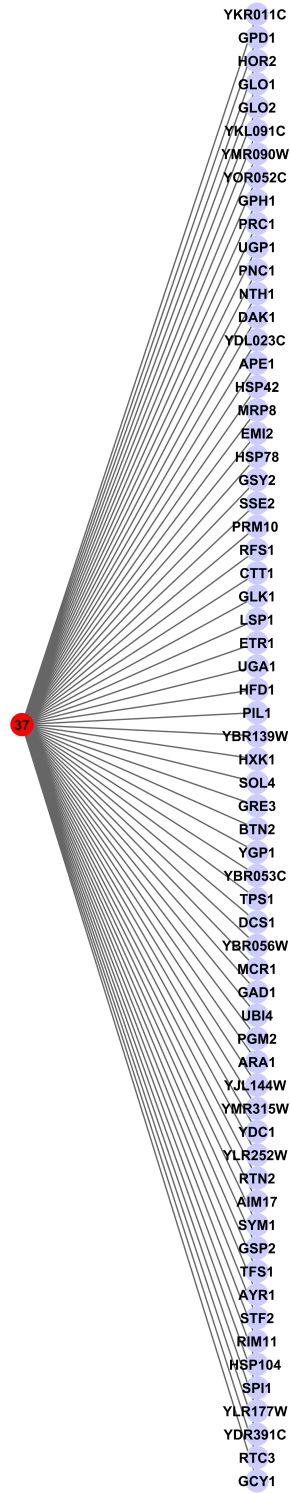


Figure 4: The extended neighborhood of latent node 37, which contains all of the yellow nodes in Figure 3b.

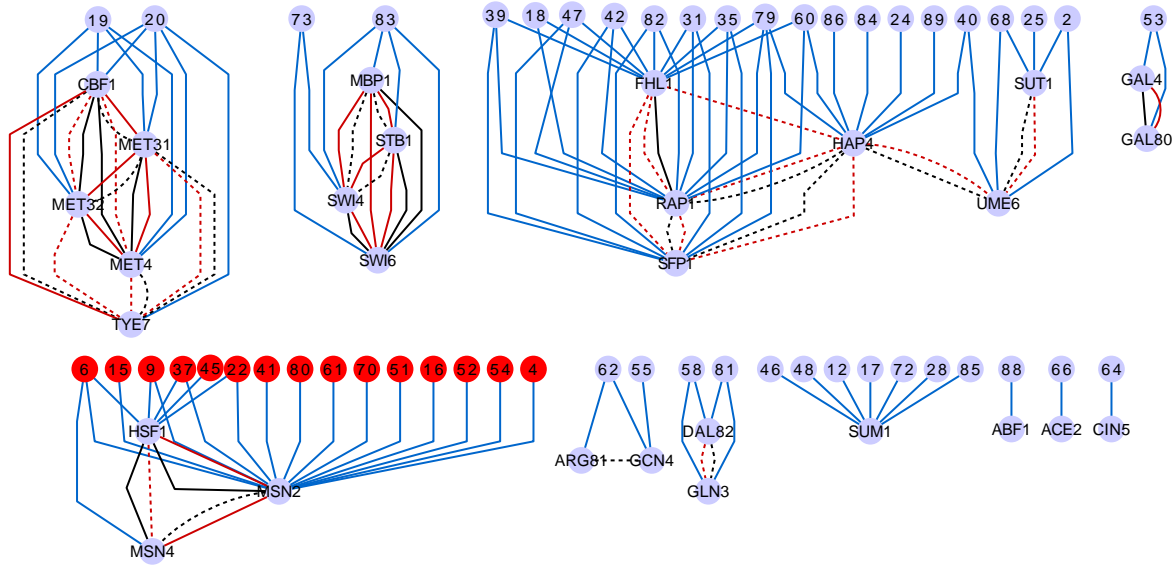


Figure 5: Bipartite graph between TFs and LNs (blue edges). Latent variables are the numbered nodes, where the number is an arbitrarily assigned index. TF-TF edges are overlaid on the bipartite TF-LN graph and represent known associations between pairs of TFs that are assigned to the same latent node. Black edges are physical interactions and red edges are genetic interactions. Solid edges are direct TF-TF interactions and dashed edges signify the TFs interact through an intermediate protein. Red nodes are the latent nodes that are active in osmotic stress response. Other latent nodes that are active in this stress condition but not associated with any TFs are omitted.

genes that are included in our latent tree model and are known stress regulators are prevalent among the significant TF-LN pairs, affirming our model’s ability to correctly identify regulators of co-expressed genes. Examples include Gcn4 in amino acid starvation [33], Hsf1 in heat shock [34], Sfp1 in DNA damage [35], and Msn2 and Msn4 as general stress regulators [36]. Other components of the TF-LN bipartite graph capture non-stress biological processes such as the cell cycle (Mbp1, Stb1, Swi4, and Swi6) [37] and ribosomal protein transcription (Fhl1 and Rap1) [38] because the genes functioning in these processes are co-expressed in a sufficient number of the samples. Latent node 85, the hub in the latent tree that is connected to many sporulation genes, is associated with Sum1, a known sporulation gene repressor [39].

Not all latent nodes correspond to transcription factor activity. In some cases, a latent node can capture other hidden external effects or represent an entire biological process. We explore whether the latent

node neighborhoods that do not significantly overlap any TFs could relate to biological processes by assessing whether they are enriched for Gene Ontology (GO) terms [40] (Table 1 and Supplementary Table 4). Although these latent nodes may not correspond to specific regulatory mechanisms, they remove direct dependence among the neighboring genes because the latent tree model posits that the genes are conditionally independent given the state of the latent process. Latent node 1 is one such example. Its gene neighborhood significantly overlaps with genes annotated with the GO term ‘regulation of cell cycle’, but these genes do not appear to be under the direct control of the cell cycle TFs Mbp1, Stb1, Swi4, or Swi6 (Figure 5). This reveals the presence of an additional hidden effect besides the activities of these four TFs that explains dependencies among a subset of cell cycle genes. In other cases, the GO terms are associated with latent nodes that do correspond to specific TFs and complement the known functions of those TFs. For example, all seven latent nodes associated with Sum1 overlap with the ‘sporulation’ GO term.

Table 1: Statistically significant overlaps between GO biological process terms and latent nodes.

| GO biological process | Latent nodes |
|---|---|
| carbohydrate metabolic process (GO:0005975) | 4 37 49 52 61 70 78 |
| cell wall organization or biogenesis (GO:0071554) | 12 17 28 46 48 72 85 |
| cellular amino acid metabolic process (GO:0006520) | 20 |
| cellular respiration (GO:0045333) | 24 39 40 60 84 86 89 |
| cytoplasmic translation (GO:0002181) | 18 31 35 39 42 47 60 79 82 |
| DNA recombination (GO:0006310) | 19 90 |
| generation of precursor metabolites and energy (GO:0006091) | 24 40 60 68 84 86 89 |
| mitochondrion organization (GO:0007005) | 89 |
| nuclear transport (GO:0051169) | 47 82 |
| oligosaccharide metabolic process (GO:0009311) | 4 52 61 70 |
| organelle assembly (GO:0070925) | 31 34 35 47 82 |
| protein folding (GO:0006457) | 45 |
| regulation of cell cycle (GO:0051726) | 1 |
| ribosomal large subunit biogenesis (GO:0042273) | 3 5 13 30 32 34 47 57 77 82 |
| ribosomal small subunit biogenesis (GO:0042274) | 3 5 13 32 34 42 47 57 77 82 |
| ribosomal subunit export from nucleus (GO:0000054) | 13 34 47 82 |
| ribosome assembly (GO:0042255) | 34 47 82 |
| RNA modification (GO:0009451) | 5 13 32 34 47 57 77 82 |
| rRNA processing (GO:0006364) | 3 5 13 14 30 32 34 42 47 50 57 77 82 88 |
| sporulation (GO:0043934) | 12 17 28 46 48 72 85 90 |
| transcription from RNA polymerase I promoter (GO:0006360) | 34 |
| translational elongation (GO:0006414) | 31 35 |

Latent Variable Conditional Mean Estimation

Because many of the latent nodes are associated with the same TFs, we examine how their activities vary across the biological conditions to establish unique roles for latent nodes that initially appear to be redundant. We employ Gaussian belief propagation, which is computationally efficient in the latent tree model, to infer the conditional means for LNs under all the 498 conditions present in the dataset (Figure 6a). The conditional means can be thought of as the signed activity of a latent node in each condition. The associations between TFs and latent nodes can be used to transfer the latent node activities to the corresponding TFs. For example, Fhl1 significantly overlaps with nine latent nodes (Figure 6b), and the activity levels of these latent nodes are shown in Figure 6c. Although there is a common general trend in the activity profiles of all nine latent nodes — negative activity in conditions to the left and positive activity in conditions to the right — there are obvious differences in the conditional means of these latent nodes as well. Figure 6c highlights one group of conditions where this is particularly evident, which is shown in greater detail in Figure 6d. In these conditions, which are primarily late time points in nitrogen depletion and stationary phase, four Fhl1-associated latent nodes show positive activity and the other five show little or negative activity. Although the latent nodes appear similar because their gene neighborhoods all overlap with Fhl1-bound genes, they in fact capture different aspects of Fhl1 regulation.

Combinatorial Regulation

Transcription factors do not act in isolation, but rather participate in complex interactions with other TFs such as cooperative regulation [41], competitive DNA binding [42], and functional redundancy [43]. Our mapping between latent nodes and TFs can suggest such combinatorial relationships among TFs by revealing groups of TFs that are associated with the same LN. These groups of TFs potentially work together or in competition to regulate similar genes in specific stress conditions and allow us to selectively focus on 30 pairs of TFs that are likely to function jointly from among all 4560 TF-TF pairs.

Our filtering strategy effectively identifies pairs of TFs having the capacity to operate jointly. We used BioGRID [44] to confirm whether there are known relationships between the TFs we predict to function cooperatively or competitively. Many of these TF pairs interact physically, supporting their putative combinatorial regulation (Figure 5, black edges). For instance, we predict a relationship between Gal4 and Gal80, which are both associated with latent node 53. Gal80 has been shown to bind to Gal4’s activation domain

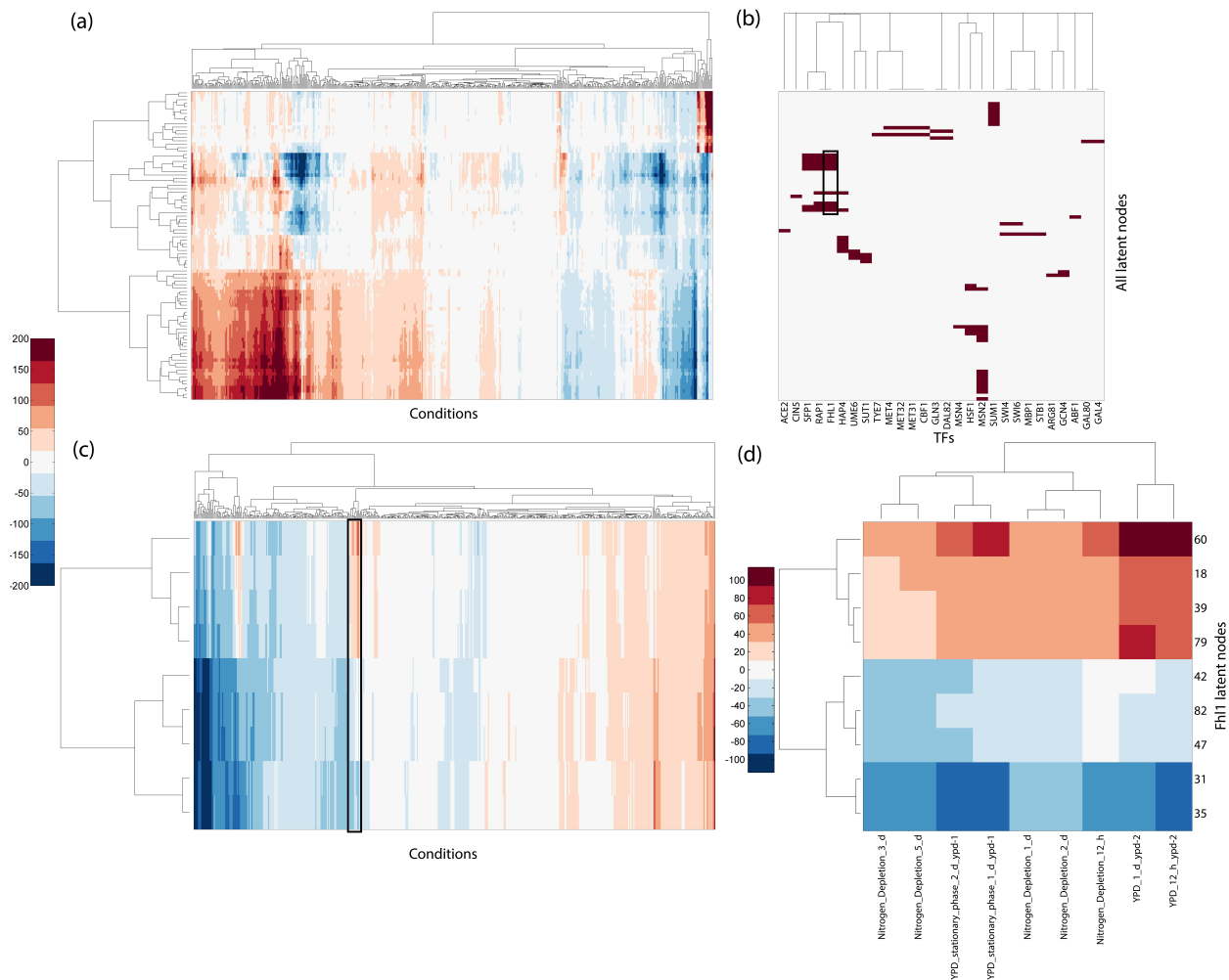


Figure 6: (a) A hierarchical clustering of the latent node activities and biological conditions. Red denotes positive regulatory influence and blue reflects inhibition. (b) The mapping of TFs and latent nodes (as in Figure 5) reveals TF activity across the conditions. The latent node activities can be transferred to the corresponding TFs. The black box highlights latent nodes mapped to Fhl1. (c) The activity levels of the nine latent nodes that map to Fhl1. Fhl1 shows distinct modes of regulation. It functions primarily as a repressor in conditions to the left and as an activator in conditions to the right. Columns do not appear in the same order as in (a). In some conditions, such as those outlined in the black box, the different latent nodes mapped to Fhl1 capture distinct activity levels. (d) The highlighted conditions from (c), which are mostly nitrogen depletion and stationary phase, demonstrate how latent nodes that correspond to the same TF can represent unique activity profiles. Rows are in the same order as in (c).

in a galactose-dependent manner, which inhibits Gal4’s transcriptional activity [45]. Other TF pairs exhibit various types of genetic interactions (Figure 5, red edges), which can indicate cooperative or redundant relationships depending on the specific type of genetic interaction. Furthermore, when we consider indirect physical and genetic interactions in which the two TFs do not directly interact but interact with a common third protein, we find that all of the TF-TF pairs we identified have some form of known physical or functional relationship (Figure 5, dashed edges).

To further probe TF-TF relationships, we study the DNA binding motifs [31] of TFs that are associated with the same LN. Some groups of TFs, such as Arg81-Gcn4 and Met4-Met31-Met32, have very similar binding motifs suggesting the presence of a multi-TF regulatory mechanism. The exploration of the exact nature of that relationship (competitive, redundant, or cooperative) can be guided by the latent tree, which provides the specific conditions in which these TFs are expected to be active together.

Motif Enrichment of Unannotated Latent Nodes

Our previous association of TFs and latent nodes depended on TF-gene interaction data, which are not available for many species and cell types in higher-order organisms. However, even without TF-gene interactions it is still possible to annotate latent nodes. We can perform *de novo* motif discovery on the gene neighborhood of each latent node, searching for common DNA sequences in the genes’ promoter regions that may be bound by the hidden regulator. Given significant motifs, we can scan motif databases to identify TFs with a known binding motif that matches the *de novo* motifs.

We demonstrate this general strategy by searching for regulators whose activity could be represented by the 39 latent nodes that do not significantly overlap any TFs (Supplementary Table 3). The WebMOTIFS [46] motif discovery tool identifies significant motifs for 35 latent nodes. These latent node neighborhoods are enriched for one to three motifs, except latent node 43, which has ten distinct motifs. We use STAMP [47] to align the discovered motifs to known yeast TF binding motifs in the JASPAR [48] database and find strong matches for all *de novo* motifs (Supplementary Table 5).

Nine of the TFs — Cup9, Gcr1, Ime1, Ndt80, Nhp10, Rei1, Rgm1, Stb3, and Tod9 — that are the best match for a *de novo* motif are absent from the TF-gene interaction dataset [31]. The other 14 are represented in the TF binding interactions but do not exhibit significant overlaps with these latent nodes.

One explanation is that these are lower confidence TF-LN associations. Alternatively, these associations may have been missed previously because the TF-gene interaction dataset represents a limited number of experimental conditions (typically rich medium). Yeast TFs can change their binding patterns in different conditions [49] so it is possible that the genes in a latent node neighborhood are bound by a TF in the stress conditions in which the latent node is active but not in normal growth conditions. This could explain why Msn4 only annotates a single latent node when using the TF-gene binding data despite its role as a primary general stress response TF. Msn4 has been characterized as a ‘condition-altered’ regulator [49] (it binds different, but partially overlapping, groups of genes in different conditions), and it is the best matching TF for nine latent nodes in our motif discovery analysis. Thus, not only can *de novo* motif discovery be used to annotate latent trees in settings where TF-gene interactions are not available, but it can also compensate for TF associations that are missed due to the condition-specific nature of TF binding.

Comparison with ARACNE

Algorithms that only model observed gene expression variables and implicitly assume that gene expression is representative of regulator activity recover a fundamentally different type of network structure than the latent tree graphical model. To illustrate, we compare with ARACNE [17,32], a regulatory network inference algorithm that performed comparably with the best methods on the *S. cerevisiae* dataset in the DREAM5 network inference challenge [8]. Briefly, it utilizes the pairwise mutual information between genes’ expression values to learn a general network structure (in contrast with a tree) using the data processing inequality. ARACNE does not distinguish between genes and TFs. Each TF is represented by the expression levels of the gene that encodes it. Hidden nodes are not incorporated in ARACNE so it cannot recover the true conditional independence structure when latent effects are present, as they typically are in biological settings.

We run ARACNE using the same yeast gene expression data that has been filtered to remove genes that do not covary with any other genes (Methods). The ARACNE network contains only nine TFs; all other TFs did not vary substantially across the biological conditions. This reinforces the limitations of assuming that TF activity is coupled with TF expression. Our latent tree analysis (Figure 5) and prior research into the conditions targeted in the expression dataset provide strong evidence that far more than nine TFs are involved in regulating these processes. Alternative strategies for filtering the expression data and other network reconstruction algorithms could recover additional TFs, but this coarse comparison demonstrates

the distinct assumptions and goals of the latent tree model and traditional methods.

Discussion

The latent tree graphical model is a powerful unsupervised technique for recovering hidden structure in gene expression compendia and detects different types of gene expression relationships than traditional regulatory network reconstruction techniques. Our study of stress response and non-stress conditions in yeast reveals 90 latent factors that explain gene co-expression. Many of these latent nodes represent specific TFs or groups of TFs, and the latent tree model recovers the condition-specific activities of these TFs. *De novo* motif discovery reveals additional TF-latent node associations, many involving the general stress transcription factor Msn4, that were missed due to condition-altered TF binding. We further demonstrate that the latent tree’s predictions are not captured by an algorithm that assumes gene expression is representative of TF activity.

Latent variable models have been successfully applied in a variety of gene expression analyses. Surrogate Variable Analysis [50,51] accounts for unobserved confounding factors, and CellCODE [52] models variation in cell type composition during differential expression analysis. Analogous to principal component analysis (PCA), latent variable methods can recover informative low-dimensional representations of high-dimensional expression data [53,54]. Autoencoders trained on gene expression data [55,56] provide an alternative compact representation, and the hidden units may be associated with transcription factors and biological pathways or attributes such as disease status and patient survival in a clinical setting. The Latent Differential Graphical Model [57] identifies rewiring in gene regulatory networks from two conditions, and BicMix [58] can detect subsets of genes that are co-regulated only in specific conditions.

Early methods that incorporated latent variables in regulatory network modeling were more restricted than our latent tree graphical model. Bayesian networks with latent variables were applied to study the yeast galactose regulatory network but only considered a limited number of genes [59,60]. Another graphical model [61] assumed that gene expression represents TF activity and only included hidden variables to model potentially unknown TFs and confounding factors. More recently, INSPIRE [62] developed a latent variable Gaussian graphical model for a compelling study of gene expression in ovarian cancer. INSIPRE is unique in its ability to combine gene expression samples that contain different genes, but some of the underlying motivations are similar to our latent tree graphical model. INSPIRE represents gene modules with latent

variables, learns conditional dependencies among the module latent variables, and assigns each gene to exactly one module. Unlike INSPIRE, our latent tree approach allows direct gene-gene edges in the graph but is restricted to tree topologies. In the latent tree, genes may belong to multiple modules, which provides additional flexibility, but the modules are not provided by the graph structure. Rather, we must define the extended neighborhood of influence for each latent node in a post-processing step. Directly comparing INSPIRE and the latent tree graphical model is an important topic for future work.

Numerous methods cluster genes based on their expression levels. Given a cluster of genes, it is possible to find putative regulators using the same type of TF enrichment analysis we performed to annotate the latent nodes in our latent tree model. However, most clustering approaches cannot model the dependencies among genes and regulators or recover the TF activity levels across the biological conditions, which requires inference in the graphical model. The inferred regulator activities can reveal biological phenomena, such as our prediction that Fhl1-associated latent nodes can exhibit both positive and negative activity in some stress conditions (Figure 6d).

Many network reconstruction algorithms integrate gene expression and regulator binding (from DNA binding motifs, ChIP-chip, ChIP-seq, analysis of epigenetic features, or other similar sources) and derive regulator activities from the expression levels of the bound targets. These methods work well when the experimental conditions (for example, cell type and environment) of the regulator binding data match the gene expression conditions. However, when species- and condition-specific binding data are unavailable, these methods are either inapplicable or can miss important regulators, as we observed with the condition-altered binding of Msn4. When appropriate regulator binding interactions are available, they provide a natural way to interpret some of the latent nodes in the latent tree model. Even in these cases the latent tree can still provide a complementary perspective about gene expression. By first learning the latent tree structure and then annotating latent nodes in an optional post-processing step, the latent tree model can detect other unobserved factors that induce dependencies among co-expressed genes without being biased by regulator binding interactions.

In contrast to algorithms that assume TF activity is accurately represented by gene expression, our latent tree algorithm infers the activities of the hidden regulators without using TF expression. Previous studies have shown that mRNA expression is not always a reliable proxy for protein abundance or activity due to post-transcriptional regulation and other effects [13, 63–68]. The regulatory activity of a protein is a

function of its protein abundance, sub-cellular location, post-translational modifications, and other factors, which makes it difficult to observe and poorly approximated by gene expression. Consequently, algorithms that tie regulator activity to expression can miss important TFs that are not differentially expressed. Indeed, the DREAM5 network inference challenge revealed that algorithms that assume mRNA levels of TFs and their target genes are mutually dependent can successfully reconstruct *in silico* regulatory networks (where this assumption holds) but perform poorly on expression data from a eukaryotic organism like yeast (where the assumption fails) [8].

Our model’s predicted osmotic stress regulators illustrate the advantages of decoupling TF expression and regulator activity. The primary drivers of the transcriptional response to hyperosmotic stress have been well-studied computationally [69,70] and experimentally and include the TFs Hot1, Msn2, Msn4, and Sko1 [71]. We identified the LNs that are most active in the osmotic stress conditions in our expression dataset and used the TF-LN bipartite graph to recover the TFs that those LNs represent (Figure 5). Although Msn2 and Msn4 only display at most 2.2- and 1.7-fold increases in expression, respectively, across the 14 osmotic stress samples and decreases in expression of similar magnitudes, the latent tree correctly recovers them as osmotic stress regulators. Hsf1 is also predicted to be an osmotic stress regulator, most likely because some of the osmotic stress experiments were performed simultaneously with a mild heat shock [1], and Hsf1 controls transcription in heat shock response. Hot1 may be represented by a latent node, but was not annotated because it is not present in the TF binding data [31]. The Module Networks method [3], which uses regulator expression to construct condition-specific regulatory modules, predicts seven regulators that control its ‘Energy and Osmotic stress’ modules [72]. However, Module Networks fails to recover any of these four core osmotic stress TFs because their expression does not correlate strongly enough with the expression profiles of the genes that respond to this stress.

Representing the drivers of transcriptional regulation as latent variables instead of observed variables whose activity is approximated by gene expression could also guide the discovery of other classes of regulators. It is well-understood that RNA levels are not controlled by TFs alone. MicroRNAs are recognized as an important type of expression regulator [4] with relevance to human disease, and competitive RNA binding can indirectly influence mRNA expression in some cases [73]. Although the latent tree model cannot directly identify these novel mechanisms, recognizing that some latent variables do not correspond to known transcriptional regulators provides a direction for further investigation.

Our current analysis focuses on transcriptional regulation in yeast because the vast collection of previous studies allows us to verify many of the latent tree model’s predictions. Having established the utility of the latent tree approach, future work could include applying the model to provide insights into other species and human disease. Transcriptional regulation in plants is poorly understood relative to yeast, but hundreds of gene expression samples are publicly available for plant species — including crops such as barley, grape, maize, rice, soybean, tomato, and wheat — and can be used to estimate gene co-expression [74]. This allows us to study transcriptional regulation in these crops using our latent tree approach, and plant TF databases such as PlantTFDB [75] could be used to annotate the latent nodes.

In humans, efforts to profile cancer cell lines [76] and primary tumors [77] have yielded large-scale expression datasets. Despite the lack of a comprehensive map of TF binding in specific types of cancer cells, latent nodes could be annotated using the same motif discovery approach we described for yeast or by integrating predicted TF binding interactions derived from epigenetic features [78] and microRNA binding [79]. The latent variables could also generate insights into cancer phenotypes. INSPIRE demonstrated that latent variables learned with an unsupervised algorithm can be more predictive of histological and clinical phenotypes such as stroma type, patient survival, and tumor resectability than the expression levels of all genes [62]. The latent tree algorithm is not limited to gene expression data and could be used to probe the relationships between mutations, copy number variation, protein abundance, and phenotypic data available in the cancer collections as well.

Methods

Unsupervised Learning of Latent Tree Models

Learning latent tree models involves discovering structural relationships (structure learning) and estimating the strength of such relationships (parameter estimation). Many methods have been developed previously for learning the structure of latent trees [23,80,81], and we employ the method of Choi et al. [23]. Parameter estimation is carried out through the standard expectation maximization (EM) [82] procedure. Below, we introduce graphical models for Gaussian distributions and then describe the main steps of the latent tree method of Choi et al. [23].

Gaussian Graphical Models

A Gaussian graphical model is a family of jointly Gaussian distributions that factor in accordance with a given graph. Let V represent the node set and E the edge set. Given a graph $\mathcal{G} = (V, E)$, we consider a vector of Gaussian random variables $X = [X_1, X_2, \dots, X_p]^T$ where each node $i \in V$ is associated with a scalar Gaussian random variable X_i and $p = |V|$. A Gaussian graphical model Markov on \mathcal{G} has a probability density function (pdf) that is parameterized as

$$f_X(x) \propto \exp \left[\frac{1}{2} x^T J_{\mathcal{G}} x + h^T x \right], \quad (1)$$

where $J_{\mathcal{G}}$ is a positive-definite symmetric matrix whose sparsity pattern corresponds to that of the graph \mathcal{G} . More precisely, $J_{\mathcal{G}}(i, j) = 0 \iff (i, j) \notin E$. The matrix $J_{\mathcal{G}}$ is known as the potential or information matrix with the non-zero entries $J(i, j)$ as the edge potentials, and the vector h is the potential vector. This form of parameterization is known as the information form and is related to the standard mean-covariance parameterization of the Gaussian distribution as $\mu = J^{-1}h$, $\Sigma = J^{-1}$, where $\mu := \mathbb{E}[X]$ is the mean vector and $\Sigma := \mathbb{E}[(X - \mu)(X - \mu)^T]$ is the covariance matrix.

Additive Metric in Tree Graphical Models

For Gaussian models, the information distance between any two nodes i and j in a tree T is defined as

$$d_{ij} := -\log |\rho_{i,j}|, \quad (2)$$

where $\rho_{i,j} := \frac{\text{Cov}(X_i, X_j)}{\sigma_{X_i} \sigma_{X_j}}$ denotes the correlation coefficient between nodes i and j . Note that $\text{Cov}(\cdot)$ is the covariance and σ is the standard deviation. These distances $\{d_{ij}\}$ can thus be estimated using the samples corresponding to the observed nodes. If the joint probability distribution is a tree-structured graphical model, then the information distances are additive along paths in the tree as

$$d_{kl} = \sum_{(i,j) \in \text{Path}(k,l)} d_{ij} \quad (3)$$

Learning a latent tree can thus be reformulated as learning a tree structure T given pairwise (estimated) distances $\mathbf{d} := \{\hat{d}_{ij} : i, j \in V\}$ between the observed nodes i and j , $\forall i, j \in \mathcal{G}$.

Sibling Grouping and Recursive Grouping

The latent tree algorithm constructs a tree in a bottom-up manner. Every node (except the root) has exactly one parent in the tree. We call a group of nodes siblings if they share the same parent. The algorithm first classifies nodes that are under consideration at the current iteration into sibling or parent-child groups [23]. A sibling test is conducted to ascertain which nodes under consideration are siblings. A family is a group of nodes that are either leaf-siblings or a group of leaf-siblings with their common parent. We then eliminate the nodes that are in families of more than one member. Let $\Delta_{ijk} := d_{ik} - d_{jk}, \forall k \in V \setminus \{i, j\}$ denote the difference between two information distances d_{ik} and d_{jk} . We see that if $\Delta_{ijk} \equiv C, \forall k \in V \setminus \{i, j\}$, where C is some constant, then i and j belong to the same family. Furthermore, if $C = d_{ij}$, we know that i is a leaf and j is its parent.

After the sibling test, we identify nodes that belong to the same family. Latent nodes are introduced when a family of nodes does not contain an observed parent. First, we group all observed nodes into families. Four scenarios may happen: (1) a single node family, (2) a leaf node with a parent, (3) siblings with a parent, and (4) siblings with no parent. The algorithm introduces a new hidden parent in the fourth scenario and removes the current lowest layer nodes from further consideration, fixing this portion of the tree structure. This procedure is repeated recursively and is termed the recursive grouping procedure (Figure 7).

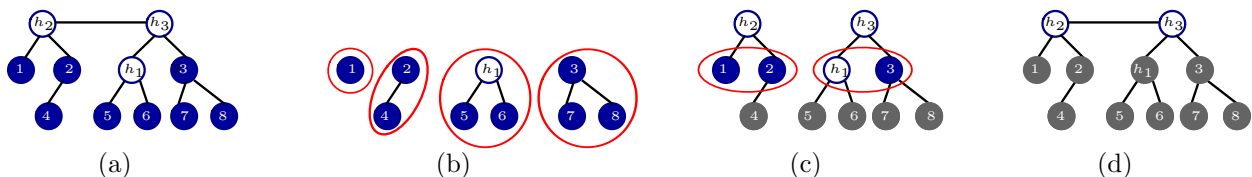


Figure 7: An example of the recursive grouping algorithm to learn the latent tree structure. (a) The ground truth latent tree that we would like to recover. We only observe distances between observed nodes (blue). (b) After grouping the observed variables into families (red curves), the first latent variable is introduced. (c) A portion of the latent tree structure is fixed (grey nodes) and the family identification step is repeated. (d) The latent tree learned by the recursive grouping algorithm.

The latent tree structure learning algorithm recovers accurate models even when applied to noisy biological data. It can be verified from the Central Limit Theorem and continuity arguments [23] that the gap between the true (d_{ij}) and estimated (\hat{d}_{ij}) distances between observable nodes i and j is bounded as

$$\hat{d}_{ij} - d_{ij} = O(n^{-1/2}), \tag{4}$$

where n is the number of samples.

CL-grouping Procedure

The recursive grouping algorithm requires a large number of sibling tests and thus is not computationally efficient. Moreover, merging the results of all these tests can lead to error accumulation. A more efficient alternative is the CLGrouping method (Chow-Liu Grouping) [23]. This method recursively modifies the estimate of the latent nodes by operating on local neighborhoods and adding new latent nodes. In the initial step, the method constructs a minimum spanning tree $\text{MST}(V; \mathbf{d})$ (also known as the Chow-Liu Tree) over the observed nodes (gene nodes in this context) using distances \mathbf{d} . This global step groups the observed nodes that are likely to be close to each other in the true underlying latent tree, thereby guiding subsequent applications of the recursive grouping algorithm. Specifically, the recursive grouping procedure recursively conducts sibling tests in local neighborhoods of the $\text{MST}(V; \mathbf{d})$ in each iteration. Furthermore, if adding latent variables in a local neighborhood does not improve the Bayesian Information Criterion, which is a penalized likelihood score, then those latent variables are rejected. Because the sibling tests are limited to local neighborhoods, the learning procedure is efficient.

Parameter Tuning and Contraction of Latent Nodes

In practice, biological data are noisy and the criteria for the sibling tests need to be relaxed because the estimated distances are imperfect. As discussed above, the family identification test between i and j requires examining estimated $\widehat{\Delta}_{ijk}$ for all $k \in V \setminus \{i, j\}$. If we allow for relaxation, then

$$\widehat{\Delta}_{ijk} \equiv C + \epsilon_1 \tag{5}$$

is the relaxed condition for the family test, and

$$C = \widehat{d}_{ij} + \epsilon_2 \tag{6}$$

is the relaxed condition for the parent-child test. ϵ_1 and ϵ_2 are relaxation parameters. k -means clustering using the silhouette method is employed to select the relaxation parameters for the sibling tests.

In addition, a contraction step removes latent nodes that have information distance smaller than some predefined threshold to some observed node(s). In our experiments, latent nodes are contracted if the information distance is less than 0.9365. We select this threshold because it produces a latent tree with 90 latent nodes, which is similar to the number of yeast TFs in the TF-gene binding dataset we use to annotate

the latent tree [31]. General strategies for controlling model complexity in graphical models are beyond the scope of this work. However, when there is no prior expectation for the number of latent nodes, the latent node activity signatures (as shown in Figure 6a) can guide the choice of the contraction parameter. If there are many latent nodes with similar activity signatures, then the contraction step may be beneficial for controlling redundancy in addition to the Bayesian Information Criterion regularization that is already part of the structure learning.

After learning the structure and parameters of the latent tree model, we perform Gaussian belief propagation [83] on the tree to obtain the conditional means for LNs conditioned on the samples of the observed gene nodes. In order to estimate these conditional means, we use the signed correlation coefficients, whereas the information distances calculated for structure learning use absolute values of the correlation coefficients.

Data

We downloaded microarray expression data from [30], which consists of 498 samples including many stress response conditions as well as other experiments such as cell cycle (data compiled from [1, 84, 85] and other studies)¹. We imputed the expression data to account for missing data (Supplementary Methods). For the TF-gene binding data we used only interactions with p -value less than 0.001 and binding motifs conserved in at least two other yeast species [31]. We filtered these TF-gene interactions to remove gene targets that are not expressed and retain 96 TFs that bind at least one expressed gene. We obtained physical protein-protein and genetic interactions from BioGRID version 3.2.96 [44] and removed all self-interactions and interactions involving non-yeast proteins. To identify indirect TF-TF interactions we searched for pairs of TFs that do not interact directly but have a common neighbor in the interaction network. We excluded physical-genetic interactions where the neighbor has a protein-protein interaction with one TF and a genetic interaction with the other.

Gene Selection

In large-scale genome-wide expression data, the number of genes is usually much larger than the number of samples. Dimensionality reduction or feature selection is beneficial when analyzing such datasets. Traditional

¹The dataset is not longer available from the original website but can be accessed from the Internet Archive at <https://web.archive.org/web/20130629025218/http://gasch.genetics.wisc.edu/datasets.html>

methods such as PCA select principle components for dimensionality reduction, but this leads to a loss of information from minor components. In our work, PCA is not used to select features because there is no biological reason to believe the minor components are unimportant. Instead, we focus on genes that exhibit condition-specific expression changes because these genes enable us to recover condition-specific regulatory modules. Specifically, we compute the covariance for all gene-gene pairs and remove a gene if its maximum covariance with all other genes is less than 0.8683. We choose this threshold because it selects approximately 1000 covarying genes. After filtering, 1035 genes are retained out of all 5998 genes.

Annotating LNs with TFs and GO Terms

The TF binding data are used to annotate the LNs of the latent tree. We search for a TF that binds many of the gene nodes that are in the neighborhood of a latent node, which suggests that the TF may cause the gene nodes’ expression levels to be correlated across the samples. We define an extended neighborhood of influence of a LN and compare it to the set of genes bound by a TF to see if there is a statistically significant overlap between these two groups of genes.

The extended neighborhood of influence for each LN consists of the set of genes that are highly correlated with that latent node. More precisely, a distance matrix $Dist \in \mathbb{R}^{k \times p}$ is calculated with each entry representing

$$Dist_{i,j} := -\log(|\rho(y_i, g_j)|), \quad (7)$$

where k is the number of latent variables, p is the number of observed gene variables, and $\rho(\cdot)$ is the correlation coefficient between latent node y_i and gene g_j . Thus a smaller information distance implies a higher correlation.

The threshold to select the extended neighborhoods of influence for latent nodes is dynamically tuned. We define $d_{\min} := \min_{i,j} Dist_{i,j}$ and $d_{\max} := \max_{i,j} Dist_{i,j}$ and select the neighbors of latent variable y_i

$$\mathcal{N}(y_i) := \{g_j : Dist_{i,j} \leq d_{\min} + \lambda(d_{\max} - d_{\min})\}, \quad (8)$$

where λ is a tunable parameter. Smaller λ leads to more stringent neighborhood selection. In our experiments, we set $\lambda = 0.15$.

We compare the extended neighborhood of influence of a LN with the set of genes bound by a TF to see if the overlap is statistically significant. We use Fisher’s exact test [86] to test all TF-LN pairs using the

1035 genes and control the FDR under 0.05. Benjamini–Yekutieli’s [87] method can be used to adjust raw p -values and obtain the decision rule based on the adjusted p -values, thus controlling the FDR. However, Benjamini–Yekutieli’s method assumes positive regressive correlations, which need not hold in our setting. Instead, we employ a Bayesian approach to estimate the FDR under arbitrary correlations using the `fdrtool` R package [88]. This approach can directly calculate FDR instead of thresholding on the adjusted p -values (Supplementary Methods).

We annotate latent nodes with GO biological process terms using the same statistical test described above for associating TFs and latent nodes and the same FDR threshold of 0.05. We downloaded yeast GO Slim mappings from the *Saccharomyces* Genome Database [89,90] on May 9, 2013 and retained only the biological process terms.

Motif Enrichment

We performed *de novo* motif discovery with WebMOTIFS [46]. For each latent node that did not overlap with any TFs, we searched for significant motifs in its extended neighborhood using the default WebMOTIFS settings (search sequences from -500 to +200 of the transcription start site, expected motif length less than 12 nucleotides, and strict significance testing). WebMOTIFS ran four motif discovery algorithms — AlignACE [91], MDscan [92], MEME [93], and Weeder [94] — on each set of genes and clustered the significant motifs. We aligned the significant motifs to yeast motifs in JASPAR [48] using STAMP [47] with the default parameters (Pearson correlation coefficient comparison metric and ungapped Smith-Waterman alignment) but did not trim motif edges. Supplementary Table 5 reports the discovered motifs and the top matching known TF binding motifs. We filtered the matching motifs to exclude non-yeast TFs.

Osmotic Stress Regulators

To predict TFs that are active in the osmotic stress response, we identified 14 samples in the expression dataset that contain the word ‘sorbitol’ in their name. Sorbitol induces hyperosmotic stress response [1]. We ranked the latent nodes by their conditional means in these osmotic stress conditions and selected the top 50% as the active hidden regulators. We defined the osmotic stress TFs as all TFs that are significantly associated with these latent nodes (Figure 5). The osmotic stress TFs did not change when we selected only

the top 40% or 30% of latent nodes or restricted the analysis to the seven sorbitol samples that did not involve a simultaneous temperature change.

ARACNE Comparison

We ran ARACNE on the filtered expression dataset of 1035 genes with geWorkbench [95], which took approximately 8 hours on machine with a 2.9 GHz Intel Core i7 and 8 GB 1600 MHz DDR3 memory. We were unable to run ARACNE on the full unfiltered expression matrix in a reasonable amount of time. We set the ‘Kernel Width’ to be ‘Inferred’, ‘p-value’ to be ‘1.0E-7’, and ‘DPI Tolerance’ to be ‘0.1’.

Software

The latent tree method from [23] is implemented in MATLAB and is available at [96].

Authors’ Contributions

AG and FH collected the data and performed the computational analysis. FH and RV implemented the latent tree algorithm. AG, FH, EF, and AA designed the study, analyzed the data, and wrote the manuscript. EF and AA supervised the study. All authors read and approved the final manuscript.

Acknowledgements

FH is supported by NSF BIGDATA IIS-1251267. AA is supported in part by a Microsoft Faculty Fellowship, NSF Career Award CCF-1254106, NSF Award CCF-1219234, and ARO YIP Award W911NF-13-1-0084. EF is supported in part by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the US Army Research Office (the content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred) and by NIH grant R01-GM089903.

References

1. Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, Brown PO: **Genomic expression programs in the response of yeast cells to environmental changes.** *Molecular Biology of the Cell* 2000, **11**(12):4241–4257.

2. Eisen MB, Spellman PT, Brown PO, Botstein D: **Cluster analysis and display of genome-wide expression patterns.** *Proceedings of the National Academy of Sciences* 1998, **95**(25):14863–14868.
3. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N: **Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data.** *Nature Genetics* 2003, **34**(2):166–176.
4. Pasquinelli AE: **MicroRNAs and their targets: recognition, regulation and an emerging reciprocal relationship.** *Nature Reviews Genetics* 2012, **13**(4):271–282.
5. Lefebvre C, Rajbhandari P, Alvarez MJ, Bandaru P, Lim WK, Sato M, Wang K, Sumazin P, Kustagi M, Bisikirska BC, Basso K, Beltrao P, Krogan N, Gautier J, Dalla-Favera R, Califano A: **A human B-cell interactome identifies MYB and FOXM1 as master regulators of proliferation in germinal centers.** *Molecular Systems Biology* 2010, **6**.
6. Bansal M, Belcastro V, Ambesi-Impiombato A, di Bernardo D: **How to infer gene networks from expression profiles.** *Molecular Systems Biology* 2007, **3**.
7. De Smet R, Marchal K: **Advantages and limitations of current network inference methods.** *Nature Reviews Microbiology* 2010, **8**(10):717–729.
8. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, Allison KR, The DREAM5 Consortium, Kellis M, Collins JJ, Stolovitzky G: **Wisdom of crowds for robust gene network inference.** *Nature Methods* 2012, **9**(8):796–804.
9. Chasman D, Fotuhi Siahpirani A, Roy S: **Network-based approaches for analysis of complex biological systems.** *Current Opinion in Biotechnology* 2016, **39**:157–166.
10. Bar-Joseph Z, Gerber GK, Lee TI, Rinaldi NJ, Yoo JY, Robert F, Gordon DB, Fraenkel E, Jaakkola TS, Young RA, Gifford DK: **Computational discovery of gene modules and regulatory networks.** *Nature Biotechnology* 2003, **21**(11):1337–1342.
11. Liao JC, Boscolo R, Yang YL, Tran LM, Sabatti C, Roychowdhury VP: **Network component analysis: reconstruction of regulatory signals in biological systems.** *Proceedings of the National Academy of Sciences* 2003, **100**(26):15522–15527.
12. Wu M, Liu L, Hijazi H, Chan C: **A multi-layer inference approach to reconstruct condition-specific genes and their regulation.** *Bioinformatics* 2013, **29**(12):1541–1552.
13. Arrieta-Ortiz ML, Hafemeister C, Bate AR, Chu T, Greenfield A, Shuster B, Barry SN, Gallitto M, Liu B, Kacmarczyk T, Santoriello F, Chen J, Rodrigues CD, Sato T, Rudner DZ, Driks A, Bonneau R, Eichenberger P: **An experimentally supported model of the Bacillus subtilis global transcriptional regulatory network.** *Molecular Systems Biology* 2015, **11**(11):839.
14. Markowetz F, Kostka D, Troyanskaya OG, Spang R: **Nested effects models for high-dimensional phenotyping screens.** *Bioinformatics* 2007, **23**(13):i305–i312.
15. Haynes BC, Maier EJ, Kramer MH, Wang PI, Brown H, Brent MR: **Mapping functional transcription factor networks from gene expression data.** *Genome Research* 2013, **23**(8):1319–1328.
16. MacNeil LT, Pons C, Arda HE, Giese GE, Myers CL, Walhout AJM: **Transcription factor activity mapping of a tissue-specific in vivo gene regulatory network.** *Cell Systems* 2015, **1**(2):152–162.
17. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A: **ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.** *BMC Bioinformatics* 2006, **7**(suppl 1):S7.
18. Friedman J, Hastie T, Tibshirani R: **Sparse inverse covariance estimation with the graphical lasso.** *Biostatistics* 2008, **9**(3):432–441.
19. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P: **Inferring regulatory networks from expression data Using tree-based methods.** *PLoS ONE* 2010, **5**(9):e12776.
20. Roy S, Lagree S, Hou Z, Thomson JA, Stewart R, Gasch AP: **Integrated module and gene-specific regulatory inference implicates upstream signaling networks.** *PLoS Computational Biology* 2013, **9**(10):e1003252.
21. Lauritzen S: *Graphical models.* Clarendon Press 1996.

22. Karger D, Srebro N: **Learning Markov networks: maximum bounded tree-width graphs**. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms* 2001:392–401.
23. Choi M, Tan V, Anandkumar A, Willsky A: **Learning latent tree graphical models**. *Journal of Machine Learning Research* 2011, **12**:1771–1812.
24. Choi M, Torralba A, Willsky A: **Context models and out-of-context objects**. *Pattern Recognition Letters* 2012, **33**(7):853–862.
25. Wang F, Li Y: **Beyond physical connections: tree models in human pose estimation**. In *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition* 2013.
26. Valluvan R, Almquist ZW, Butts CT, Anandkumar A: **Semi-parametric vertex set prediction for dynamic networks using latent tree models**. In *International Conference for Social Network Analysis (Sunbelt 2012)* 2012.
27. Pearl J: *Causality: models, reasoning and inference, Volume 29*. Cambridge Univ Press 2000.
28. Anandkumar A, Hsu D, Javanmard A, Kakade SM: **Learning linear Bayesian networks with latent variables**. In *Proceedings of the 30th International Conference on Machine Learning* 2013.
29. Anandkumar A, Ge R, Hsu D, Kakade SM, Telgarsky M: **Tensor decompositions for learning latent variable models**. *Journal of Machine Learning Research* 2014, **15**:2773–2832.
30. **Mega yeast gene expression data**[http://gasch.genetics.wisc.edu/datasets/Mega_YeastX.txt.gz].
31. MacIsaac KD, Wang T, Gordon DB, Gifford DK, Stormo GD, Fraenkel E: **An improved map of conserved regulatory sites for *Saccharomyces cerevisiae***. *BMC Bioinformatics* 2006, **7**:113.
32. Margolin AA, Wang K, Lim WK, Kustagi M, Nemenman I, Califano A: **Reverse engineering cellular networks**. *Nature Protocols* 2006, **1**(2):662–671.
33. Natarajan K, Meyer MR, Jackson BM, Slade D, Roberts C, Hinnebusch AG, Marton MJ: **Transcriptional profiling shows that Gcn4p is a master regulator of gene expression during amino acid starvation in yeast**. *Molecular and Cellular Biology* 2001, **21**(13):4347–4368.
34. Hahn JS, Hu Z, Thiele DJ, Iyer VR: **Genome-wide analysis of the biology of stress responses through heat shock transcription factor**. *Molecular and Cellular Biology* 2004, **24**(12):5249–5256.
35. Xu Z, Norris D: **The SFP1 gene product of *Saccharomyces cerevisiae* regulates G2/M transitions during the mitotic cell cycle and DNA-damage response**. *Genetics* 1998, **150**(4):1419–1428.
36. Estruch F: **Stress-controlled transcription factors, stress-induced genes and stress tolerance in budding yeast**. *FEMS Microbiology Reviews* 2000, **24**(4):469–486.
37. de Bruin RAM, Kalashnikova TI, Wittenberg C: **Stb1 collaborates with other regulators to modulate the G1-specific transcriptional circuit**. *Molecular and Cellular Biology* 2008, **28**(22):6919–6928.
38. Wade JT, Hall DB, Struhl K: **The transcription factor Ifh1 is a key regulator of yeast ribosomal protein genes**. *Nature* 2004, **432**(7020):1054–1058.
39. Pierce M, Benjamin KR, Montano SP, Georgiadis MM, Winter E, Vershon AK: **Sum1 and Ndt80 proteins compete for binding to middle sporulation element sequences that control meiotic gene expression**. *Molecular and Cellular Biology* 2003, **23**(14):4814–4825.
40. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: **Gene Ontology: tool for the unification of biology**. *Nature Genetics* 2000, **25**:25–29.
41. Ravasi T, Suzuki H, Cannistraci CV, Katayama S, Bajic VB, Tan K, Akalin A, Schmeier S, Kanamori-Katayama M, Bertin N, Carninci P, Daub CO, Forrest ARR, Gough J, Grimmond S, Han JH, Hashimoto T, Hide W, Hofmann O, Kamburov A, Kaur M, Kawaji H, Kubosaki A, Lassmann T, van Nimwegen E, MacPherson CR, Ogawa C, Radovanovic A, Schwartz A, Teasdale RD, Tegnér J, Lenhard B, Teichmann SA, Arakawa T, Ninomiya N, Murakami K, Tagami M, Fukuda S, Imamura K, Kai C, Ishihara R, Kitazume Y, Kawai J, Hume DA, Ideker T, Hayashizaki Y: **An atlas of combinatorial transcriptional regulation in mouse and man**. *Cell* 2010, **140**(5):744–752.
42. Wasson T, Hartemink AJ: **An ensemble model of competitive multi-factor binding of the genome**. *Genome Research* 2009, **19**(11):2101–2112.

43. Gitter A, Siegfried Z, Klutstein M, Fornes O, Oliva B, Simon I, Bar-Joseph Z: **Backup in gene regulatory networks explains differences between binding and knockout results.** *Molecular Systems Biology* 2009, **5**.
44. Chatr-aryamontri A, Breitkreutz BJ, Heinicke S, Boucher L, Winter A, Stark C, Nixon J, Ramage L, Kolas N, O'Donnell L, Reguly T, Breitkreutz A, Sellam A, Chen D, Chang C, Rust J, Livstone M, Oughtred R, Dolinski K, Tyers M: **The BioGRID interaction database: 2013 update.** *Nucleic Acids Research* 2013, **41**(D1):D816–D823.
45. Traven A, Jelicic B, Sopta M: **Yeast Gal4: a transcriptional paradigm revisited.** *EMBO Reports* 2006, **7**(5):496–499.
46. Romer KA, Kayombya GR, Fraenkel E: **WebMOTIFS: automated discovery, filtering and scoring of DNA sequence motifs using multiple programs and Bayesian approaches.** *Nucleic Acids Research* 2007, **35**(suppl 2):W217–W220.
47. Mahony S, Benos PV: **STAMP: a web tool for exploring DNA-binding motif similarities.** *Nucleic Acids Research* 2007, **35**(suppl 2):W253–W258.
48. Portales-Casamar E, Thongjuea S, Kwon AT, Arenillas D, Zhao X, Valen E, Yusuf D, Lenhard B, Wasserman WW, Sandelin A: **JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles.** *Nucleic Acids Research* 2010, **38**(suppl 1):D105–D110.
49. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW, Hannett NM, Tagne JB, Reynolds DB, Yoo J, Jennings EG, Zeitlinger J, Pokholok DK, Kellis M, Rolfe PA, Takusagawa KT, Lander ES, Gifford DK, Fraenkel E, Young RA: **Transcriptional regulatory code of a eukaryotic genome.** *Nature* 2004, **431**(7004):99–104.
50. Leek JT, Storey JD: **Capturing heterogeneity in gene expression studies by surrogate variable analysis.** *PLoS Genetics* 2007, **3**(9):e161.
51. Leek JT, Johnson WE, Parker HS, Jaffe AE, Storey JD: **The sva package for removing batch effects and other unwanted variation in high-throughput experiments.** *Bioinformatics* 2012, **28**(6):882–883.
52. Chikina M, Zaslavsky E, Sealfon SC: **CellCODE: a robust latent variable approach to differential expression analysis for heterogeneous cell populations.** *Bioinformatics* 2015, **31**(10):1584–1591.
53. He Y, Qi Y, Kavukcuoglu K, Park H: **Learning the dependency structure of latent factors.** In *Advances in Neural Information Processing Systems 25*. Edited by Pereira F, Burges CJC, Bottou L, Weinberger KQ 2012:2366–2374.
54. Chen X, Storey JD: **Consistent estimation of low-dimensional latent structure in high-dimensional data.** *arXiv:1510.03497* 2015.
55. Tan J, Ung M, Cheng C, Greene CS: **Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders.** *Pacific Symposium on Biocomputing* 2014, **20**:132–143.
56. Chen L, Cai C, Chen V, Lu X: **Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model.** *BMC Bioinformatics* 2016, **17**:97–107.
57. Tian D, Gu Q, Ma J: **Identifying gene regulatory network rewiring using latent differential graphical models.** *Nucleic Acids Research* 2016.
58. Gao C, McDowell IC, Zhao S, Brown CD, Engelhardt BE: **Context specific and differential gene co-expression networks via Bayesian biclustering.** *PLoS Computational Biology* 2016, **12**(7):e1004791.
59. Hartemink AJ, Gifford DK, Jaakkola TS, Young RA: **Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks.** *Pacific Symposium on Biocomputing* 2001, **6**:422–433.
60. Yoo C, Thorsson V, Cooper GF: **Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data.** *Pacific Symposium on Biocomputing* 2002, **7**:498–509.
61. Zhang X, Cheng W, Listgarten J, Kadie C, Huang S, Wang W, Heckerman D: **Learning transcriptional regulatory relationships using sparse graphical models.** *PLoS ONE* 2012, **7**(5):e35762.

62. Celik S, Logsdon BA, Battle S, Drescher CW, Rendi M, Hawkins RD, Lee SI: **Extracting a low-dimensional description of multiple gene expression datasets reveals a potential driver for tumor-associated stroma in ovarian cancer.** *Genome Medicine* 2016, **8**:66.
63. Nagashima T, Oyama M, Kozuka-Hata H, Yumoto N, Sakaki Y, Hatakeyama M: **Phosphoproteome and transcriptome analyses of ErbB ligand-stimulated MCF-7 cells.** *Cancer Genomics-Proteomics* 2008, **5**(3-4):161–168.
64. Ghazalpour A, Bennett B, Petyuk VA, Orozco L, Hagopian R, Mungrue IN, Farber CR, Sinsheimer J, Kang HM, Furlotte N, Park CC, Wen PZ, Brewer H, Weitz K, Camp DG II, Pan C, Yordanova R, Neuhaus I, Tilford C, Siemers N, Gargalovic P, Eskin E, Kirchgessner T, Smith DJ, Smith RD, Lusk AJ: **Comparative analysis of proteome and transcriptome variation in mouse.** *PLoS Genetics* 2011, **7**(6):e1001393.
65. Waters KM, Liu T, Quesenberry RD, Willse AR, Bandyopadhyay S, Kathmann LE, Weber TJ, Smith RD, Wiley HS, Thrall BD: **Network analysis of epidermal growth factor signaling using integrated genomic, proteomic and phosphorylation data.** *PLoS ONE* 2012, **7**(3):e34515.
66. Osmanbeyoglu HU, Pelossof R, Bromberg JF, Leslie CS: **Linking signaling pathways to transcriptional programs in breast cancer.** *Genome Research* 2014, **24**(11):1869–1880.
67. Stefan D, Pinel C, Pinhal S, Cinquemani E, Geiselmann J, de Jong H: **Inference of quantitative models of bacterial promoters from time-series reporter gene data.** *PLoS Computational Biology* 2015, **11**:e1004028.
68. O’Connell DJ, Kolde R, Sooknah M, Graham DB, Sundberg TB, Latorre IJ, Mikkelsen TS, Xavier RJ: **Simultaneous pathway activity inference and gene expression analysis using RNA sequencing.** *Cell Systems* 2016, **2**(5):323–334.
69. Gitter A, Carmi M, Barkai N, Bar-Joseph Z: **Linking the signaling cascades and dynamic regulatory networks controlling stress responses.** *Genome Research* 2013, **23**(2):365–376.
70. Chasman D, Ho YH, Berry DB, Nemecek CM, MacGilvray ME, Hose J, Merrill AE, Lee MV, Will JL, Coon JJ, Ansari AZ, Craven M, Gasch AP: **Pathway connectivity and signaling coordination in the yeast stress-activated signaling network.** *Molecular Systems Biology* 2014, **10**(11):759.
71. Capaldi AP, Kaplan T, Liu Y, Habib N, Regev A, Friedman N, O’Shea EK: **Structure and function of a transcriptional network activated by the MAPK Hog1.** *Nature Genetics* 2008, **40**(11):1300–1306.
72. **Module networks: discovering regulatory modules and their condition specific regulators from gene expression data**[http://ai.stanford.edu/~erans/module_nets/modules.html].
73. Thomson DW, Dinger ME: **Endogenous microRNA sponges: evidence and controversy.** *Nature Reviews Genetics* 2016, **17**(5):272–283.
74. Yim WC, Yu Y, Song K, Jang CS, Lee BM: **PLANEX: the plant co-expression database.** *BMC Plant Biology* 2013, **13**:83.
75. Zhang H, Jin J, Tang L, Zhao Y, Gu X, Gao G, Luo J: **PlantTFDB 2.0: update and improvement of the comprehensive plant transcription factor database.** *Nucleic Acids Research* 2011, **39**(suppl 1):D1114–D1117.
76. Barretina J, Caponigro G, Stransky N, Venkatesan K, Margolin AA, Kim S, Wilson CJ, Lehár J, Kryukov GV, Sonkin D, Reddy A, Liu M, Murray L, Berger MF, Monahan JE, Morais P, Meltzer J, Korejwa A, Jane-Valbuena J, Mapa FA, Thibault J, Bric-Furlong E, Raman P, Shipway A, Engels IH, Cheng J, Yu GK, Yu J, Aspesi P, de Silva M, Jagtap K, Jones MD, Wang L, Hatton C, Palesscandolo E, Gupta S, Mahan S, Sougnez C, Onofrio RC, Liefeld T, MacConaill L, Winckler W, Reich M, Li N, Mesirov JP, Gabriel SB, Getz G, Ardlie K, Chan V, Myer VE, Weber BL, Porter J, Warmuth M, Finan P, Harris JL, Meyerson M, Golub TR, Morrissey MP, Sellers WR, Schlegel R, Garraway LA: **The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity.** *Nature* 2012, **483**(7391):603–607.
77. The Cancer Genome Atlas Network: **Comprehensive molecular portraits of human breast tumours.** *Nature* 2012, **490**(7418):61–70.
78. Neph S, Vierstra J, Stergachis AB, Reynolds AP, Haugen E, Vernot B, Thurman RE, John S, Sandstrom R, Johnson AK, Maurano MT, Humbert R, Rynes E, Wang H, Vong S, Lee K, Bates D, Diegel M, Roach V, Dunn D, Neri J, Schafer A, Hansen RS, Kutayavin T, Giste E, Weaver M, Canfield T, Sabo P, Zhang M, Balasundaram G, Byron R, MacCoss MJ, Akey JM, Bender MA, Groudine M, Kaul R, Stamatoyannopoulos JA: **An expansive human regulatory lexicon encoded in transcription factor footprints.** *Nature* 2012, **489**(7414):83–90.

79. Friedman RC, Farh KKH, Burge CB, Bartel DP: **Most mammalian mRNAs are conserved targets of microRNAs.** *Genome Research* 2009, **19**:92–105.
80. Erdős PL, Székely LA, Steel MA, Warnow TJ: **A few logs suffice to build (almost) all trees (I).** *Random Structures and Algorithms* 1999, **14**(2):153–184.
81. Mossel E: **Distorted metrics on trees and phylogenetic forests.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2007, **4**:108–116.
82. Dempster AP, Laird NM, Rubin DB: **Maximum likelihood from incomplete data via the EM algorithm.** *Journal of the Royal Statistical Society. Series B (Methodological)* 1977, **39**:1–38.
83. Yedidia J, Freeman W, Weiss Y: **Understanding belief propagation and its generalizations.** In *Exploring Artificial Intelligence in the New Millennium*. Edited by Lakemeyer G, Nebel B 2003:239–236.
84. Gasch AP, Huang M, Metzner S, Botstein D, Elledge SJ, Brown PO: **Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p.** *Molecular Biology of the Cell* 2001, **12**(10):2987–3003.
85. Lyons TJ, Gasch AP, Gaither LA, Botstein D, Brown PO, Eide DJ: **Genome-wide characterization of the Zap1p zinc-responsive regulon in yeast.** *Proceedings of the National Academy of Sciences* 2000, **97**(14):7957–7962.
86. Fisher RA: **On the interpretation of χ^2 from contingency tables, and the calculation of P.** *Journal of the Royal Statistical Society* 1922, **85**:87–94.
87. Benjamini Y, Yekutieli D: **The control of the false discovery rate in multiple testing under dependency.** *Annals of Statistics* 2001, **29**(4):1165–1188.
88. Strimmer K: **fdrtool: a versatile R package for estimating local and tail area-based false discovery rates.** *Bioinformatics* 2008, **24**(12):1461–1462.
89. Saccharomyces Genome Database staff: **GO Slim**[<http://www.yeastgenome.org/download-data/curation>].
90. Cherry JM, Hong EL, Amundsen C, Balakrishnan R, Binkley G, Chan ET, Christie KR, Costanzo MC, Dwight SS, Engel SR, Fisk DG, Hirschman JE, Hitz BC, Karra K, Krieger CJ, Miyasato SR, Nash RS, Park J, Skrzypek MS, Simison M, Weng S, Wong ED: **Saccharomyces Genome Database: the genomics resource of budding yeast.** *Nucleic Acids Research* 2012, **40**(D1):D700–D705.
91. Hughes JD, Estep PW, Tavazoie S, Church GM: **Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*.** *Journal of Molecular Biology* 2000, **296**(5):1205–1214.
92. Liu XS, Brutlag DL, Liu JS: **An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments.** *Nature Biotechnology* 2002, **20**(8):835–839.
93. Bailey TL, Elkan C: **Fitting a mixture model by expectation maximization to discover motifs in biopolymers.** In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, Volume 2* 1994:28–36.
94. Pavesi G, Mereghetti P, Mauri G, Pesole G: **Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes.** *Nucleic Acids Research* 2004, **32**(suppl 2):W199–W203.
95. **geWorkbench**[<http://wiki.c2b2.columbia.edu/workbench/index.php/Home>].
96. **Learning latent tree graphical models**[<http://people.csail.mit.edu/myungjin/latentTree.html>].
97. Wayman JC: **Multiple imputation for missing data: what is it and how can I use it.** In *Annual Meeting of the American Educational Research Association* 2003.
98. Honaker J, King G, Blackwell M: **Amelia II: a program for missing data**[<http://gking.harvard.edu/amelia>].
99. Grenander U: **On the theory of mortality measurement.** *Scandinavian Actuarial Journal* 1956, **1956**(2):125–153.
100. Strimmer K: **A unified approach to false discovery rate estimation.** *BMC Bioinformatics* 2008, **9**:303.

Ancillary Files

SupplementaryTables.xlsx

The following tables are contained in the spreadsheet:

- Supplementary Table 1. Latent tree network adjacency list.
- Supplementary Table 2. Latent node extended neighborhoods.
- Supplementary Table 3. Significance of the TF annotations of the latent nodes.
- Supplementary Table 4. Significance of the GO annotations of the latent nodes.
- Supplementary Table 5. *De novo* motifs discovered for the unannotated latent nodes.

Supplementary Methods and Figures

Missing Data

We use a multiple imputation method [97] to estimate the missing values in the yeast gene expression dataset. This strategy has been shown to reduce bias and increase efficiency compared to other ad-hoc methods such as listwise deletion and mean imputation. Multiple imputation is a procedure for imputing multiple values for every missing data point, which results in multiple complete data sets. Those imputed values are sampled from a distribution inferred from the observed values. After imputation with expectation maximization [82], we combine the multiple results.

We assume the gene expression levels are multivariate normal $D \sim N(\mu, \Sigma)$ and that data are missing at random. Let D_o denote observed data and D_m the missing data. The posterior $p(\mu, \Sigma | D_o) \propto p(D_o | \mu, \Sigma)$ are the main imputation parameters we want to estimate because imputations can be made by sampling missing values from $p(D_m | D_o, \mu, \Sigma)$. We use EM to find the mode of the posterior. The R package Amelia II, a general-purpose tool for data with missing values, is used to perform multiple imputation [98]. We perform imputation on the log scale gene expression data.

We checked the accuracy of the imputation by 10-fold cross validation. The average error is 0.0015, and the errors for all genes are depicted in Supplementary Figure 2.

False Discovery Rate Control for Multiple Testing

Here we describe the details of controlling for multiple hypothesis testing when associating TFs or GO terms with latent nodes. We introduce the following definitions:

1. Multiple testing: m null hypotheses $\mathcal{H}_{0i}, \forall i \in [m]$ are tested simultaneously. Each p -value associated with \mathcal{H}_{0i} is denoted as p_i , and $\hat{P}^m := \{p_1, p_2, \dots, p_m\}$ is the set of all observed p -values. m_0 of the m hypotheses are the true null hypotheses and thus the proportion of true null hypotheses is $\eta_0 = \frac{m_0}{m}$.
2. Decision rule: Decision rule θ is a mapping from observations of multiple test p -values \hat{P}^m to an action $\{A_{\text{rej}}, A_{\overline{\text{rej}}}\}$, where A_{rej} denotes the ‘reject’ action and $A_{\overline{\text{rej}}}$ denotes the ‘non-reject’ action. A decision rule with cutoff p -value y_c is defined as

$$\theta(\hat{P}^m, y_c) = \begin{cases} A_{\text{rej } \mathcal{H}_{0i}} & , \forall i : p_i \leq y_c \\ A_{\overline{\text{rej}} \mathcal{H}_{0j}} & , \forall j : p_j > y_c \end{cases} \quad (9)$$

where $A_{\text{rej } \mathcal{H}_{0i}}$ is the action to reject the i -th null hypothesis, and $A_{\overline{\text{rej}} \mathcal{H}_{0j}}$ means do not reject the j -th null hypothesis.

3. False discovery rate: defined as

$$Fdr(\theta(\hat{P}^m, y_c)) := \Pr\{A_{\text{rej } \mathcal{H}_{0i}} | p_i \leq y_c\} = \frac{FP}{TP + FP} \quad (10)$$

where TP are the true positives and FP are the false positives.

The classic tail area-based false discovery rate (FDR) is the type of false discovery rate we focus on in this work. The two component mixture of the observed p -values is

$$f(p) = \begin{cases} \eta_0 f_0(p; \phi) & , 0 \leq p \leq y_c \\ (1 - \eta_0) f_A(p) & , y_c \leq p \leq 1 \end{cases} = \eta_0 f_0(p; \phi) \mathbb{I}\{0 \leq p \leq y_c\} + (1 - \eta_0) f_A(p) \mathbb{I}\{y_c \leq p \leq 1\}, \quad (11)$$

where $f(p), p \in [0, 1]$ is the overall p -value density function. Let y_c denote the cutoff point under which null hypotheses with smaller p -values ($p(\mathcal{H}_{0m}) \leq y_c$) are rejected. Let $f_A(p)$ denote the alternative density with support $y_c \leq p \leq 1$. We assume the null density $f_0(p; \phi), 0 \leq p \leq y_c$ to be a null model with parameter ϕ whose support is the ‘uninteresting’ p -values (null hypotheses are rejected when corresponding p -values are ‘interesting’, i.e. $p \leq y_c$). Thus we have

$$F(p) = \eta_0 F_0(p; \phi) \mathbb{I}\{0 \leq p \leq y_c\} + (1 - \eta_0) F_A(p) \mathbb{I}\{y_c \leq p \leq 1\}, \quad (12)$$

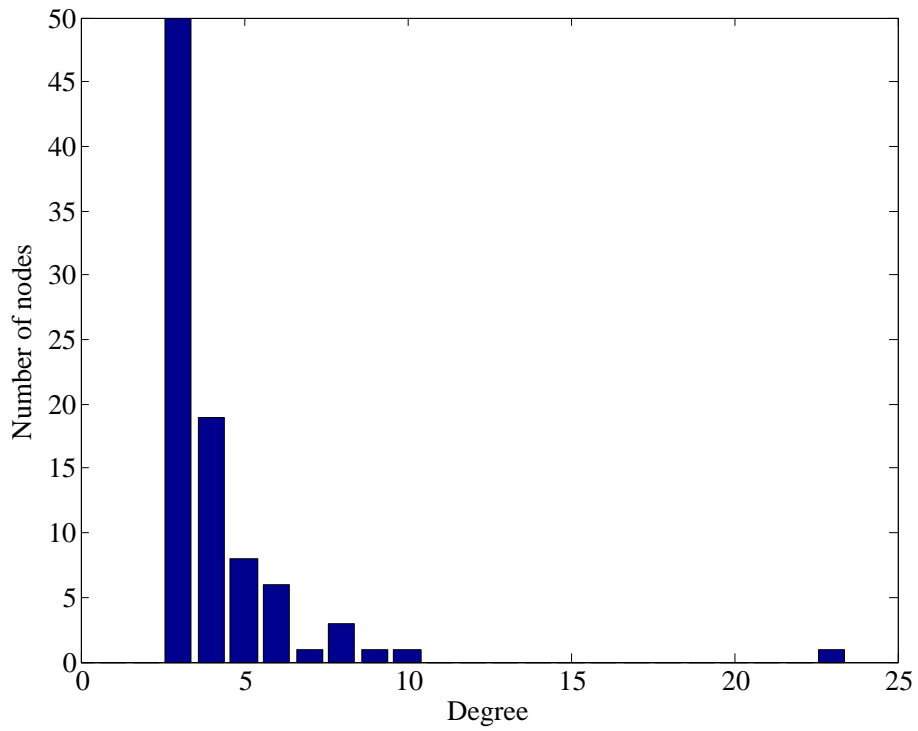
where $F(p)$ is the cumulative distribution function associated with the marginal density $f(p)$. Note that η_0 is the null proportion that we are to estimate.

Fdr is estimated by fitting the two component FDR mixture model in Equation (12) and estimating the cumulative distribution function $F(p)$, the proportion of true null hypotheses η_0 , and $F_0(p; \phi)$ because

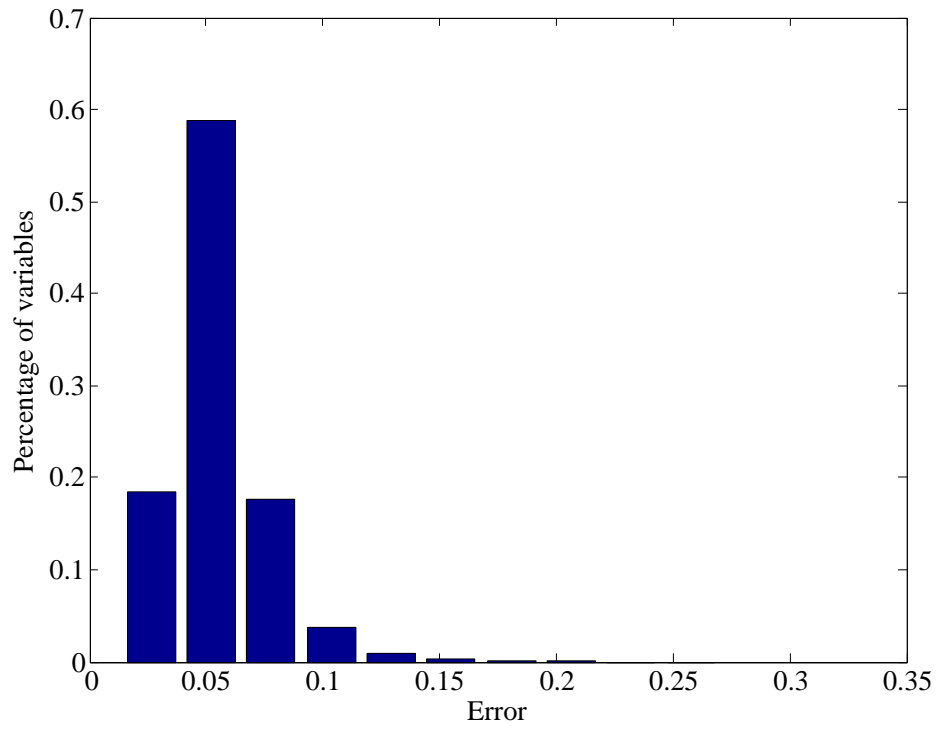
$$Fdr\left(\theta(\hat{P}^m, y_c)\right) = \eta_0 \frac{1 - F_0(y_c)}{1 - F(y_c)}. \quad (13)$$

The Grenander density estimator [99] is used to estimate the cumulative distribution function $F(p)$, which is the decreasing piecewise-constant function equal to the slopes of the least concave majorant of the empirical cumulative distribution function [100].

To estimate η_0 and $F_0(p; \phi)$, we apply a truncated maximum likelihood approach [88]. Once we obtain estimates of η_0 and $F_0(p; \phi)$ we have false discovery rate control for the latent node association testing.



Supplementary Figure 1: Degree distribution of the 90 latent nodes.



Supplementary Figure 2: Gene expression imputation error via 10-fold cross validation.