BMC Bioinformatics

BioMed Central

Proceedings

Open Access

A classification-based framework for predicting and analyzing gene regulatory response

Anshul Kundaje[1], Manuel Middendorf[2], Mihir Shah[1], Chris H Wiggins[3,4], Yoav Freund[1,4,5] and Christina Leslie*[1,4,5]

Address: [1]Department of Computer Science, Columbia University, New York, NY 10027, USA, [2]Department of Physics, Columbia University, New York, NY 10027, USA, [3]Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, USA, [4]Center for Computational Biology and Bioinformatics, Columbia University, New York, NY 10027, USA and [5]Center for Computational Learning Systems, Columbia University, New York, NY 10027, USA

Email: Anshul Kundaje - abk2001@cs.columbia.edu; Manuel Middendorf - mjm2007@columbia.edu; Mihir Shah - ms2604@columbia.edu; Chris H Wiggins - chris.wiggins@columbia.edu; Yoav Freund - freund@cs.columbia.edu; Christina Leslie* - cleslie@cs.columbia.edu

* Corresponding author

from NIPS workshop on New Problems and Methods in Computational Biology
Whistler, Canada. 18 December 2004

Published: 20 March 2006

BMC Bioinformatics 2006, **7**(Suppl 1):S5 doi:10.1186/1471-2105-7-S1-S5

Abstract

Background: We have recently introduced a predictive framework for studying gene transcriptional regulation in simpler organisms using a novel supervised learning algorithm called GeneClass. GeneClass is motivated by the hypothesis that in model organisms such as *Saccharomyces cerevisiae*, we can learn a decision rule for predicting whether a gene is up- or down-regulated in a particular microarray experiment based on the presence of binding site subsequences ("motifs") in the gene's regulatory region and the expression levels of regulators such as transcription factors in the experiment ("parents"). GeneClass formulates the learning task as a classification problem — predicting +1 and -1 labels corresponding to up- and down-regulation beyond the levels of biological and measurement noise in microarray measurements. Using the Adaboost algorithm, GeneClass learns a prediction function in the form of an alternating decision tree, a margin-based generalization of a decision tree.

Methods: In the current work, we introduce a new, robust version of the GeneClass algorithm that increases stability and computational efficiency, yielding a more scalable and reliable predictive model. The improved stability of the prediction tree enables us to introduce a detailed post-processing framework for biological interpretation, including individual and group target gene analysis to reveal condition-specific regulation programs and to suggest signaling pathways. Robust GeneClass uses a novel stabilized variant of boosting that allows a set of correlated features, rather than single features, to be included at nodes of the tree; in this way, biologically important features that are correlated with the single best feature are retained rather than decorrelated and lost in the next round of boosting. Other computational developments include fast matrix computation of the loss function for all features, allowing scalability to large datasets, and the use of abstaining weak rules, which results in a more shallow and interpretable tree. We also show how to incorporate genome-wide protein-DNA binding data from ChIP chip experiments into the GeneClass algorithm, and we use an improved noise model for gene expression data.

Results: Using the improved scalability of Robust GeneClass, we present larger scale experiments on a yeast environmental stress dataset, training and testing on *all* genes and using a comprehensive set of potential regulators. We demonstrate the improved stability of the features in the learned prediction tree, and we show the utility of the post-processing framework by analyzing two groups of genes in yeast — the protein chaperones and a set of putative targets of the Nrg1 and Nrg2 transcription factors — and suggesting novel hypotheses about their transcriptional and post-transcriptional regulation. Detailed results and Robust GeneClass source code is available for download from http://www.cs.columbia.edu/compbio/robust-geneclass.

Background

Understanding the underlying mechanisms of gene transcriptional regulation through analysis of high-throughput genomic data — including gene expression data from microarray experiments, regulatory sequence data, and protein-DNA binding data from new experimental techniques like ChIP chip assays — has become a central goal in computational biology. For simpler model organisms such as *S. cerevisiae*, there have been several computational approaches integrating heterogeneous data sources, in particular gene expression and regulatory sequence data, to solve different problems in gene regulation: identification of regulatory elements in non-coding DNA [1,2], discovery of co-occurrence of regulatory motifs and combinatorial effects of regulatory molecules [3] and organization of genes that appear to be subject to common regulatory control into "regulatory modules" [4-6]. Among recent studies that try to learn a global model of gene regulation in an organism — rather than simply extracting statistically significant regulatory patterns — most attempt to discover structure in the dataset as formalized by probabilistic models [5-10] (often graphical models or Bayesian networks). Most of these *structure learning* approaches build a model from training data and provide useful exploratory tools that allow the user to generate biological hypotheses about transcriptional regulation from the model; however, these models are rarely used to try to make accurate *predictions* about which genes will be up- or down-regulated in new or held-out experiments (test data). One partial exception is the recent work of Beer and Tavazoie [6], which does make predictions and report performance on test data: the authors cluster gene expression profiles, use a motif discovery algorithm to find putative binding motifs in the upstream sequences of genes in each cluster, and then learn a Bayes net model for predicting cluster membership based on motif data. However, since the clustering and motif discovery steps rely on using all the data (training and test), and since the predictions are relative to fixed clusters on one dataset, the method does not easily generalize to entirely new or held-out microarray experiments. While these probabilistic approaches give a rich description of biological data and a way to generate hypotheses, the often missing validation on an independent test set makes it difficult to directly compare performance of the different algorithms or to decide whether the model has overfit the training data.

We have recently presented an alternative *predictive modeling* framework for the computational study of gene regulation, based on supervised learning — in particular, large-margin classification — rather than probabilistic generative models. The core of our approach is an algorithm called GeneClass [11,12] that learns a *prediction* function for the regulatory response of genes under different experimental conditions. The inputs to our learning algorithm are the gene-specific regulatory sequences — represented by the set of binding site patterns they contain ("motifs") — and the experiment-specific expression levels of regulators ("parents"). The output is a prediction of the expression state of the regulated gene in a particular experiment. By predicting only whether the gene is up- or down-regulated — rather than a real-valued expression level — we exploit modern and effective classification algorithms. GeneClass uses the Adaboost learning algorithm with a margin-based generalization of decision trees called alternating decision trees (ADTs). We evaluated the performance of our method by measuring prediction accuracy on held-out microarray experiments and achieved very good classification results in this setting [11]; moreover, we gave basic methods for post-processing the learned ADT to extract significant regulators, motifs, and motif-regulator pairs [11,12].

In the current work, we present a robust version of GeneClass incorporating computational improvements that permit larger feature spaces and datasets, allowing us to use the full set of regulators and make predictions for all genes, and introducing a stabilized variant of boosting that yields a more robust model. The improved stability of the prediction tree allows us to perform detailed condition-specific post-processing of the learned models, including individual and group target gene analysis and signaling pathway analysis. The main idea in our stabilized boosting approach is to allow a set of correlated features, rather than single features, to be included at nodes of the tree. In regular boosting, biologically important features that are correlated with the single best feature are decorrelated in the next round of boosting and may fail to be captured by the model. Stabilized boosting retains these correlated features, so that in post-processing we obtain more stable ranked lists of features. Robust GeneClass also incorporates other computational developments, including fast matrix computation of the loss function for all features to allow scalability to large datasets and the use of abstaining weak rules, which results in a shallower and more interpretable tree. We also improve our noise model for discretization of expression values, and we show how GeneClass can integrate expression data either with motif sequence features — representing known or putative transcription factor binding sites — or with transcription factor occupancy data from genome-wide protein-DNA binding assays, enabling new analysis about the relationship between signaling molecule expression and transcription factor activity. We report large scale computational experiments on a yeast environmental stress dataset, and we apply our post-processing framework to analyze the condition-specific regulation of a group of protein chaperone genes. We also analyze a set of genes that are believed to be targets of the Nrg1 and

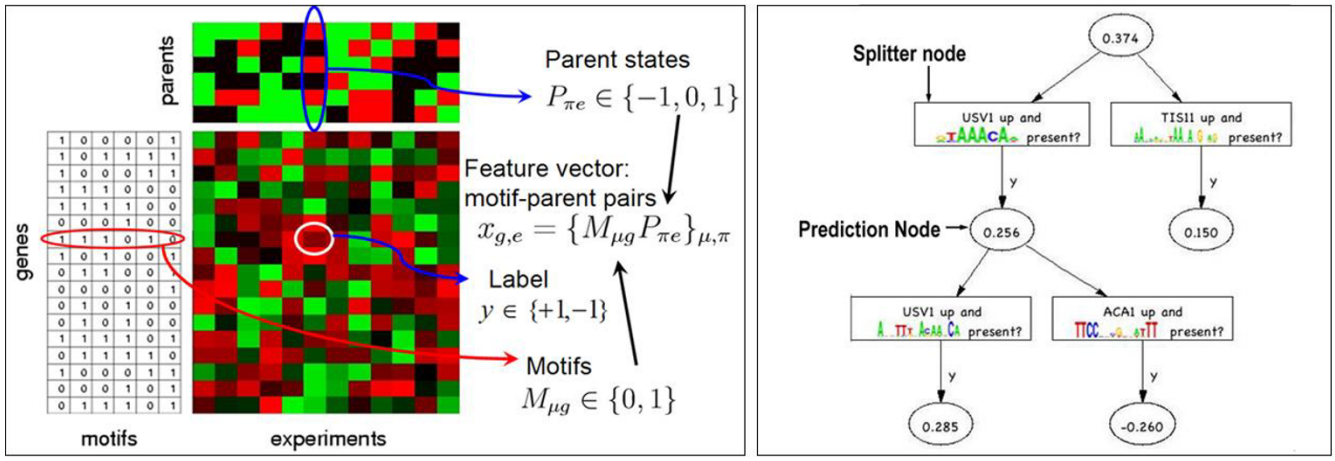


Figure 1

Overview of GeneClass algorithm. (Left) Data representation for input to the GeneClass algorithm. $P_{\pi e}$ represents the states of the parents π in each experiment e. $M_{\mu g}$ represents the presence/absence of motifs μ in the promoter of genes *g*. *y* represents the label of each example with feature vector $x_{g,e}$. (Right) The output of the algorithm is an alternating decision tree, which defines a prediction function for the up/down regulatory response of genes. Each splitter node represents a feature chosen by the algorithm, and the prediction node below it contains its prediction score.

Nrg2 transcription factors as identified by recent experiments [13].

Results

The Robust GeneClass algorithm

We apply the "robust" implementation of the GeneClass algorithm to study the regulatory response of the yeast *S. cerevisiae* under environmental stress and DNA damage conditions. GeneClass is a supervised learning algorithm, based on the boosting technique from machine learning, that learns a prediction function for the differential expression of any gene in an experimental condition given the vector of motif hits occurring in the gene's promoter region and the vector of mRNA expression levels of a set of regulators (transcription factors and signaling molecules) in the experiment. The input to the algorithm is a training set of gene expression experiments and the promoter sequences of all genes, together with a pre-determined candidate set of regulators and a set of putative transcription factor binding site motifs. The expression data is discretized into up (+1), down (-1), or baseline (0) expression levels using an expression-specific noise model (see Methods). The output is a prediction function in the form of an alternating decision tree (ADT). This function predicts up/down regulation of a gene-experiment example using a tree based on questions of the form, "Is motif *X* present in the upstream region of the gene and is the state of regulator *Y* up/down in that experiment?" Unlike regular decision trees, which make yes/no predictions, ADTs generate real-valued prediction scores whose sign gives the up/down prediction and whose size gives a measure of confidence in that prediction. An overview of the GeneClass algorithm, including the representation of the training data and a small example ADT, is given in Figure 1.

Given a training set of gene experiments, GeneClass learns a *single* prediction function (ADT) that can make predictions for *all* genes in a set of held-out experiments (test data). In our robust version of the algorithm, motif and regulator features chosen by the algorithm are more stable (see below), so that analysis of the learned ADT models reveals more confident information about gene regulation. We apply our postprocessing framework to extract detailed information about condition-specific regulation for individual target genes and groups of target genes.

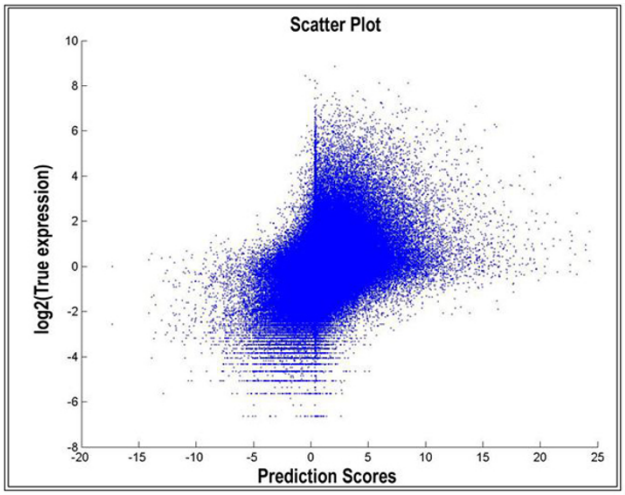
Datasets and experiments

Gene expression data

We use two datasets in our analysis: the environmental stress response (ESR) [14], consisting of 173 microarray experiments measuring the response in *S. cerevisiae* genome (6110 genes) to diverse environmental stresses; and the DNA damage dataset [15], consisting of 53 experiments measuring expression patterns (6167 genes) of wild-type and mutants cells. All measurements are given as $\log_2$ fold changes with respect to unstimulated reference expression.

Regulator set

We scale up our earlier formulation [11] by using a parent set of 475 regulators consisting of 237 known and putative transcription factors and 250 known and putative signaling molecules, with an overlap of 12 genes of



Confusion matrix

		Predicted Bins		
		Down	Baseline	Up
True Bins	Down	1.23%	0.66%	0.32%
	Baseline	6.84%	63.98%	22.25%
	Up	0.17%	1.40%	3.12%

Figure 2

Scatter plot and confusion matrix. (Left) The scatter plot shows the correlation between prediction scores (x-axis) and true log expression values (y-axis) for genes on held-out experiments. (Right) Confusion matrix: truth and predictions for all genes in the held-out experiments, including those expressed at baseline levels. Examples are binned by assigning a threshold of 0.95 for positive prediction scores and -0.2 for negative prediction scores.

unknown function. Of these, 466 are from Segal *et al*. [5] and 9 generic (global) regulators are obtained from Lee *et al*. [16].

Motif set

We use the motif matrix provided by Pilpel *et al*. [3]. The weight matrices for 356 known and putative transcription factor binding sites are derived using AlignACE [2] and matched to promoters of 5651 genes in the genome using ScanACE [2].

Occupancy data

Lee *et al*. [16] used genome-wide location analysis, based on modified chromatin immunoprecipitation (ChIP), to identify genomic binding sites for 113 transcription factors in living yeast cells under a single growth condition, using upstream regions of 6270 genes. For each genomic region, the transcription factor occupancy is reported as the log intensity ratio of the IP-enriched channel versus the genomic DNA channel, and a single array error model [16] is used to assign *p*-values to these measurements. We use the ChIP data as a binary "motif" matrix by thresholding the *p*-values, so that each target gene's motif vector is replaced by a ChIP occupancy vector for the set of transcription factors. We tried different thresholds of 0.001, 0.05 and 0.1 (results not shown) and found the best prediction accuracy with a *p*-value threshold of 0.1.

Target set

We extend the original GeneClass algorithm to use all target genes for which both motif and expression data is available. Table 2 shows the number of target genes used for different experiments (Gene lists may be found online at http://www.cs.columbia.edu/compbio/robust-geneclass[17]). We also use a new and improved technique based on the intensity-dependent noise distribution to discretize the expression data (see Methods).

Software and databases

All code is written in MATLAB and is downloadable from the supplementary website. [17] We use Graphviz [18] to visualize our trees. The Saccharomyces genome database and Incyte's Yeast proteome database are used for annotation analysis during post-processing.

Cross-validation and stabilization

In order to assess the predictive ability of our algorithm we perform 10-fold cross-validation on all genes present in the data set for 1000 boosting iterations. We divide the experiments into 10 random folds. We use each of the 10 folds as test sets while using the remaining 9 folds as training data to learn ADTs. We use the Pilpel [3] motif-matrix and the list of 475 known and putative parents. We average the test-loss (errors in prediction) over the 10 folds. The average test-loss on all genes is 16.2% ± 4.0% on the ESR data set and 23.4% ± 6.3% on the DNA damage data set. The same trees when tested on sets of selected target genes consisting of high-variance genes (standard deviation over experiments > 1.2) and genes that are part of clusters identified by hierarchical clustering analysis [14,15] (~1400 genes) give losses of 13.1% ± 2.4% and 13.3% ± 3.0%, respectively.

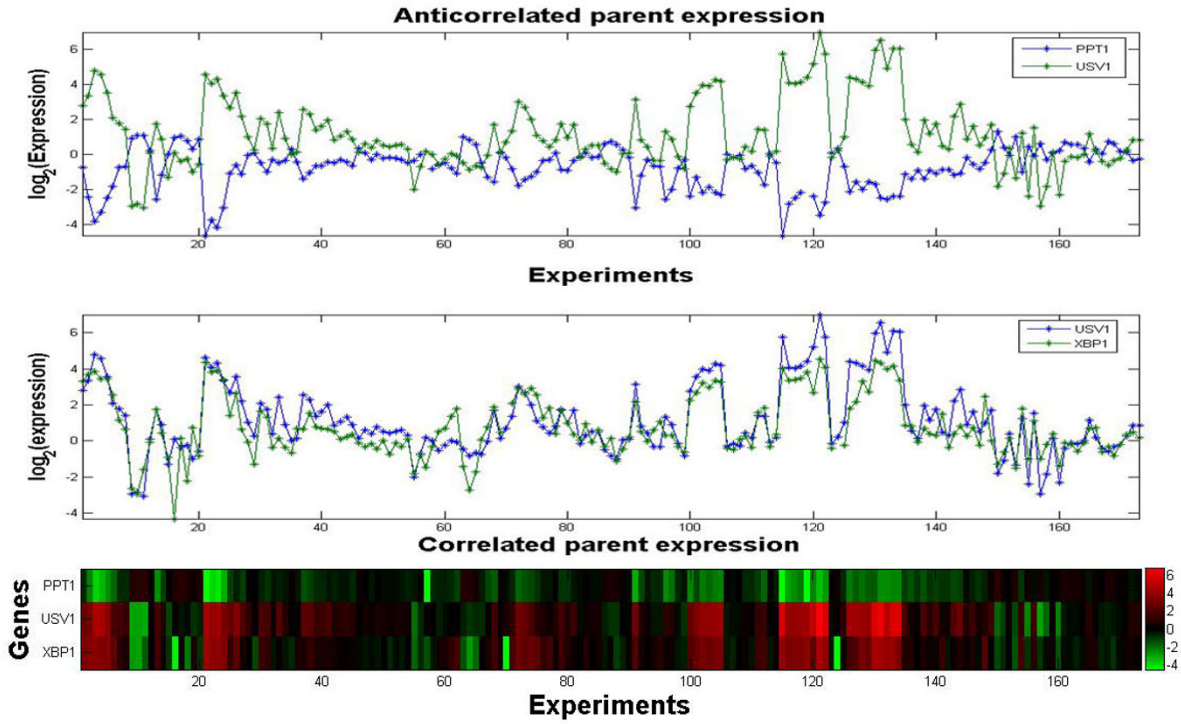


Figure 3
Stabilization: Correlated and anticorrelated features. The expression data from the ESR data set for regulators USV1, PPT1 and XBP1 are compared. All three are highly predictive parents. Without stabilization, they appear in different nodes of the learned trees. With stabilization, they are grouped together in robust nodes. Stabilization thus helps to preserve relevant information about correlated features, improving the biological interpretability during post-processing analysis.

Figure 2 shows expression values versus prediction scores for all examples (up, down, and baseline) from the held-out experiments using 10-fold cross-validation on the whole ESR data set. The correlation coefficient is 0.58 for +1 and -1 examples in the test set and 0.31 for all examples. While this correlation would not be considered high for a regression problem, it is significant in our current setting, since we do not use the true expression values or the baseline examples for training. By assigning thresholds to expression and prediction values, we bin the examples into up, down and baseline to obtain the confusion matrix in Figure 2. It is important to note that examples are labeled as baseline when they are within the replicate noise limits. These examples are not used in training since the confidence in the labels is low. However, the baseline examples predicted as +1 or -1 with high prediction scores could possibly indicate biologically meaningful differential expression within the levels of replicate noise.

In our previous work [11], at each boosting iteration we add only a single feature. In a case where several features are highly correlated with each other (e.g., parents having similar expression over experiments, or motifs that often co-occur in promoter regions), only one of them will be added to the tree, but all of them will be decorrelated after reweighting the examples. Hence, correlated features can appear in different nodes in the tree or predictive features could be missing altogether in the entire tree, a consequence that is inconvenient for extracting significant features during post-processing. Figure 3 shows the expression data for USV1, PPT1 and XBP1 in the ESR dataset. These happen to rank among the top 10 predictive regulators for this dataset. As is evident, USV1 and XBP1 are highly correlated whereas USV1 and PPT1 are anticorrelated. Stabilization tends to bring them together in robust nodes. Without stabilization, these regulators appear in different nodes in the tree, and it becomes difficult to recognize that the activities of these regulators are highly correlated.

We solve this problem in Robust Geneclass by adding a set of highly correlated weak rules. This improves the interpretability as well as stability of the ADT. Table 1 shows how Robust GeneClass stabilizes trees trained on different folds. We rank the parents (regulators) based on an iteration score (IS). The IS is the boosting iteration at which

Table 1: Stabilized trees. Top ten parents ranked by iteration score (IS) for alternating decision trees learned with and without stabilization. The stabilization uses parameters η_1 = 0.01 and η_2 = 0.03. (See Methods/Stabilization Section).

without stabilization			with stabilization		
rank	parent	iteration score	rank	parent	iteration score
1	TPK1	1.400 ± 1.265	1	TPK1	1.400 ± 1.265
2	USV1	3.500 ± 1.434	2	USV1	3.500 ± 1.434
3	AFR1	6.800 ± 3.360	3	AFR1	6.800 ± 3.360
4	ATG1	11.800 ± 20.099	4	ATG1	7.700 ± 7.747
5	MDG1	12.100 ± 11.090	5	MDG1	10.000 ± 9.369
6	XBP1	17.800 ± 6.460	6	XBP1	16.800 ± 5.287
7	ETR1	41.400 ± 24.972	7	CIN5	18.600 ± 7.604
8	YJL103C	45.000 ± 26.600	8	GIS1	20.600 ± 12.607
9	CIN5	56.100 ± 71.527	9	SDS22	20.900 ± 11.406
10	KIN82	57.800 ± 24.179	10	YFL052W	22.000 ± 6.815
11	GAT2	58.800 ± 55.249	11	YJL103C	22.200 ± 4.803
12	MSG5	61.700 ± 96.126	12	KIN82	22.400 ± 3.806
13	PDE1	65.200 ± 61.853	13	PDE1	22.800 ± 9.426
14	ASK10	68.300 ± 91.629	14	SIP4	22.900 ± 8.478
15	RME1	69.900 ± 23.572	15	ETR1	24.000 ± 3.771
16	YVH1	73.500 ± 24.865	16	GAC1	24.400 ± 4.142
17	MET28	86.300 ± 43.564	17	GAT2	25.100 ± 5.666
18	SDS22	86.800 ± 72.380	18	HAP4	25.900 ± 6.173
19	MTH1	91.900 ± 50.573	19	SIP2	26.000 ± 6.146
20	GPA2	92.800 ± 44.619	20	MTL1	26.000 ± 6.146

the parent first appears in the ADT (see Methods section for more details). We compare the 20 top-ranking parents by IS for 10-fold cross-validation runs with and without stabilization on the ESR data set. These lists are the result of the change in the tree structure due to changes in the training set by holding out different sets of experiments. The standard deviation in IS over folds decreases by up to a factor of 10. The ordering is affected especially for lower-ranking parents (rank > 6). By including more complete information about predictive features, we obtain more stable and interpretable trees.

Table 2 shows the specific effects of the new discretization scheme, abstaining and stabilization on the prediction accuracy. We perform 10-fold cross-validation on the complete ESR dataset as well as the reduced set of 1411 targets with and without these three new algorithmic changes. The table shows that the new discretization

Table 2: Effects of abstaining, stabilization and discretization on prediction accuracy.

No. of targets	No. of Iterations	Discretization	Abstain	Stabilize	Mean test-loss
1411	400	Old	No	No	11.50%
1411	400	Old	Yes	No	16.01%
1411	400	New	Yes	No	13.13%
1411	400	New	Yes	Yes	13.50%
5579	1000	Old	No	No	14.00%
5579	1000	Old	Yes	No	18.99%
5579	1000	New	Yes	No	15.80%
5579	1000	New	Yes	Yes	16.10%

The table shows different experimental setups to test the effects of abstaining, stabilization, and the new discretization technique. All the experiments involve 10-fold cross-validation on the ESR data set. Column 1 represents the number of target genes used. Column 2 indicates the number of iterations. Column 3 specifies the discretization technique used. Columns 4 and 5 specify whether abstaining and stabilization were used. Column 6 reports the average test loss. The first and fifth experiments are the original GeneClass algorithm. The fourth and eighth experiments are the Robust GeneClass algorithm with all three algorithmic improvements. Comparing rows 2, 3 and 6, 7 we see that the use of new discretization technique alone improves test loss. Comparing rows 3, 4 and 7, 8 we see that stabilization alone does not have any significant effect on test loss. Comparing rows 1, 2 and 5, 6 we observe that abstaining increases test loss. The combined effect of the three algorithmic improvements results in a small increase in test loss due to the somewhat poorer accuracy of abstaining weak rules; however, abstaining improves the interpretability of the model and significantly speeds up training time.

scheme leads to a marked improvement (~3%) in the test error over using a simple fold change cut-off. The prediction accuracy is not significantly affected due to stabilization. This result is expected since stabilization basically prevents the algorithm from missing correlated features and helps to retain biologically relevant features in the tree. Abstaining on the other hand does cause an increase (~4.5%) in the test error. However, abstaining leads to more interpretable and shallower trees by avoiding long paths of "yes" and "no" edges; individual paths in the prediction tree are also shorter and more statistically significant with abstaining. Moreover, by adding a single node rather than a pair of yes/no nodes at each iteration of boosting, the search space of nodes is only half as large, and overhead associated with the nodes is also reduced. We find that abstaining leads to a 4-fold reduction in running time on the ESR dataset for 1000 iterations.

To assess the difficulty of the classification task, we also compare to a baseline method, *k*-nearest neighbor classification (kNN), where each test example is classified by a vote of its *k* nearest neighbors in the training set. For a distance function, we use a weighted sum of Euclidean distances $d((g_1, e_1), (g_2, e_2))^2 = w_m || \mathbf{m}_{g_1} - \mathbf{m}_{g_2} ||^2 + w_p || \mathbf{p}_{e_1} - \mathbf{p}_{e_2} ||^2$, where $\mathbf{m}_g$ represents the vector of motif counts for gene g and $\mathbf{p}_e$ represents the parent expression vector in experiment e. We try various weight ratios $10^{-3} < (w_m/w_p) < 10^3$ and values of $k < 20$, and we use both binary and integer representations of the motif data. We obtain minimum test loss of 25.5% for the whole ESR data set at k = 15 for integer motif counts using a weight-ratio of 1, giving much poorer performance than boosting with ADTs (test loss of 16.2%).

Motif data versus CHIP chip data

In order to study different aspects of target gene regulation we use different sets of motifs and parents with the GeneClass algorithm. Since these experiments are primarily used for post-processing analysis, we use a single random held-out test set consisting of one tenth of the non-zero labeled (gene, experiment) pairs in order to get an estimate of test error. The experiments are as shown in Table 3. In the following sections we shall refer to these experiments by their abbreviated names.

Post-processing case study

We use the post-processing framework on individual genes, sets of genes and regulators for a case study related to heat shock genes and protein folding chaperones as well as the heat shock transcription factor HSF1. (The complete gene list and analyses of additional gene sets can be downloaded from the supplementary website [17].) Unless otherwise mentioned, all analysis is on the ESR dataset [14].

Heat shock genes, protein folding chaperones and HSF1

It has been observed that a subset of around 26 protein folding chaperones is induced by a variety of stress conditions [14]. The Hsf1p transcription factor along with Msn2p/Msn4p are known to be the prime regulators for this set of genes. We first use the group target gene analysis framework to analyze this set of genes across all 173 ESR experiments. We analyze specific relevant experiments — MSN2/4 deletion mutants, MSN2 and MSN4 over-expression mutant, YAP1 deletion mutants and YAP1 over-expression mutant — as well as studying regulatory phenomena in opposite polarities of heat shock, i.e. temperature increase versus temperature decrease (see Figure 4).

Group target gene analysis: Protein folding chaperones

We start with a global analysis of the protein folding chaperones in all experiments. We rank the motif-regulator pairs using the frequency score (FS). The FS of a feature is the number of target gene-experiment examples that satisfy that feature (see Methods for details).

In both the *ChIP+all* and *Pilpel+all* experimental setups, we find CMK2 and SLT2 among the top scoring features as parents. Slt2p is the terminal MAPKinase in the PKC pathway and is known to be involved in regulating response to heat shock, hypo-osmotic shock, polarized cell growth and response to nutrient availability [19]. In both experimental setups, SLT2 is found associated in a motif-parent pair with the HSF1 binding site, indicating that HSF1 may be a target of the PKC pathway in many of these stresses. CMK2 is also found associated with the HSF1 "motif" (binding occupancy) in the *ChIP+all* setup. In mamma-

Table 3: Different experimental setups and their performance

Experiment	Motif data	Parents	Targets	Error on ESR	Error on DNA damage
ChIP+all	ChIP	475 SM+TF	6102	17.29%	23.56%
ChIP+SM	ChIP	250 SM	6102	16.7%	23.33%
Pilpel+all	Pilpel motifs	475 SM+TF	5579	12.7%	18.87%
Pilpel+TF	Pilpel motifs	237 TF	5579	13.7%	20.34%
Pilpel+SM	Pilpel motifs	250 SM	5579	14.11%	19.58%

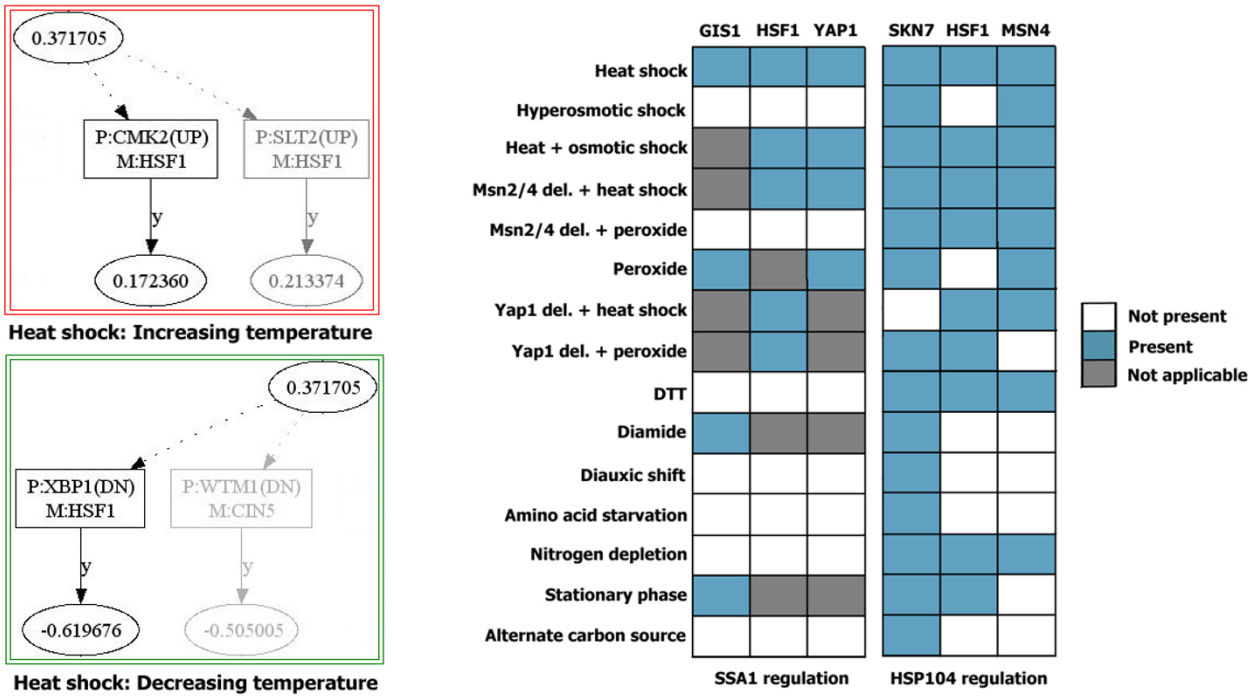


Figure 4

Heat shock protein analysis. (Left) Comparison of trees for response of heat shock proteins to increasing and decreasing heat shock. It is interesting to note that XBP1 and WTM1 are repressors. (Right) Condition-specific regulation of SSA1 and HSP104. The grey squares indicate that the target gene was in the baseline state for those experiments.

lian cells, CaMKII which is an ortholog of CMK2 has been found to significantly affect HSF1 function, [20] and association between CMK2 and the HSF1 motif might indicate a similar relationship in yeast. Other high scoring parents include USV1 and TPK1. The high scoring MSN4 motif is found to be associated with SLT2 and PTP2, the latter being part of the HOG MAPKinase pathway. It is interesting to note that Ptp2p can inactivate Slt2p via phosphorylation and also that a Ptp2 mutant has been found to be hyper-sensitive to heat [21]. MSN4 could thus be a downstream target of pathways involving these signaling molecules. The (TPK1 parent, SKN7 occupancy) pair is also found to be high scoring. Skn7p has a DNA binding domain homologous to that of Hsf1p and is considered to be an integrator of signals from various MAPKinase pathways. It is once again interesting to note that neither HSF1 nor MSN4 are high scoring parents: their mRNA levels do not vary significantly in the dataset, and their activity is primarily controlled post-transcriptionally and by cellular localization. Thus, the mRNA levels of these transcription factors do not appear to be predictive of their targets. However, their motifs/DNA-binding profiles are found to be strongly predictive.

We also use the group target gene analysis framework in sets of experiments consisting of specific stresses, again ranking features by FS, to examine phenomena unique to these stress responses.

Figure 5 shows the parents and motifs that are predictive of the differential expression of the heat shock genes in the simultaneous heat and osmolarity shock experiments. The predictive parents and motifs are the same as the ones found in the global analysis of these targets in all experiments. However, in the alternate carbon source response and diauxic shift experiments, we specifically find the (SNF3 parent, HAP4 occupancy) pair to be the highest scoring feature in the *ChIP+all* setup. Snf3p is part of the glucose sensor family and the Hap4p is a transcription factor involved in regulating growth in non-fermentable carbon sources [22]. Similarly, the (PTP2 parent, MSN4 occupancy) feature is particularly prominent in the hyperosmotic stress indicating possible activation of the HOG1 pathway. Skn7p and Hsf1p have been shown to induce several heat shock proteins in response to oxidative stress [23]. We observe this phenomenon in the features extracted for response to peroxide, DTT and diamide. For the starvation responses, which are unique in that the cells undergo permanent cell-cycle arrest, we see the emergence of CLB2 and CDC5 as high scoring parents in the *Pilpel+all* setup. Both these regulators control exit from mitosis, and CLB2 (necessary for G2 repression of the SCB factor) is

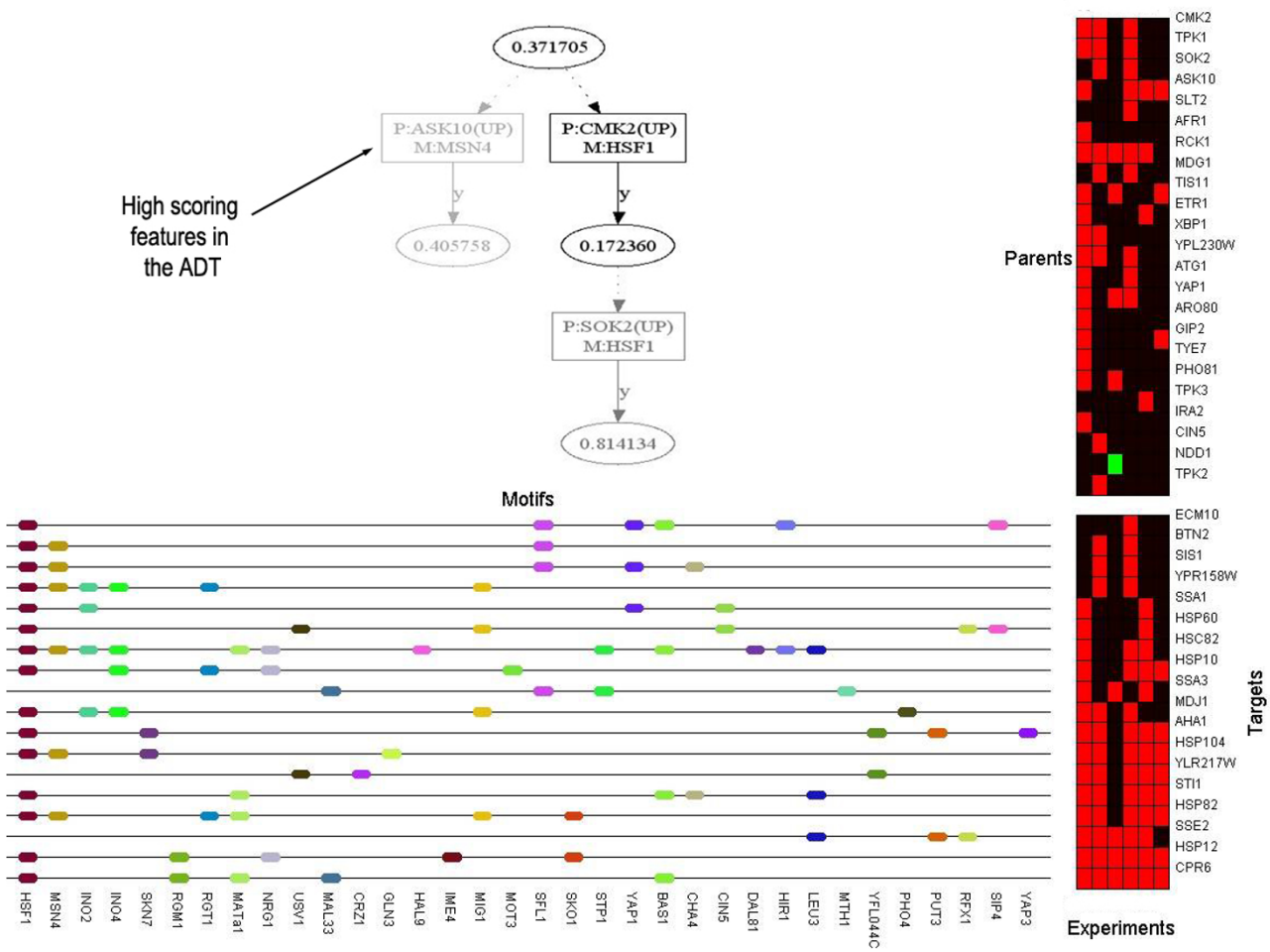


Figure 5
Group target gene analysis. The figure shows the regulatory motifs and parents that are predictive of the expression of the heat shock proteins in the simultaneous heat and osmolarity shock experiments. The bottom right rectangle represents the discretized expression of the targets in the experiments under study. Red represents +1 (up regulation). Green represents -1 (down regulation). Black represents 0 (baseline). The top right rectangle shows the expression of the predictive parents. The parents are ordered from top to bottom in decreasing order of frequency score (number of examples that pass through nodes containing the parent). The bottom left illustration represents the upstream regulatory promoter regions of the target genes. The motifs are arranged in decreasing order of frequency score from left to right. A reduced section of the subtree with the top 3 predictive features is also shown. The intensity of the nodes (in gray scale) reflect the frequency scores. Darker nodes have higher frequency scores.

found to be associated with the SCB motif. This finding is clear evidence that, in addition to global regulatory mechanisms, we are also able to extract important *context-specific* regulatory features.

Individual target gene analysis: Protein folding chaperones

We perform individual target gene analysis for two heat shock proteins, SSA1 and HSP104. We use the target gene analysis framework with the *ChIP+allreg* setup and observe interesting aspects of condition-specific regulation, summarized in Figure 4.

We look for high scoring HSF1 and MSN4 motifs to account for activity of these factors. SSA1 seems to be independent of MSN4 in all stresses. It appears to be primarily regulated by HSF1 and YAP1 in heat shock, simultaneous heat and osmotic shock, YAP1 deletion mutant exposed to heat shock and YAP1 deletion mutant exposed to peroxide. YAP1 seems to have exclusive control in the peroxide response while GIS1 is found to be the key regulator (parent) in the diamide response and stationary phase response. Gis1p is known to regulate some heat shock proteins [24]. It is not known if Gis1p binding is

dependent on Hsf1p binding. It appears from our analysis that the two might be independent at least in the case of SSA1. HSP104 has Msn4p, Hsf1p and Skn7p binding sites in its upstream region and appears to be actively regulated by these transcription factors in a stress specific manner. Skn7p, Msn4p and Hsf1p appear to jointly control regulation in almost all stresses. The exceptions are hyper-osmotic stress and peroxide stress where only Skn7p and Msn4p seem to be active and the response to stationary phase induction where Skn7p and Hsf1p seem to be active.

Regulator and signaling pathway analysis: Protein folding chaperones

We also study the HSF1 regulator and extract the signaling molecules that are predictive of its activity and its targets. We use the *ChIP+SM* setup and extract all signaling molecules that associate with HSF1 occupancy in the entire tree. We find an important section of the PKA signaling pathway (TPK1, BCY1, PDE1, YAK1) as well as parts of the PKC pathway (WSC4, SLT2 and CDC28). We also find SDS22, GIP2 and GAC1, all of which are sub-units of the Glc7p protein phosphatase, which has been identified as an Hsf1p binding protein [25].

Group target gene analysis of putative Nrg1/Nrg2 targets

The transcriptional repressors Nrg1p and Nrg2p (Nrg1/Nrg2) have been previously implicated in glucose repression. Recent genome-wide expression analysis experiments by Vyas *et al*. [13] identify 150 genes that are up-regulated in NRG1/NRG2 double mutant cells, relative to wild-type cells, during growth in glucose. These genes are involved in mito-chondrial function, carbon utilization and signaling, nitrogen utilization and pseudophyphal growth, cell wall and cell surface function, transcriptional control, mating, transport of nutrients and ions, and other cellular processes. They further show that Nrg1 and Nrg2 might affect a larger set of stress response genes.

Using the group target gene analysis framework (*ChIP+all* setup), we analyze the regulatory machinery that is predictive of the expression of these genes in the entire ESR dataset. We extract the subtree corresponding to the expression of these 150 target genes in all 173 experiments and rank the parents and motifs by frequency score. 228 parents and 100 motifs appear in the subtree. Interestingly, NRG1 appears at rank 14 among the parents and at rank 17 among the motifs with high frequency scores. This strongly supports the hypothesis that these genes could be regulated by Nrg1p. The expression of NRG2 does not change significantly in most of the experiments, and the binding sites for Nrg1p and Nrg2p are very similar. These reasons could explain why the latter does not appear in the subtree.

Previous studies have implicated Nrg1p and/or Nrg2p in glucose repression, nitrogen starvation response and pseudohyphal growth, adaptation to alkaline pH, and ion tolerance. Our analysis finds that the other top scoring parents and motifs have similar functions. CUP1A and CUP1B, which are involved in Cu ion homeostasis, are among the top 10 motifs and parents. Other regulators with strongly predictive expression profiles include TPK1, CMK2, MDG1, USV1, GIS1 (stress response regulators), MTH1 (carbohydrate metabolism, hexose transport), XBP1 (pseudohyphal growth) and PPZ2 (K+ ion and pH homeostasis). The high scoring predictive motifs are those of RGM1 (transcriptional repressor), MAL13, MAL33 (maltose catabolism), GPT2, INO2, INO4 (phospholypid biosynthesis), DAL82, GAT1 (nitrogen metabolism), STE12, SKN7 (pseudohyphal growth), ECM22, GPT2, CBF1, CIN5 (drug, ion and osmotic stress response). A figure on the supplementary website [17] illustrates the regulatory mechanisms in more detail.

Discussion & Conclusion

Our work on the Robust GeneClass algorithm is motivated by two important challenges in learning models of transcriptional gene regulation from high throughput data. The first challenge is to find a favorable trade-off between the statistical validity of the model — most convincingly measured by its ability to generalize to test data — and biological interpretability. Clearly, an interpretable model that overfits the training data is not meaningful, while a fully "black box" prediction rule, however accurate its generalization performance, tells us little about biology. The second challenge is to capture condition-specific rather than static models of regulation. A model based on partitioning genes into static clusters, for example, fails to address the fact that under different conditions, a gene could be controlled by different regulators and share transcriptional programs with different sets of target genes.

Most work on modeling gene regulation has focused on the problem of learning interpretable structure and placed less emphasis on quantifying how well the models generalize. The most popular structure-learning approach, probabilistic graphical models, can certainly be used to make predictions in various ways and can generalize well in the presence of sufficient training data. However, since both the underlying regulatory mechanisms and the probabilistic model trying to represent them are complex, and since training data is limited, it is critical to demonstrate the statistical validity of the learned structure, or at least to investigate how much of the structure is robust to noise or small perturbations in the data. For example, the Bayesian network-based MinReg algorithm [10] has been shown to improve the probability of correct target gene state prediction in cross-validation over a clustering approach, and

bootstrapping has been used to extract robust subnetworks in Bayesian network learning [9]. More prevalent use of statistical validation of these kinds is essential to assess progress in modeling efforts.

In the GeneClass approach, we formulate gene regulation as a binary prediction problem (i.e. predicting up/down regulatory response of target genes), and we demonstrate very strong predictive performance on test data. Our main goal in Robust GeneClass is not to improve prediction accuracy, which is already good, but to increase the stability of features included in the prediction tree to enable the detailed target gene analysis that we present in our post-processing framework. As we show in Table 1, our stabilization technique greatly improves the robustness of the ranked list of features added to the model. Improved stability allows the reliable analysis of subtrees corresponding to specific target genes or experiments, giving more meaningful biological interpretation. We have also improved our discretization approach, which does slightly improve test accuracy. One technical change we have incorporated — the use of abstaining weak rules — is specifically intended to improve interpretability of the prediction tree. Abstaining makes the trees and subtrees shallower and easier to understand and makes individual paths shorter and more statistically significant. However, abstaining does weaken test accuracy by a small but significant amount. In future work, it might be useful to revisit the choice of weak rules, since it appears that the richer combinatorial interaction of regular (yes/no) weak rules is a slightly better predictive model. Nonetheless, the accuracy/interpretability trade-off in Robust GeneClass allows us to extract interpretable and stable subtrees for target gene analysis, enabling a more sensitive, detailed, and biologically relevant study of gene regulatory response.

The second modeling challenge that we address in this work is the issue of capturing condition-specific regulation. The GeneClass approach learns a single predictive model for all target genes based on the presence of binding site motifs in the promoter sequence and the activity of regulators in the experiment. However, different paths of prediction tree affect different targets under different conditions, as represented by the state of the regulators. In this way, the GeneClass model naturally captures condition-specific regulation, though the original work presented only global feature analysis. The post-processing method described in the current work addresses condition-specific regulation by extracting and analyzing subtrees corresponding to related sets of experiments.

Robust GeneClass also incorporates computational improvements that allow us to scale to larger problem sizes, using all regulators in yeast as the candidate parent set and using all genes as targets. In addition, we present results based on using transcription factor occupancy as measured by ChIP chip assays to replace binding site data, and in examples of our post-processing framework for target gene, we also perform simple signaling pathway analysis. In other work, we have extended the GeneClass model to incorporate motif discovery with the MEDUSA algorithm [26], and we are investigating ways to use protein-protein interaction data to better model and retrieve signaling pathways. We anticipate that the predictive modeling methodology that we develop here will become a valuable new approach for gaining biological insight from high throughput genomic data sources.

Methods

The GeneClass algorithm

The GeneClass algorithm for predicting differential gene expression starts with a candidate set of *motifs* μ representing known or putative regulatory element sequence patterns and a candidate set of regulators or *parents* π. For each (gene, experiment) example in our gene expression dataset, we have two sources of feature information relative to the candidate motifs and candidate parent sets: a bit vector $M_{\mu g}$ of motif occurrences of patterns μ in the regulatory sequence of gene g, and the vector $P_{\pi e} \in \{-1, 0, 1\}$ of expression states for parent genes π in the experiment e.

GeneClass uses the AdaBoost algorithm to iteratively build an alternating decision tree (ADT) based on motif-parent features. Adaboost is a general algorithm for binary prediction problems, where the training set consists of pairs (x_1, y_1), ..., (x_m, y_m), x_i corresponds to the features of an example, and $y_i \in \{-1, +1\}$ is the binary label to be predicted. In our case, training examples x_i are gene-experiment instances, represented by the motif occurrence and parent expression features, and labels y_i represent whether the gene is over-expressed (+1) or under-expressed (-1) in the experiment. An ADT is a margin-based generalization of a decision tree that consists of alternating layers of splitter nodes, which ask yes/no questions based on a particular feature, and prediction nodes, which contain a real-valued score associated with the yes or no answer. Boosting works in rounds and maintains a weight distribution over training examples, i.e. an assignment of a non-negative real value W_i to each example (x_i, y_i). At each round of boosting, the ADT algorithm selects a (μ, π) feature and adds a splitter node — based on a boolean condition such as "motif μ is present and regulator π is over-expressed (or under-expressed)" — and an associated prediction node. The feature is chosen to optimize a loss function Z (see below) that depends on the current weighting of the examples. The weighting is then updated in order to focus on training examples on which the current ADT model still performs poorly. The final prediction score for a (gene, experiment) example is the sum of all prediction nodes in all paths in the ADT that are consistent with the

example. (See our original paper [11] for more algorithmic details.)

Computational and statistical improvements

In the following, $M_{\mu g}$ denotes the binary motif matrix (indicating the presence of motif μ in the promoter region of gene g), and $P^{\uparrow}_{\pi e}$ ($P^{\downarrow}_{\pi e}$) the binary parent-state matrix (indicating an up- or down-regulated state of parent π in experiment e). The features are pairs (μ, π^s), $s \in \{\uparrow, \downarrow\}$, while examples are pairs (g, e). $\mathcal{W}_{ge}$ is the matrix of all boosting weights at a given iteration. We define $W^{+}_{\mu\pi^s}$ ($W^{-}_{\mu\pi^s}$) to be the sum of weights over up- (down-) examples having feature (μ, π^s) and $W^{0}_{\mu\pi^s} \equiv 1 - W^{+}_{\mu\pi^s} - W^{-}_{\mu\pi^s}$.

Faster computation of boosting statistics

The Adaboost loss function [27] $Z(\mu, \pi^s) = W^{0}_{\mu\pi^s} + 2\sqrt{W^{+}_{\mu\pi^s} W^{-}_{\mu\pi^s}}$, for weak learners predicting in $\{0, 1\}$, can be calculated for outer-product features (μ, π^s) in an efficient way using sparse matrix multiplication. For example, $W^{+}_{\mu\pi^{\uparrow}}$ the weight of all examples with label +1 having feature $(\mu, \pi^{\uparrow})$, can be computed as $W^{+}_{\mu\pi^{\uparrow}} = \sum_{ge} M_{\mu g} \mathcal{W}_{ge} I^{+}_{ge} P^{\uparrow}_{\pi e}$, where I^{+}_{ge} is a binary matrix indicating the examples (g, e) having positive labels. This efficiency improvement allows us to scale up to the full set of regulators and include expression data for all genes as training examples.

Abstaining

As a consequence of the sparseness of $M_{\mu g}$ and $P^{\uparrow\downarrow}_{\pi e}$, the no-answer (absence of the feature) for a given splitter node is true for the predominant part of the data. By abstaining from predicting "no", trained ADTs become shallower, and specific paths in the tree are statistically more significant and more easily intepretable in biological terms. Also, since at each iteration a single prediction node rather than a "yes" and "no" pair of nodes is added to the tree, the search space at each iteration is half as big as before.

Stabilization

At each iteration boosting adds the weak rule with the smallest loss Z. The training examples are then reweighted such that they become decorrelated with the previously added rule. As discussed in the Results section, this leads to stability and interpretability issues.

We solve these problems by averaging the prediction of several weak rules in the case where the rules with smallest loss Z are highly correlated with each other. We determine whether the empirical correlation is statistically significant by comparing it with a threshold which is is a function of the weights of the examples used for choosing the rules. The function we use is motivated by Hoeffding bounds [28]. As shown in the Results section, this scheme stabilizes the trees trained on different folds.

At every boosting iteration we consider the overlap of each feature with the feature of smallest boosting loss Z; i.e., the total weight of the examples with both features equal to 1. We average over those features for which the overlap is smaller than $\eta_1 \sqrt{\sum_{ge} \mathcal{W}^2_{ge}}$, where η_1 is an empirically chosen parameter. The dependence on the weights $\mathcal{W}_{ge}$ is motivated by Hoeffding bounds [28,29]. In addition, our algorithm abstains from stabilization if the advantage of the weak rules over random-guessing is so small that the interpretability of the selected features becomes questionable. The stabilization is thus skipped if the difference of the weighted loss and 1/2 is smaller than $\eta_2 \sqrt{\sum_{ge} \mathcal{W}^2_{ge}}$, where η_2 is a second empirically chosen parameter. For all runs in this paper we used $\eta_1 = \eta_2 = 0.1$.

Detailed pseudo-code for this algorithm is available on the supplementary website [17].

Improved noise model

In earlier work [11], we used a simple noise model to discretize the gene expression data into three levels — down-regulation (-1), up-regulation (+1), and no significant change beyond noise levels (0) or baseline — based on the empirical noise distribution around the baseline (0). We now extend the noise model to account for intensity specific effects using the raw data from both the Cy3(R) and Cy5(G) channels. In order to estimate the null model, we use a control experiment [15] for the DNA damage dataset and the three replicate unstimulated heat-shock experiments for the ESR dataset [14]. The $\mathrm{M}\left(= \log_2\left(\frac{R}{G}\right)\right)$ versus $\mathrm{A}(= \log_2(\sqrt{RG}))$ plots (Figure 6) show the intensity specific distribution of the noise in the log-ratios. We compute the cumulative empirical null distribution of M conditioned on A by binning the A var-

iable into small bin sizes, maintaining a good resolution while having sufficient data points per bin. For any expression value (M, A) of a gene in an experiment, we estimate a p-value based on the null distribution conditioned on A, and we use a p-value cutoff of 0.05 to discretize the expression values into +1, -1 or 0 (see supplementary website [17] for details).

Post-processing

In our previous work [11], we used basic scoring metrics — namely the abundance score (AS: the number of times a particular motif, regulator or motif-parent pair occurs in the tree) and the iteration score (IS: the earliest iteration at which a feature occurs in the tree) — to rank features in the full learned ADT, obtaining a global view of various stress regulatory responses. However, since we build a single predictive model for regulation in (gene, experiment) examples, we can restrict to the regulation program for a particular target gene or set of genes in a particular experiment or a set of experiments, giving a detailed and local view.

Individual and group target gene analysis

To consider a gene or group of genes in a single experiment, we extract all paths in the ADT whose splitter nodes evaluate true for the (gene, experiment) pairs in question. We then rank motifs, parents and motif-parent pairs using AS and IS in the extracted subtree.

When considering target genes in multiple experiments, we also use the *frequency score* (FS), defined as the number of times any target gene passes through a splitter node containing the feature in all the experiments for which the gene's label is correctly predicted. This technique is useful for identifying regulators and motifs that are actively regulating the target genes in different conditions.

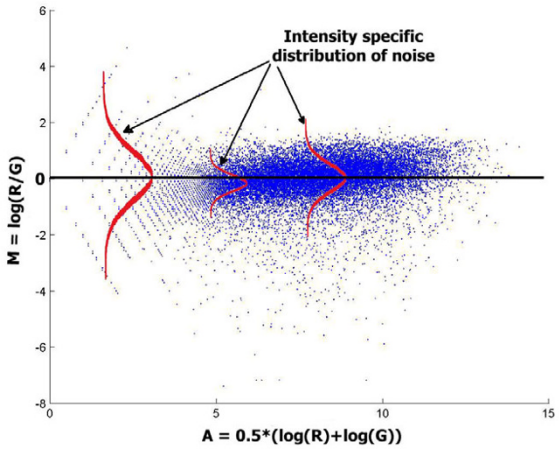


Figure 6
Improved noise model. We use an expression specific noise model to discretize gene expression data.

Signaling pathways and regulator analysis

Different signaling pathways are activated under different stress conditions, and these highly interconnected pathways affect regulation via activation or repression of sets of transcription factors. Since many kinases are auto-regulated or are in tight positive and/or negative feedback mechanisms with the transcription factors that they regulate [14], we hypothesize that mRNA levels of signaling molecules in particular pathways might be predictive of expression patterns of targets genes of downstream transcription factors. First, we use individual target gene analysis to study regulators that are predictive of the mRNA of other regulators (including regulators in the target gene set). Second, we use ChIP data [16] in place of motif data — representing the binding potential of a target gene's regulatory sequence by a bit vector of transcription factor occupancies rather than a motif bit vector — and then study the signaling molecules that associate with the motif in high scoring features.

Authors' contributions

Anshul Kundaje performed the post-processing analysis described in the Results and helped to run the computational experiments. Manuel Middendorf implemented Robust GeneClass, helped to develop the stabilization technique for the algorithm, and helped with the computational experiments. Mihir Shah assisted with code implementation for the postprocessing analysis. Chris Wiggins helped to supervise the research and suggest experiments. Yoav Freund helped to supervise the research, provided technical advice on the ADT algorithm, and proposed the stabilization technique. Christina Leslie supervised the research and helped to design experiments and direct technical developments of the algorithm. The manuscript was drafted by Anshul Kundaje, Manuel Middendorf, and Christina Leslie.

Acknowledgements

This work was partially supported by NSF grants ECS-0332479 and ECS-0425850 and NIH grants GM36277 and LM07276-02. We thank Marian Carlson and Valmik Vyas for generously sharing their data and results with us.

References

1. Bussemaker HJ, Li H, Siggia ED: **Regulatory element detection using correlation with expression.** *Nature Genetics* 2001, **27:**167-171.
2. Hughes JD, Estep PW, Tavazoie S, Church GM: **Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae.** *J Mol Biol* 2000, **296(5):**1205-14.
3. Pilpel Y, Sudarsanam P, Church GM: **Identifying regulatory networks by combinatorial analysis of promoter elements.** *Nature Genetics* 2001, **2:**153-159.

4. Ihmels J, Friedlander G, Bergmann S, Sarig O, Ziv Y, Barkai N: **Revealing modular organization in the yeast transcriptional network.** *Nature Genetics* 2002, **31:**370-377.
5. 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.
6. Beer MA, Tavazoie S: **Predicting gene expression from sequence.** *Cell* 2004, **117(2):**185-98.
7. Segal E, Yelensky R, Koller D: **Genome-wide Discovery of Transcriptional Modules from DNA Sequence and Gene Expression.** *Bioinformatics* 2003, **19:**273-282.
8. Hartemink AJ, Gifford DK, Jaakkola TS, Young RA: **Using Graphical Models and Genomic Expression Data to Statistically validate Models of Genetic Regulatory Networks.** *Pac Symp Biocomp* 2001:422-33.
9. Pe'er D, Regev A, Elidan G, Friedman N: **Inferring subnetworks from perturbed expression profiles.** *Proc of the Ninth International Conf on Intelligent Systems for Molecular Biology* 2001:215-224.
10. Pe'er D, Regev V, Tanay A: **A Fast and Robust Method to Infer and Characterize and Active Regulator Set for Molecular Pathways.** *Proc of the Tenth International Conf on Intelligent Systems for Molecular Biology* 2002:258-267.
11. Middendorf M, Kundaje A, Wiggins C, Freund Y, Leslie C: **Predicting Genetic Regulatory Response using Classification.** *Proceedings of the Twelfth International Conference on Intelligent Systems for Molecular Biology (ISMB 2004)* 2004 [http://www.cs.columbia.edu/compbio/geneclass].
12. Middendorf M, Kundaje A, Wiggins C, Freund Y, Leslie C: **Predicting Genetic Regulatory Response using Classification: Yeast Stress Response.** *Proceedings of the First Annual RECOMB Regulation Workshop* 2005. [Lecture Notes in Bioinformatics]
13. Vyas VK, Berkey CD, Miyao T, Carlson M: **Repressors Nrg1 and Nrg2 Regulate a Set of Stress-Responsive Genes in Saccharomyces cerevisiae.** *Eukaryotic Cell* 2005 in press.
14. 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.** *Mol Biol Cell* 2000, **11:**4241-4257.
15. 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.** *Mol Biol Cell* 2001, **12(10**2987-3003 [http://www.molbiolcell.org/cgi/content/abstract/12/10/2987].
16. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, Hannett NM, Harbison CR, Thompson CM, Simon I, Zeitlinger J, Jennings EG, Murray HL, Gordon DB, Ren B, Wyrick JJ, Tagne J, Volkert TL, Fraenkel E, Gifford DK, Young RA: **Transcriptional Regulatory Networks in Saccharomyces cerevisiae.** *Science* 2002, **298:**799-804.
17. Kundaje A, Middendorf M, Shah M, Wiggins C, Freund Y, Leslie C: [http://www.cs.columbia.edu/compbio/robust-geneclass]. [Web supplement]
18. Gansner ER, North SC: **An open graph visualization system and its applications to software engineering.** *Softw Pract Exper* 2000, **30(11):**1203-1233.
19. Zarzov P, Mazzoni C, Mann C: **The SLT2(MPK1) MAP kinase is activated during periods of polarized cell growth in yeast.** *EMBO J* 1996, **15:**83-91.
20. Holmberg CI, Hietakangas V, Mikhailov A, Rantanen JO, Kallio M, Meinander A, Hellman J, Morrice N, MacKintosh C, Morimoto RI, Eriksson JE, Sistonen L: **Phosphorylation of serine 230 promotes inducible transcriptional activity of heat shock factor 1.** *EMBO J* 2001, **20(14):**3800-3810.
21. Ota I, Varshavsky A: **A Gene Encoding a Putative Tyrosine Phosphatase Suppresses Lethality of an N-End Rule-Dependent Mutant.** *PNAS* 1992, **89(6**2355-2359 [http://www.pnas.org/cgi/content/abstract/89/6/2355].
22. Ramil E, Agrimonti C, Shechter E, Gervais M, Guiard B: **Regulation of the CYB2 gene expression: transcriptional co-ordination by the Hap1p, Hap2/3/4/5p and Adr1p transcription factors.** *Mol Microbiol* 2000, **37(5):**1116-32. http://www.blackwell-synergy.com/links/doi/10.1046/j.1365-2958.2000.02065.x/abs
23. Raitt DC, Johnson AL, Erkine AM, Makino K, Morgan B, Gross DS, Johnston LH: **The Skn7 Response Regulator of Saccharomyces cerevisiae Interacts with Hsf1 In Vivo and Is Required for the Induction of Heat Shock Genes by Oxidative Stress.** *Mol Biol Cell* 2000, **11(7):**2335-2347.
24. Pedruzzi I, Burckert N, Egger P, Virgilio CD: **Saccharomyces cerevisiae Ras/cAMP pathway controls post-diauxic shift element-dependent transcription through the zinc finger protein Gis1.** *EMBO J* 2000, **19(11):**2569-2579.
25. Lin JT, Lis JT: **Glycogen Synthase Phosphatase Interacts with Heat Shock Factor To Activate CUP1 Gene Transcription in Saccharomyces cerevisiae.** *Mol Cell Biol* 1999, **19(5**3237-3245 [http://mcb.asm.org/cgi/content/abstract/19/5/3237].
26. Middendorf M, Kundaje A, Freund Y, Leslie CWC: **Motif discovery through predictive modeling of gene regulation.** *Proceedings of the Ninth International Conference on Research in Computational Molecular Biology (RECOMB 2005)* 2005 [http://www.cs.columbia.edu/compbio/medusa].
27. Schapire RE, Singer Y: **Improved boosting algorithms using confidence-rated predictions.** *Machine Learning* 1999, **37(3):**297-336.
28. Hoeffding W: **Probability inequalities for sums of bounded random variables.** *Journal of the American Statistical Association* 1963, **58(301):**13-30.
29. Pollard D: *Convergence of Stochastic Processes Springer-Verlag*; 1984:191-192.