

Predictive models of gene regulation

Anshul Bharat Kundaje

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2008

© 2008

Anshul Bharat Kundaje
All Rights Reserved

Abstract

Predictive models of gene regulation

Anshul Bharat Kundaje

The regulation of gene expression plays a central role in the development and function of a living cell. A complex network of interacting regulatory proteins bind specific sequence elements in the genome to control the amount and timing of gene expression. The abundance of genome-scale datasets from different organisms provides an opportunity to accelerate our understanding of the mechanisms of gene regulation. Developing computational tools to infer gene regulation programs from high-throughput genomic data is one of the central problems in computational biology.

In this thesis, we present a new *predictive modeling* framework for studying gene regulation. We formulate the problem of learning regulatory programs as a binary classification task: to accurately predict the condition-specific activation (up-regulation) and repression (down-regulation) of gene expression. The gene expression response is measured by microarray expression data. Genes are represented by various genomic regulatory sequence features. Experimental conditions are represented by the gene expression levels of various regulatory proteins. We use this combination of features to learn a prediction function for the regulatory response of genes under different experimental conditions. The core computational approach is based on *boosting*. Boosting algorithms allow us to learn high-accuracy, large-margin classifiers and avoid overfitting. We describe three applications of our framework to study gene regulation:

- In the *GeneClass* algorithm, we use a compendium of known transcription factor

binding sites and gene expression data to learn a global context-specific regulation program that accurately predicts differential expression. GeneClass learns a prediction function in the form of an alternating decision tree, a margin-based generalization of a decision tree. We introduce a novel robust variant of boosting that improves stability and biological interpretability in the presence of correlated features. We also show how to incorporate genome-wide protein-DNA binding data from ChIP-chip experiments into the framework.

- In several organisms, the DNA binding sites of many transcription factors are unknown. Hence, automatic discovery of regulatory sequence motifs is required. In the *MEDUSA* algorithm, we integrate raw promoter sequence data and gene expression data to simultaneously discover cis regulatory motifs ab initio and learn predictive regulatory programs. MEDUSA automatically learns probabilistic representations of motifs and their corresponding target genes. We show that we are able to accurately learn the binding sites of most known transcription factors in yeast.
- We also design new techniques for extracting biologically and statistically significant information from the learned regulatory models. We use a margin-based score to extract global condition-specific regulomes as well as cluster-specific and gene-specific regulation programs. We develop a post-processing framework for interpreting and visualizing biological information encapsulated in our models.

We show the utility of our framework in analyzing several interesting biological contexts (environmental stress responses, DNA-damage response and hypoxia-response) in the budding yeast *Saccharomyces cerevisiae*. We also show that our methods can learn regulatory programs and cis regulatory motifs in higher eukaryotes such as worms and humans. Several hypotheses generated by our methods are validated by our collaborators using biochemical experiments. Experimental results demonstrate that our framework is quantitatively and qualitatively predictive. We are able to achieve high prediction accuracy on test data and

also generate specific, testable hypotheses.

Contents

List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 Contributions	2
1.2 Outline	4
2 Biology background	7
2.1 Functional units in a living cell	7
2.2 Regulatory sequences in the genome	10
2.3 Elements of transcriptional regulation	12
2.4 Characteristics and representation of cis regulatory motifs	13
2.5 High-throughput genomic data	16
3 Machine learning approaches to modeling gene regulation	19
3.1 Related methods	19
3.2 Our approach: Predictive modeling of gene regulation	24
3.3 Introduction to Boosting algorithms	25
3.3.1 Adaboost: An adaptive Boosting algorithm	26

3.3.2	Alternating decision trees	29
4	Learning regulatory programs: GeneClass	33
4.1	Introduction	33
4.2	Related methods	34
4.3	Learning regulatory programs as alternating decision trees	36
4.3.1	Feature space	36
4.3.2	GeneClass weak learner	38
4.3.3	Predicting gene expression using a regulatory program	39
4.3.4	Stabilized boosting	39
4.4	Extracting predictive features from regulatory programs	43
4.4.1	Extracting global features	44
4.4.2	Gene set analysis: Extracting context-specific features	44
4.4.3	Signaling pathways and regulatory cascades	45
4.5	Datasets	45
4.5.1	Expression data	45
4.5.2	Discretization of expression data	46
4.5.3	Candidate set of regulators	47
4.5.4	Motif data	48
4.5.5	ChIP-chip data	48
4.6	Statistical validation	49
4.6.1	Prediction accuracy in cross-validation experiments	49
4.6.2	Motif data versus ChIP-chip data	49
4.6.3	Prediction scores show correlation with expression data	50
4.6.4	Comparison to a baseline classification method	51
4.6.5	Randomization experiments	52
4.6.6	Stabilized boosting results in robust ADTs	53
4.7	Biological validation	55

4.7.1	Globally predictive regulators and motifs	55
4.7.2	Regulators of functionally related genes	56
4.7.3	Regulation of individual target genes	58
4.7.4	Identifying signaling pathways	60
4.7.5	In silico knockouts to identify transcription factor targets	60
4.8	Conclusions	61
5	Learning cis regulatory motifs: MEDUSA	64
5.1	Introduction	64
5.2	Related methods	66
5.3	Learning cis regulatory motifs from regulatory sequence and expression data	67
5.3.1	Overview of the MEDUSA learning algorithm	67
5.3.2	Feature space	69
5.3.3	MEDUSA weak learner	70
5.3.4	Learning PSSMs using hierarchical sequence agglomeration	72
5.4	Extracting predictive features	75
5.5	Datasets	76
5.5.1	Yeast (<i>S. cerevisiae</i>) datasets	76
5.5.2	Worm (<i>C. elegans</i>) dataset	78
5.5.3	Human B-cell dataset	79
5.6	Statistical validation	80
5.6.1	Prediction accuracy in cross-validation experiments	80
5.6.2	Comparison to GeneClass	82
5.6.3	Prediction accuracy for the three-class (up/down/baseline) predic- tion problem	83
5.6.4	Comparison to simple methods based on clustering or correlation	84
5.7	Biological validation	89

5.7.1	MEDUSA discovers most known transcription factor binding sites in yeast	89
5.7.2	MEDUSA regulatory programs uncover key regulators of the DNA damage signature	92
5.7.3	Lineage-specific regulation in the early worm embryo	96
5.7.4	Condition-specific regulators and motifs in human B cells	98
5.8	Conclusion	100
6	Case Study: Regulation of hypoxia response in yeast	103
6.1	Introduction	103
6.2	Microarray experiments	105
6.3	Perturbations of the oxygen regulatory network reveal diverse expression signatures	106
6.4	Learning procedure	109
6.5	Cis regulatory motifs: Comparison to “cluster-first” motif discovery algorithms	110
6.5.1	Global comparison of motifs	110
6.5.2	Motifs specific to a functional regulon	113
6.6	Post-processing and visualization framework	114
6.6.1	Regulation of the <i>OLE1</i> gene: Biochemical experiments validate hypotheses	115
6.6.2	Context-specific regulators in different experimental conditions . .	116
6.6.3	The hypoxia regulome	119
6.7	Conclusion	124
7	Conclusion	126
7.1	Summary	126
7.2	Limitations and Future directions	128
7.2.1	Representing other modes of regulation	128

7.2.2	Discretization of expression data	129
7.2.3	Integration of other high-throughput data sources	130
7.2.4	Motif discovery in higher eukaryotes: cis regulatory modules	132
7.2.5	Comparative genomic approaches for learning conserved regulation	133
7.2.6	Use of epigenomic data to improve identification of cis regulatory elements	134
7.3	End note	135
8	Bibliography	137
A	Stabilization criteria for GeneClass	146
B	GeneClass pseudocode	150
C	MEDUSA pseudocode	153
D	Yeast samples and microarray data generation for the hypoxia dataset	156
D.1	Yeast cell growth and treatment	156
D.2	RNA preparation and microarray gene expression profiling	157
D.3	Normalization and discretization of microarray data	159
E	Functional annotations for the hypoxia expression signatures	160
F	Downloadable source code and data	163

List of Figures

2.1	The structure and composition of DNA	8
2.2	Regulation of gene expression	10
2.3	Mathematical representations of a DNA binding site	15
2.4	DNA microarrays and ChIP-chip assays	16
3.1	Use of expression data in clustering, Bayesian networks, and GeneClass . .	20
3.2	A schematic overview of Adaboost	26
3.3	Alternating decision trees	30
4.1	Training data for GeneClass	37
4.2	Interpreting regulatory programs	40
4.3	Noise model for discretizing data	47
4.4	Correlation of prediction scores with expression data	50
4.5	Test error for GeneClass with randomized data	52
4.6	Regulation of protein-folding chaperones	57
4.7	Regulation of heat shock proteins in heatshock and osmotic stress	59
5.1	Overview of the MEDUSA learning algorithm	68
5.2	Overview of the MEDUSA weak learner	70
5.3	Overview of hierarchical sequence agglomeration in MEDUSA	72

5.4	Three-class prediction performance on the hypoxia dataset	83
5.5	Confusion matrix for three-class classification on the hypoxia dataset	84
5.6	MEDUSA motifs learned from the ESR dataset	90
5.7	Motifs identified by MEDUSA for DNA damage	93
5.8	Key regulators of DNA damage signature genes as identified by MEDUSA	94
5.9	Context-specific regulation for target genes relevant to touch cell differentiation	97
5.10	Context-specific regulation in human B cells	98
6.1	Expression signatures identified by perturbation of the oxygen regulatory network	107
6.2	Comparison of motifs learned by MEDUSA and AlignACE on the hypoxia dataset	111
6.3	Experimental confirmation of the oxygen regulators identified by MEDUSA	115
6.4	Heat maps showing predictive regulators, motifs, and targets induced by oxygen and heme	117
6.5	Venn diagrams showing context-specific regulators in the hypoxia dataset	120
6.6	The hypoxia regulome	121

List of Tables

4.1	Different experimental setups and their performance	50
4.2	The effect of stabilized boosting on ranking of predictive features	54
5.1	Prediction performance for MEDUSA across multiple yeast, worm and human data sets	80
5.2	Strawman methods for comparison to MEDUSA	85
5.3	Comparison of MEDUSA prediction performance to simple clustering- based methods on the ESR data set	87
5.4	Comparison of MEDUSA prediction performance to simple correlation- based methods on the hypoxia data set	88

Acknowledgments

First and foremost I would like to thank my advisor, Dr. Christina Leslie, for her guidance and support. In my early years as a graduate student, it was Christina who inspired me to take up computational biology as my career path. She has always provided me with direction when I needed it most and also given me the freedom to pursue ideas and topics that I personally found interesting. The most valuable lesson that I have learned from her is to fearlessly and relentlessly pursue radical and unconventional ideas. Christina is everything a student could ask for in an advisor. She has had the most significant impact on my development as a scientist and a researcher. For that I am eternally grateful.

I would like to thank my unofficial co-mentor, Prof. Chris Wiggins, for introducing me to the physicist's view of biology. Chris has a very unique perspective and I have thoroughly enjoyed each and every one of our discussions. Over the years, I have collaborated with Chris and his students on several interesting projects, many of which form the bulk of my thesis. Chris once told me that in scientific research, negative results are just as interesting as positive ones and that failure is the path to success. These are golden words that I will never forget.

Prof. Li Zhang was the first biologist to have faith in our computational techniques. We collaborated with her laboratory to study the regulation of hypoxia response in yeast. By validating several hypotheses generated by our methods using biochemical experiments, she has added much-needed credibility to our work. In an interdisciplinary field such as

computational biology, it is very important to develop methods that are able to address important biological questions and present the results in a manner that is comprehensible to biologists. Li has provided us with the biologist's perspective. She has helped us enhance our post-processing and visualization framework to extract biologically meaningful and relevant information from our learned models.

I would also like to thank the other members of my thesis committee namely Prof. Kathy Mckeown, Prof. Itsik Pe'er and Prof. Tony Jebara for their valuable comments which have helped me improve the quality of this dissertation.

I would like to thank Prof. Yoav Freund for introducing me to boosting and the alternating decision tree formulation which form the core of the algorithms presented in this thesis. I am also grateful to Prof. Rocco Servidio for teaching some fascinating courses on computational learning theory. He is undoubtedly one of the best teachers in the Computer Science department at Columbia and his insightful lectures have greatly improved my understanding of machine learning.

I would also like to express my gratitude to Dr. Gustavo Stolovitzky, who was my manager during my internship at IBM Research. Gustavo introduced me to the several intriguing aspects of noise in biological systems. He has been a great mentor and friend. I would like to thank Jared Roach, Glenn Held, Keith Duggar, Christian Haudenschild, Daixing Zhou, Tom Vasicek, Kelly Smith and Alan Aderem who collaborated with me during my internship at IBM.

I would also like to thank Prof. Henning Schulzrinne and Prof. Dimitris Anastassiou for their guidance and support during my early years as a graduate student.

Ofcourse, none of this work would have been possible without the help and support of so many students and co-workers that I have had the privilege of working with. A special thanks to Manuel Middendorf with whom I closely collaborated to develop the GeneClass and MEDUSA algorithms. A big thank you to Steve Lianoglou, Xuejing Li, Marta Arias, David Quigley, Mihir Shah, Feng Gao, Omar Antar, Aaron Arvey, Franck Rappaport, Phaedra

Agius and David Sussilo for their support and friendship.

Thanks to NIH and NSF for providing grants to support my PhD. Thanks to the cBio group at The Memorial Sloan Kettering Cancer Center, The Center for Computational Learning Systems at Columbia and the Computer Science department at Columbia for providing me with computational resources.

I would also like to take this opportunity to extend my gratitude to several friends in and around New York City who kept me sane and provided moral and emotional support during the good times and the bad times. A special shout out to Shane Eisenman, Vishal Singh, Sowmya Vishwanath, Nikhil Suri, Ripla Arora, Parag Purohit, Nupur Hiremath, Dhruv Mahajan, Pranay Tigga, Nithyanandan Thyagarajan and the rest of the volleyball gang and the Indian Students Association at Columbia.

Finally and above all, I am eternally grateful to my family for their love and support. I would never have made it so far without the encouragement from my parents Bharat and Kalpana Kundaje, my little brother Kunal and my fiancée Nalini. They have always been there for me especially during the hardest times and have been a constant source of emotional stability and motivation.

Dedication

Dedicated to my grandparents

Girish Kodkani

and

Shalini Kodkani

Introduction

The complex behavior of cellular biological systems — including growth, reproduction, and adaptation to environmental changes — derives from the molecular interactions of thousands of genes and their products in a highly intricate and poorly understood network. Understanding how this network operates and predicting its behavior are primary goals of biology and have broad implications for science, medicine, and biotechnology.

The genomic information revolution of the last decade has made it possible to study complex cellular networks from a global and data-driven perspective. We now have the complete DNA sequences for scores of organisms, giving an increasingly detailed “parts list” of the makeup of the cell and the structure of genes. High-throughput molecular assays provide copious but noisy and incomplete data on the molecular state of the cell under different experimental conditions.

In all organisms, the cellular network operates on a few well established general principles. Proteins, encoded by the DNA of the genome (genes), are translated from intermediate messenger RNA molecules (mRNAs). The production of mRNA from the genome (transcription) is controlled by regulatory proteins (transcription factors) that bind to the DNA in special regions referred to as promoters. The transcription factors are in turn regulated by complex pathways of molecular interactions in the cell. The switching on and off of genes is one of the key methods that the cell uses for controlling its behavior and response to environ-

mental changes. Many members of the transcriptional regulatory machinery, and to a lesser extent the DNA sequences they bind to on promoters, are already described. We are only beginning to discover how these parts work together. While only part of the picture, studying these gene regulatory mechanisms — the interplay of mRNA expression (measured with microarrays), regulators, and promoter sequences — has already emerged as an important paradigm for dissecting molecular networks with machine learning methods [34, 39].

Machine learning approaches offer powerful new tools for using these data to make predictions of the underlying structure of the network and its behavior, and developing these techniques has become a central problem in computational biology [34, 39]. The ultimate goal is to provide biologists with new computational tools to generate hypotheses and guide wet lab experiments in an iterative process of prediction and testing.

1.1 Contributions

Our main contributions in this thesis are enumerated below.

- We present a predictive modeling framework, based on *boosting* algorithms for learning details of transcriptional regulation from heterogeneous sources of high-throughput genomic data. We integrate regulatory sequence data, DNA-binding data and microarray expression data into a unified model that is based on biologically meaningful assumptions. From a computational perspective, we present stabilized variants of boosting algorithms that work well in the presence of meaningful, correlated features.
- Our models are quantitatively and qualitatively predictive. We show that it is possible to learn prediction functions that are both accurate and qualitatively interpretable. In machine learning, algorithms often tend to be evaluated primarily on the basis of their statistical and computational performance. However, in an applied setting, it is important to develop algorithms that are also able to generate well-defined testable hypotheses. In this thesis, we develop new scores to rank and extract features from our

models, that provide potential answers to specific biological questions. We introduce a post-processing framework to extract and display interesting biological information.

- We introduce a novel algorithm called *GeneClass* to learn gene regulatory programs from gene expression data and regulatory sequence data. Specifically, we model the problem of learning regulatory programs as a binary classification task in which we predict the up-regulation and down-regulation of genes in different sets of experimental conditions. In contrast to most methods, GeneClass avoids grouping genes into static clusters and is able to learn a single global model of gene regulation over all genes and all experimental conditions. This approach allows genes to be coregulated with different sets of genes in a context dependent manner. The prediction function learned by GeneClass is context-specific. In order to predict the expression level of a gene in an experiment, we need to represent the gene context and the experimental context. We use regulatory sequence information such as known transcription factor binding sites and high-throughput DNA binding data to represent the gene context. We use the expression levels of regulatory proteins to represent the experimental (cellular) context. The basic modeling assumption is that we should be able to predict the expression level of any gene in any experimental condition by using the regulatory sequence of the gene and the expression levels of regulatory proteins in that experiment. Thus, using GeneClass we attempt to answer the most basic and often poorly understood biological question - “Which regulatory proteins regulate which target genes through which sequence elements under what conditions?” We specifically apply GeneClass to learn regulatory programs relevant to environmental stress responses and DNA damage stress responses in yeast.
- One limitation of GeneClass is that it relies on having information about regulatory sequence motifs or binding targets of regulatory proteins. Unfortunately, in most organisms very little is known about the specifics of DNA binding sites of regulatory

proteins. Thus, we introduce *MEDUSA* (Motif Element Detection Using Sequence Agglomeration), an algorithm that is not only able to learn regulatory programs but also discover DNA binding sites de novo. The experimental context is once again represented by the expression levels of regulatory proteins. However, *MEDUSA* represents the gene context by its promoter sequence. We search through all possible subsequences (k -mers) and gapped elements (dimers) in the promoter sequences of all genes to discover probabilistic motifs that allow us to predict up/down expression of target genes. Most motif discovery algorithms either use regulatory sequence data alone or use gene expression data indirectly to cluster genes as a preprocessing step. *MEDUSA* is able to elegantly integrate these two complementary sources of information. We apply *MEDUSA* to various gene expression datasets of different sizes in organisms such as yeast, worm and humans.

- In order to test the usefulness of our algorithms in the field, we used GeneClass and *MEDUSA* to rigorously analyze a small gene expression dataset that probes the response of yeast to hypoxia (low oxygen levels). Little is known about the regulators of hypoxia stress response. Using our framework, we were able to decipher the hypoxia regulome. We not only identified several known regulators and sequence motifs but also discovered several new ones. Some of our hypotheses were validated by our collaborators through biochemical experiments. Thus, we show that our methods are able to decipher novel regulatory relationships even in the presence of limited amounts of noisy data.

1.2 Outline

The outline of the thesis is as follows.

In Chapter 2, we start with a brief overview of the biology of gene regulation. We describe the characteristics of the main functional elements in the cell such as DNA, RNA

and proteins. We discuss key gene regulatory processes specifically focusing on the process of transcription. We describe characteristics of and representations for DNA binding sites. We introduce the main types of high-throughput assays and data types that we refer to in this thesis.

In Chapter 3, we review several machine learning approaches used to study gene regulation. We introduce our predictive modeling approach and show how we improve upon the state of the art. Finally, we give an overview of boosting and alternating decision trees.

In Chapter 4, we present a novel classification-based algorithm called *GeneClass* for learning gene regulatory programs from gene expression data and a candidate set of regulatory proteins and sequence motifs. We also show how to incorporate genome-wide protein-DNA binding data into the *GeneClass* algorithm. In computational experiments based on yeast environmental stress response and DNA damage datasets, we show that *GeneClass* predicts up- and down-regulation on held-out experiments with high accuracy. We explore a range of experimental setups related to environmental stress response, and we retrieve important regulators, binding site motifs, and relationships between regulators and binding sites that are known to be associated with specific stress response pathways. We present a postprocessing framework for biological interpretation, including gene and gene set analysis to reveal condition-specific regulatory programs and to suggest signaling pathways. This chapter is based on work presented in [79], [80] and [62].

In Chapter 5, we present *MEDUSA*, an integrative method for learning motifs representing transcription factor binding sites by incorporating promoter sequence and gene expression data. Like *GeneClass*, *MEDUSA* also produces a model of the transcriptional control logic that can predict the expression of any gene in the organism, given the sequence of the promoter region of the target gene and the expression state of a set of known or putative regulatory proteins. We apply *MEDUSA* to various datasets of different sizes in yeast, worm and human B-cells. We learn yeast motifs whose ability to predict differential expression of target genes outperforms motifs from a compendium of known binding

sites and from a previously published candidate set of learned motifs. We also show that MEDUSA retrieves many experimentally confirmed transcription factor binding sites. We introduce a novel margin-based score to extract significant context-specific regulators and motifs. This chapter is based on work presented in [78].

In Chapter 6, we present a specific case study where our collaborators validate some of our regulatory hypotheses using biochemical experiments. We use GeneClass and MEDUSA to study the oxygen regulatory network in the yeast (*S. cerevisiae*), using a small data set of perturbation experiments that probe the response of yeast to hypoxia (low oxygen levels). We assemble a global map of the oxygen sensing and regulatory network. We also identify many DNA motifs that are consistent with previous experimentally identified transcription factor binding sites. Our collaborators directly test a set of regulators predicted by MEDUSA for the *OLE1* gene that is specifically induced under hypoxia, by experimental analysis of the activity of its promoter. In each case, deletion of the candidate regulator results in the predicted effect on promoter activity, confirming that several novel regulators identified by MEDUSA are indeed involved in oxygen regulation. This chapter is based on work presented in [61].

Chapter 7 contains a brief summary of the thesis and concluding remarks. We provide a discussion of some of the limitations of our framework and future challenges and extensions to our current framework.

Biology background

In this chapter, we start with a brief overview of the biology of gene regulation. We describe the characteristics of the main functional elements in the cell such as DNA, RNA and proteins. We discuss key gene regulatory processes specifically focusing on the process of transcription. We describe characteristics of and representations for DNA binding sites. We then introduce the main types of high-throughput assays and data types that we learn from.

2.1 Functional units in a living cell

The living cell is a complex system of interacting and tightly regulated biochemical entities. The primary heritable, encoding unit is the *genome* which is enclosed within the nucleus in eukaryotic cells. The genome is made up of a biopolymer known as *deoxyribose nucleic acid* (DNA). DNA is a double stranded, linear, unbranched polymer. Each monomeric subunit is known as a *nucleotide*. A nucleotide is made up of a pentose sugar molecule known as deoxyribose, a nitrogenous base and a phosphate group. The most common structural conformation of DNA known as B-DNA is a right-handed double helix (See Figure 2.1). Two complementary strands of DNA run in opposite directions. *Base-pairing* between the two strands stabilizes the structure. This base-pairing involves the formation of hydrogen bonds between complementary bases of the nucleotides. The bases are of four types: adenine

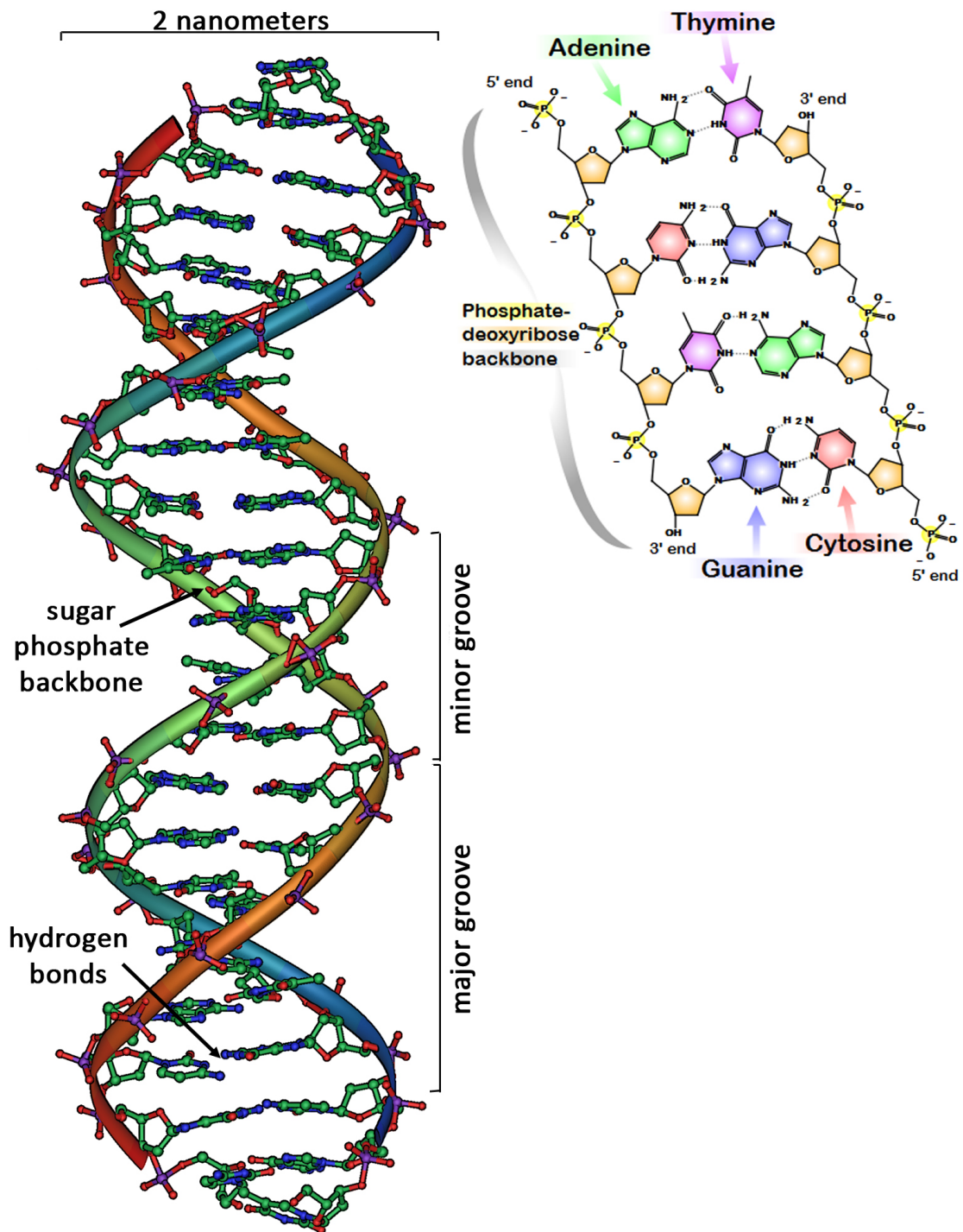


Figure 2.1: **The structure and composition of DNA:** Two representations of the double helix. On the left the structure is shown with the sugar-phosphate backbones of each polynucleotide with the base pairs involving hydrogen bonds. On the right the chemical structure for four base pairs is given. A base-pairs with T, and G base-pairs with C. (Image taken from <http://en.wikipedia.org/wiki/DNA>)

(A), cytosine (C), guanine (G) and thymine (T). Adenine pairs with thymine (A-T) and guanine pairs with cytosine (G-C). The exact sequence of nucleotides in the genome and the complementarity of *base pairs* (bp) is the fundamental property of DNA that allows it to store, replicate and transfer information. *DNA polymerases* are enzymes that read off one strand of DNA and exploit the complementarity to synthesize a new DNA molecule.

Ribose nucleic acid (RNA) is another polynucleotide similar to DNA but with two differences. The sugar in an RNA nucleotide is ribose and RNA contains uracil (U) instead of thymine (T). Also, most RNA molecules in their functional form are single stranded. RNA molecules are less stable than DNA. Hence, they are generally upto a few thousand nucleotides in length. *RNA polymerases* are enzymes that read off sections of DNA to make complementary RNA copies. This process is known as *transcription* (See Figure 2.2 (A)). RNA is the primary molecule that transfers biological information out of the genome.

The sequence of nucleotides in the genome encode various functional units. *Genes* represent regions of DNA that are copied into various classes of small and large RNAs. The messenger RNAs (mRNA) are a specific class of RNAs that are further decoded into proteins which form the basic units of various sensing, regulatory and functional subsystems in the cell. The process of decoding mRNA molecules into proteins is known as *translation*. *Proteins*, like DNA, are biopolymers (rarely more than 2000 units in length) where the monomeric subunits are called *amino acids*. There are 21 types of amino-acids. Specific triplets of nucleotides known as *codons* encode specific amino acids. This is known as the *genetic-code*. The genetic code is partially degenerate i.e. multiple codons can code for the same amino acid. This redundancy allows genetic polymorphisms and resistance to genomic instability due to mutations. In most cases, the entire sequence of a protein-coding gene does not get translated. The portions of the gene that code for the protein are called *exons* and the intermediate regions are known as *introns*. Introns are spliced out of the mRNA molecules before their translation to proteins.

The transfer of genetic information from the nucleotide sequence of a gene to the

nucleotide sequence of RNA or the amino acid sequence of a protein is termed *gene expression* (See Figure 2.2 (A)). However, for the purpose of this thesis, gene expression primarily refers to the process of transcription.

2.2 Regulatory sequences in the genome

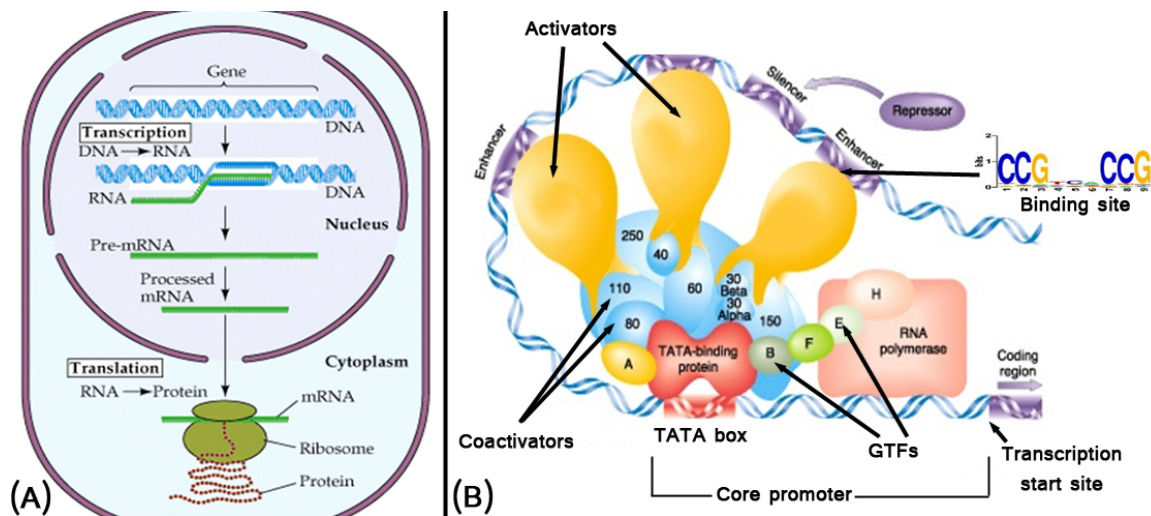


Figure 2.2: Regulation of gene expression: (A) shows the processes involved in eukaryotic gene expression namely transcription and translation (Image taken from 2001 Sinauer Associates Inc.) (B) shows the structure of a typical eukaryotic transcriptional system. The core promoter consists of the TATA box and the transcription start site. RNA polymerase and general transcription factors (GTFs) bind to this area. Activators and repressors bind to sequence-specific binding sites in the proximal promoter as well as enhancers and silencers which are distal regulatory regions in the genome. (Image modified from http://scienceblogs.com/pharyngula/2007/01/basics_what_is_a_gene.php)

Living cells are extremely robust and adaptable systems. The presence/absence of external/internal perturbation signals such as temperature and chemical concentrations, is sensed and transduced by several interconnected signal transduction pathways. These signaling pathways are generally cascades of interacting proteins and the signals are various chemical modifications that change the conformational structure and hence the activity of the proteins. The signaling cascades ultimately converge on the DNA by modifying specific DNA binding proteins known as transcription factors that regulate the process of

gene transcription. *Transcription factors* regulate the activation and/or repression of gene expression by binding to sequence elements (motifs) on DNA. The DNA sequence thus also encodes a plethora of regulatory information. The regulatory sequence units tend to localize to certain regions of the genome.

Promoters are regulatory sequence-rich regions which are proximal to genes. Promoters of genes that encode proteins have three identifying characteristics (See Figure 2.2 (B)). A transcription start site, the TATA box or other initiation regions and transcription factor binding sites. The typical length of a core promoter (containing the start site and the TATA box) is approximately 100 bp. In lower eukaryotes such as the yeast *S. cerevisiae* and the worm *C. elegans*, binding sites for transcription factors extend upto 1000 bp. upstream of the transcription start site. This is typically the length of the entire promoter. However, in higher eukaryotes such as mammals, promoter sequences can sometimes be as long as 5000 bp. Recent studies which are part of the ENCODE project [2] have shown that transcription factor binding sites are also abundant downstream of the transcription start site upto and including the first intron. The TATA box or other initiation elements are structural elements which define a minimal promoter required for recruiting RNA polymerase. In higher eukaryotes, the TATA box is normally around 30 bp upstream of the transcription start site. These initiation elements are sufficient for formation of the basal transcriptional apparatus which allows a basal transcription activity. A regulated transcription (upregulation or downregulation) requires the sequence-specific transcription factors to bind to the regulatory sequences in the genome.

Enhancers, silencers and *insulators* are regulatory regions that can be several thousands of base pairs away from the genes they affect. Enhancers and silencers typically contain clusters of DNA-binding motifs that affect transcription independent of their orientation and position.

2.3 Elements of transcriptional regulation

The essential elements of regulation at the level of transcription involve:

Cis regulatory sequences: Transcription factors tend to bind sequence specific regulatory motifs in the proximal promoters of genes or in distal enhancers and silencers. In higher eukaryotes such as mammals, these cis elements are also abundantly found within the transcribed region. Transcription factor binding to these cis elements can have an activating or inhibiting effect on transcription. The DNA binding sites for sequence-specific transcription factors are usually around 3-8 base pairs in length. Many transcription factors have dimeric DNA binding domains. The symmetry of the DNA binding domain of the protein is often reflected in the corresponding DNA binding site. Also, several transcription factors tend to bind DNA in the form of dimers or higher-order structures. Hence, the dimer binding sites are commonly arranged either palindromically, in direct repeats or inverted repeats. The distance between and the nature of the bases in the dimeric binding sites plays an important role. Changes in distance by even a few base pairs can cause a loss in cooperativity. Another aspect of multimeric cis regulatory motifs is the formation of heterodimers. This generally occurs when different interacting transcription factors collaboratively bind DNA.

Concentration of trans-acting transcription factors: The presence of a transcription factor binding site in the promoter or enhancer of a gene is not sufficient to activate or repress it. The transcription factor itself must be in its active conformation and must be present in sufficient concentration. The concentration of transcription factors affects the transcription rate in a non-linear fashion. Transcription factors tend to bind DNA stochastically. They can bind different sequence elements with different affinities. When a transcription factor is present in high concentrations, low affinity binding sites can substantially affect gene expression. In most cases, it is difficult to measure the exact concentration of active transcription factors. In this thesis, we use

the mRNA expression levels of transcription factors and upstream signaling molecules as a surrogate for their true activity.

2.4 Characteristics and representation of cis regulatory motifs

An important property of transcription factors is that they do not bind sequence elements in a binary manner. The affinity of a transcription factor to a particular sequence element is determined by the types and positions of the nucleotides in the sequence motif. Some nucleotides in some positions are generally more important than others. The degeneracy of a cis regulatory motif is captured by using a *consensus sequence* or a position independent probabilistic representation known as a *position-specific scoring matrix* (PSSM).

The *consensus sequence* of a DNA binding site is obtained by aligning known variants of the site. It represents an idealized sequence motif that represents the predominant bases at each position. The following IUPAC letters are used to represent ambiguity in DNA sequences:

R = G,A (purine)	S = G,C	H = A,C,T
Y = T,C (pyrimidine)	W = A,T	V = G,C,A
K = G,T (keto)	B = G,T,C	N = A,G,C,T
M = A,C (amino)	D = G,A,T	

While a consensus sequence (See Figure 2.3) is a compact representation of a binding site it provides little insight into the quantitative conservation of base pairs at each position in a DNA binding site. Also, the construction of a consensus sequence is relatively arbitrary since it is not clear what fraction of degeneracy warrants the use of a particular consensus symbol.

A *position-specific scoring matrix* (PSSM) is a richer probabilistic representation for a set of aligned sequence elements. It captures relative preference for the four base pairs

at each position. The basic assumption is that the probability of observing a base pair at a particular position in the sequence motif is independent of all other positions. This assumption has been shown to be not entirely accurate [17], [76]. However, it is generally difficult to reliably learn more complex models that involve positional dependencies due to the limited number of degenerate observations of a binding site. For a binding site of length k , a PSSM P is a $k \times 4$ matrix that assigns a probability $p_i(x)$, for each position $i = 1 \dots k$ and nucleotide $x \in \{A, C, G, T\}$. These probabilities can be obtained from an aligned set of sequences by calculating the normalized fraction of every nucleotide at every position. A typical and convenient visualization of a PSSM is via a *PSSM logo* [103] as shown in Figure 2.3. The uncertainty at each position i is given by the entropy defined as $H(i) = -\sum_{x \in \{A, C, G, T\}} p_i(x) \log_2 p_i(x)$. The information at the position is represented by the decrease in uncertainty i.e. $R(i) = 2 - H(i)$. The height of each base in the logo is given by $p_i(x)R(i)$. The bases are then stacked on top of each other in increasing order of their frequencies and plotted. An example is shown in Figure 2.3.

In order to search for hits of a binding site of length k in a longer sequence, we score all overlapping subsequences of length k using a *log-odds score*. For a subsequence $a_1, a_2 \dots a_k$ where $a_i \in \{A, C, G, T\}$, and a PSSM as defined above, the log-odds score is defined as $\sum_{i=1 \dots k} \log_2 (p_i(x = a_i) / p^{bg}(x = a_i))$. The background probability of nucleotide x is given by $p^{bg}(x)$. This log-odds score is compared to some threshold θ to determine if the subsequence is a hit. Determining this threshold accurately is a challenging task. Most motif discovery methods determine this threshold by optimizing for false positives or false negatives. However, this is difficult to determine since true target sites of a DNA binding protein are rarely available. In MEDUSA, we are able to automatically optimize this threshold.

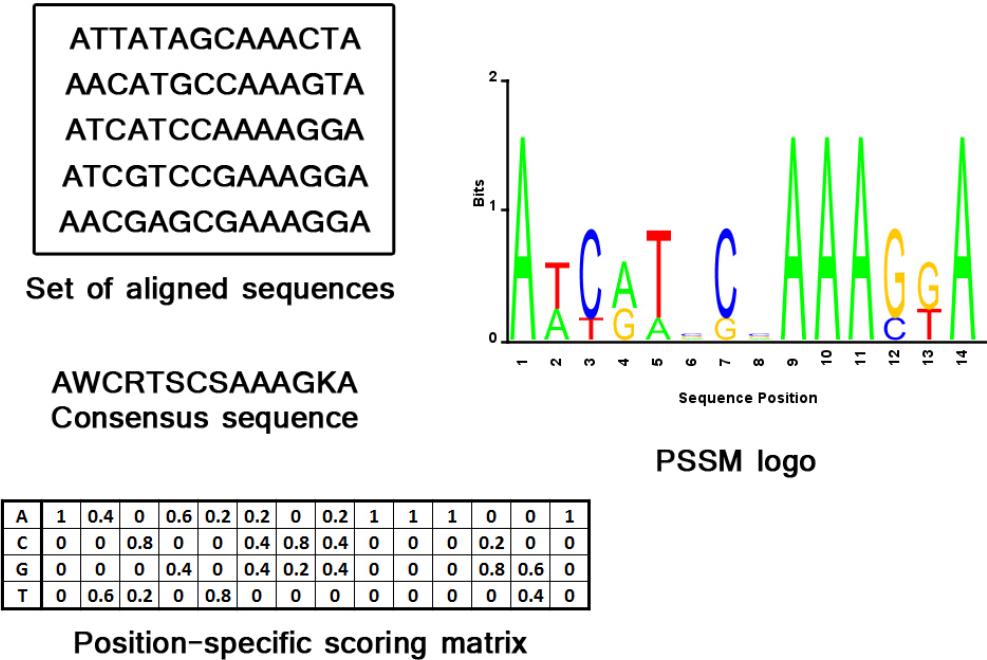


Figure 2.3: **Mathematical representations of a DNA binding site:** The figure shows a set of aligned sequences and their corresponding representations as a consensus sequence, position-specific scoring matrix and a PSSM logo

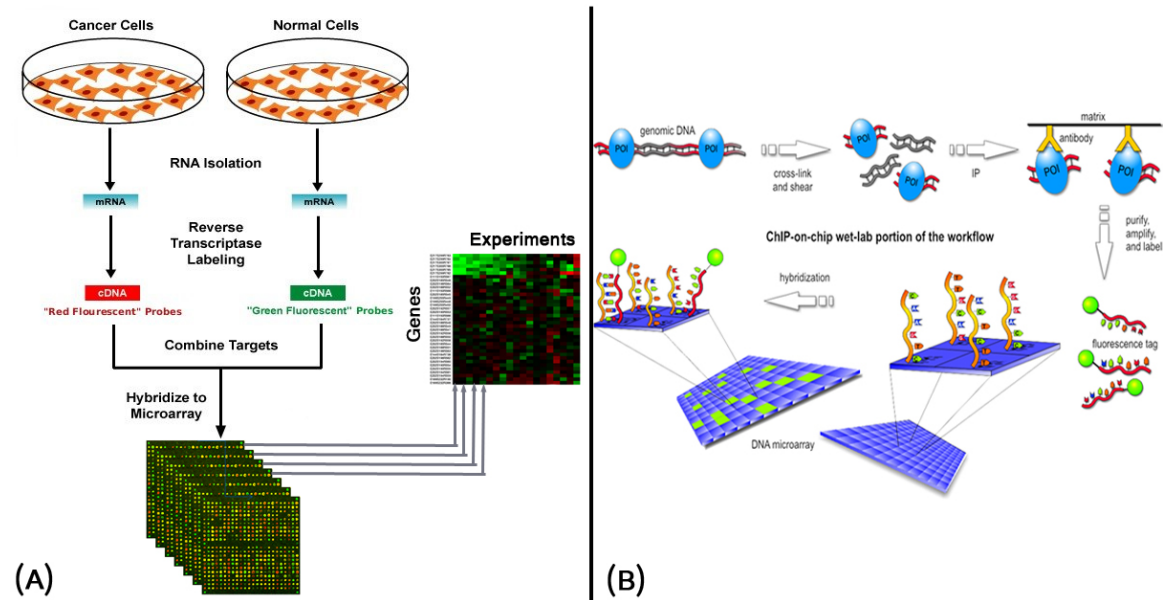


Figure 2.4: **DNA microarrays and ChIP-chip assays:** (A) shows a schematic overview of the process of gene expression measurement using dual-channel cDNA microarrays (Image modified from http://en.wikipedia.org/wiki/DNA_microarray) (B) shows workflow overview of a ChIP-chip assay. (Image taken from <http://en.wikipedia.org/wiki/ChIP-on-chip>)

2.5 High-throughput genomic data

DNA microarrays [48] allow global and parallel measurements of cellular activity. Microarrays are typically used to measure genome-wide gene expression profiles (gene expression microarrays). They can also be used to simultaneously measure the affinity of transcription factors to promoter regions of all genes in a genome. Below is a brief description of some of the high-throughput technology that is referenced in this thesis.

Gene Expression Microarrays: A DNA microarray [127] is a collection of microscopic DNA spots. The spots contain probes for single genes or gene products, arrayed on a solid surface by covalent attachment to chemically suitable matrices. DNA microarrays are based on the process of *hybridization* which is a process by which a DNA or RNA strand binds to its unique complementary strand. Qualitative or quantitative measurements with DNA microarrays utilize this selective nature of DNA-DNA or DNA-RNA hybridization under high-stringency conditions and fluorophore-

based detection. DNA arrays are most commonly used for *gene expression profiling* i.e. monitoring expression levels of thousands of genes simultaneously.

The two most common DNA microarrays are spotted arrays and oligonucleotide arrays. In spotted arrays, the probes are typically cDNA (DNA that obtained from reverse-transcribing mRNA). This type of array is hybridized with cDNA from two samples to be compared (e.g. normal vs. cancer) that are labeled with two different fluorophores (typically red and green). The samples can be mixed and hybridized to one single microarray that is then scanned, allowing the genome-wide visualization and quantification of up-regulated and down-regulated genes. The degree of up or down regulation of a gene in an experiment is typically displayed by varying intensities of red or green respectively. In oligonucleotide arrays, the probes are designed to match parts of the sequence of known or predicted mRNAs. These microarrays give estimations of the absolute value of gene expression and therefore the comparison of two conditions requires the use of two separate microarrays.

Typically, a gene expression dataset consists of multiple microarray experiments represented as a matrix of expression fold changes in log scale, where each row in the matrix represents a gene and each column represents a single microarray experiment for spotted, dual-channel arrays or a comparison of two microarray experiments for oligonucleotide, single-channel arrays (See Figure 2.4 (A)).

Protein-DNA binding arrays (ChIP chips): Chromatin immunoprecipitation (IP) [126] is an in-vivo technique used to determine whether DNA binding proteins such as transcription factors bind to a particular region on the genome. DNA-bound proteins in living cells are cross-linked using formaldehyde fixation to the DNA sequences to which they are bound. The cells are lysed and sonicated to break the DNA into fragments. The protein-DNA complexes are then immunoprecipitated using an antibody specific for the protein (See Figure 2.4 (B)). The fragments are then allowed

to hybridize to DNA microarrays. Each spot on the microarray covers a genomic region of interest such as the promoter region of a particular gene. For each genomic region, the protein-DNA binding affinity is generally reported as the log intensity ratio of an IP-enriched channel versus a background genomic DNA channel. ChIP-chip technology has been used to measure the genome-wide binding profiles of most yeast transcription factors [42, 69].

Machine learning approaches to modeling gene regulation

In this chapter, we present a review of a few machine learning approaches used to study gene regulation. We introduce our predictive modeling approach and show how we improve upon the state of the art. Finally, we give an overview of boosting and alternating decision trees.

3.1 Related methods

Due to the complexity of high-throughput genomic data and limited biological understanding of the underlying regulatory network, the first main challenge in applying machine learning to study gene regulation is deciding how to formulate the problem as a learning task.

The first and still most widely used method of learning about gene regulation from mRNA expression data is the clustering of genes by their expression profiles. Hierarchical clustering of gene expression profiles was introduced a decade ago [27], followed by numerous papers proposing alternate clustering algorithms and thousands of studies reporting the results of gene cluster analysis. From a machine learning perspective, clustering views the gene expression matrix obtained by measuring genes across multiple conditions as a set of “row

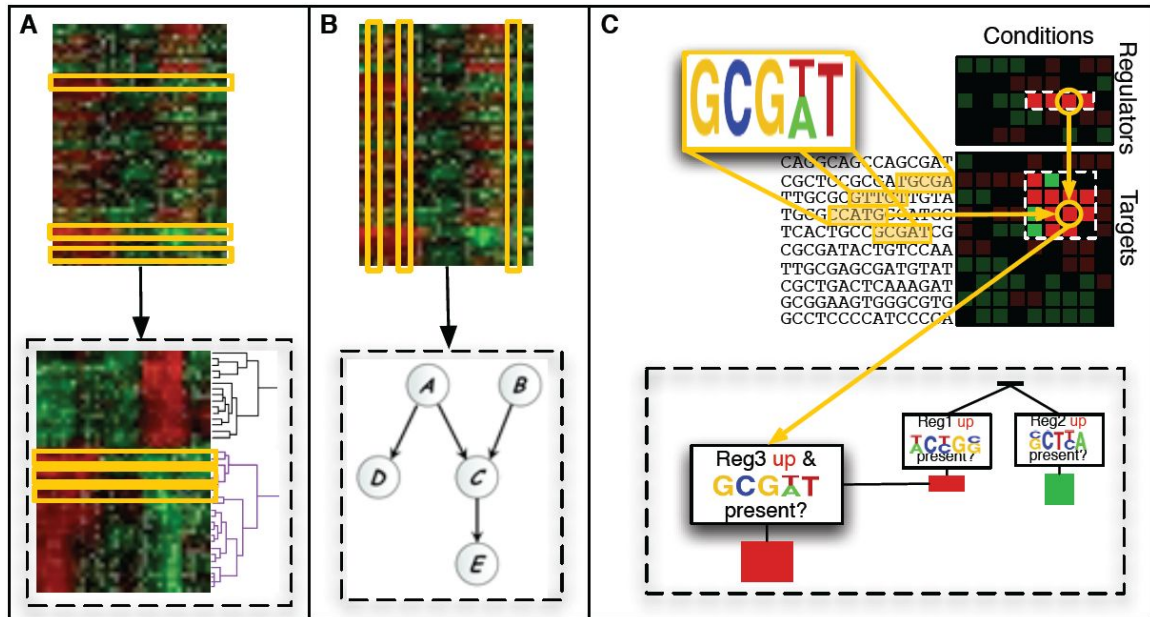


Figure 3.1: Use of expression data in clustering, Bayesian networks, and GeneClass/MEDUSA: (A) Clustering considers rows of the expression matrix, representing expression profiles for genes, and computes pairwise similarities between rows. The pairwise similarities are used to produce a set of gene clusters. (B) Bayesian networks treat each column of the expression matrix, representing the expression levels of all genes in a particular experiment, as the joint observation of thousands of gene random variables. Joint expression observations are modeled probabilistically to determine a model that maximizes the likelihood (or a related Bayesian objective function) of the data. (C) In GeneClass and MEDUSA, every differentially expressed target gene-experiment example is a separate training example, represented by sequence data (the gene's promoter sequence) and regulation expression data (the expression states of regulators in the experiment). We discover sequence motifs and select regulators that jointly help predict target expression across the entire training set.

The regulatory program can then be used to predict up/down target gene expression in held-out data.

vectors” (Figure 3.1(A)), and makes use of the strong correlation structure between rows in order to find clusters. The biological purpose of clustering is the expectation that genes with correlated expression patterns may be co-regulated. In an important study establishing a link between co-expression and co-regulation, Tavazoie and colleagues [117] showed that common motifs could be found in the promoters of co-clustered genes, yielding a clustering-based method for discovering putative *cis* regulatory elements. Numerous motif discovery algorithms based on this “cluster-first” approach – that is, first clustering genes based on expression profiles, annotations, or both, and then looking for overrepresented patterns in the promoters of genes in each cluster – have since been proposed, including MEME [5], Consensus [46], Gibbs Sampler [67], AlignACE [53] and many others.

Despite its popularity, clustering is widely understood to have limitations, both for discovering regulatory motifs and more generally for modeling gene regulation. A recent comment on the challenges of scaling to human promoters describes the following difficulties [35]: “correlation between clusters and motifs is not a one-to-one relationship [20]; often many genes in a cluster do not contain any known motif, and not all genes that contain a motif belong to the cluster from which it was derived. Furthermore, motif combinatorics could not be easily deduced. For instance, two motifs may be derived from a cluster either because they truly synergize in regulating the cluster’s genes, or simply because they form alternative regulatory programs that converged onto a similar pattern [92].” More generally, clusters are static: they imply a regulatory model where co-clustered genes are controlled by the same set of regulators across all experiments. Static clusters do not represent context-specific regulation, i.e. a model where different sets of genes are co-regulated by different regulators, as mediated by different DNA motifs, under different experimental conditions.

One approach to the context-specificity problem of clustering is the development of biclustering algorithms (reviewed in [116]). Biclustering attempts to find blocks of genes and experiments with a coordinated expression response, though the notion of a bicluster is not as well defined as a cluster [116]. An important advance to address the context-specificity

problem in modeling gene regulation came from the machine learning community, when Friedman and colleagues followed by other groups applied Bayesian networks, also called probabilistic graphical models, to this problem domain [33,44,86]. In the classical Bayesian network formulation, every “column vector” or experiment of the gene expression matrix is a joint observation of thousands of random variables (genes), and the goal is to find the structure of a probabilistic network that best accounts for the conditional dependencies and independencies of these variables (Figure 3.1(B)). These approaches had encouraging successes at retrieving pieces of known regulatory networks in yeast [86]. There have also been non-Bayesian approaches that construct network models based on the statistics of “column vectors” of the expression matrix, in particular, methods based on mutual information (e.g. [6,87]). Other authors have tried to learn explicit parameterized models for pieces of the regulatory network by fitting linear models to the training data [25,134].

More recently, there have been several efforts to incorporate the advantages of clustering – e.g. a cluster of genes gives a stronger statistical signal than a single gene, standard motif discovery algorithms can be applied to clusters of genes – within a probabilistic graphical modeling framework, creating a hybrid of the data views in Figure 3.1(A) and (B) where cluster assignment is modeled as a hidden variable. One such hybrid approach is the module networks algorithm of Segal *et al.* [104], which partitions genes into clusters or “modules”, each of which is assigned a set of regulators that control the module genes in a condition-specific manner. In this setting, the assignment of genes to clusters is still static, but the model proposes an explanation for the cluster’s variation in expression across conditions in terms of the activity of regulators. Global regulatory models that integrate gene expression data and promoter sequence data have been relatively rare. There have been two approaches that conceptually try to reverse the traditional flow of information from clusters to motifs. One approach, due to Beer and Tavazoie [10], clusters genes, learns cluster-specific motifs by standard motif-discovery methods, and then learns a Bayesian network model to assess how well a gene’s cluster membership can be predicted by the motif content of its promoter. The

second approach, due to Segal *et al.* [105], uses a technically more sophisticated algorithm called probabilistic relational models to learn clusters of genes whose shared expression patterns are also explained by shared motifs; the algorithm is seeded by database motifs, though the motifs may be re-estimated based on the data in the course of training the model. Most recently, in the special case of time series expression data, Ernst *et al.* [29] have proposed a probabilistic model that induces a temporally-organized hierarchical clustering of genes, where bifurcations of genes that go up or down at specific time points are explained by shared motifs or ChIP chip occupancy data. There have been attempts to generalize beyond the static cluster assumption, for example by allowing overlapping clusters via probabilistic assignments, but statistically these models do not yet seem to perform as well as the static models [7].

As structure learning approaches become more complex and attempt to integrate multiple kinds of noisy data, a few general comments are in order. First, learning complex structure from limited data – at best a few 100 microarray experiments, and typically much fewer – is statistically problematic. The key challenge is to avoid overfitting, i.e. the scenario where too complicated a model is fit to the training data and fails to generalize to new (test) data. Second, if the goal is to learn the “true” structure (network, modules, etc.), in most cases there is no gold standard by which to evaluate success. While one can evaluate how well a particular structure learning algorithm performs on simulated data sampled from a known model (e.g. [112]), it is more difficult to assess how well the model assumptions reflect the underlying biology in real data. Third, recent structure learning work on gene regulatory models has incorporated a cluster or modular assumption in order to gain statistical power, but in virtually all cases, this has meant relying on static clusters.

3.2 Our approach: Predictive modeling of gene regulation

We propose a new algorithmic approach for learning and interpreting predictive models of gene regulation. In the context of this thesis, a predictive model is one that accomplishes two goals. First, the model represents a regulatory program that predicts the differential expression of target genes in terms of biologically meaningful regulatory inputs, including the context-specific expression of transcriptional regulators and signal transducers and the presence of shared motifs in the regulatory sequences of target genes. Therefore, rather than learning a network or a set of clusters/modules, we are learning a prediction function, and we view the learning task as a prediction problem rather than a model selection problem. Second, the model should not only be able to make quantitative predictions, but it should make accurate predictions on data not seen in training (test data). The key issue is to use a learning strategy that avoids overfitting in the high-dimensional feature space of potential regulators and sequence motifs and in the presence of noisy gene expression data. Our strategy is based on boosting, a technique from statistical learning theory that has empirically shown resistance to overfitting in noisy and high-dimensional feature spaces.

The core of our approach are two novel algorithms called GeneClass [79] — which learns to predict gene regulatory response from regulatory sequence and expression data — and MEDUSA [78], which additionally discovers motifs representing transcription factor binding sites. The inputs to the GeneClass 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. The output is a prediction of the expression state of the regulated gene. In MEDUSA, the binding site motifs are learned from the raw promoter sequence while building the prediction function for differential expression. Rather than trying to predict a real-valued expression level, we formulate the task as a binary classification problem, that is, we predict only whether the gene is up- or down-regulated. This reduction allows us to exploit modern and effective classification algorithms. GeneClass and MEDUSA use stabilized variants of the Adaboost learning

algorithm with a margin-based generalization of decision trees called alternating decision trees (ADTs). Boosting, like support vector machines [120], is a large-margin classification algorithm that performs well for high-dimensional problems. We evaluate the performance of our method by measuring prediction accuracy on held-out test data, and we achieve very good classification results in this setting.

Our approach uses expression data in a significantly different way than previous approaches like clustering and Bayesian networks, as illustrated in Figure 3.1. Instead of computing correlations between rows of the expression matrix, as in clustering, or viewing every column of the matrix as a joint observation of thousands of gene variables, as in Bayesian networks, in the GeneClass/MEDUSA approach, every differentially expressed target gene example is a training example. We learn to predict the up/down expression of these training examples by using both regulatory sequence data and the expression of regulators. This way of using expression data and integrating sequence data allows us to learn from a very large training set, typically consisting of 10,000s of differentially expressed examples. Moreover, because we use biologically meaningful inputs for learning regulatory programs, we can also analyze the learned model to extract biologically meaningful information and to derive specific hypotheses about gene regulation. Computational analysis of regulatory programs learned by GeneClass and MEDUSA can generate networks showing connections from regulators and targets and among regulators or define “modules” of similarly regulated target genes.

3.3 Introduction to Boosting algorithms

Boosting is a meta-algorithm that is used to create a highly accurate prediction rule known as a “strong” hypothesis by combining several “weak” hypotheses, each of which is only slightly better than a random predictor.

predictions $h_t(x)$ and the labels y are de-correlated. The weak learner is then called with the new weights over the training examples and the process repeats. Finally, one takes a linear combination of all the weak prediction rules to obtain a real-valued *strong* prediction function or *prediction score* $F(x)$. The strong prediction rule is given by $\text{sign}(F(x))$ (See Figure 3.2).

Thus, after T iterations

$$F_T(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (3.1)$$

where the linear coefficients α_t can be positive or negative indicating the contribution of the weak rule h_t to the positive or negative class respectively. Weak rules are known as *abstaining weak rules* if they abstain from contributing to the prediction score $F_T(x)$ of an example x when the weak rule evaluates to “false” for that example.

We now describe the details of obtaining h_t and calculating α_t and w_i at each iteration.

At iteration T , the exponential loss function to be minimized is given by

$$L_T = \sum_{i=1}^m \exp(-y_i F_T(x_i)) \quad (3.2)$$

$$= \sum_{i=1}^m \exp\left(-y_i \sum_{t=1}^T \alpha_t h_t(x_i)\right) \quad (3.3)$$

$$= \sum_{i=1}^m \left(\exp\left(-y_i \sum_{t=1}^{T-1} \alpha_t h_t(x_i)\right) \cdot \exp(-y_i \alpha_T h_T(x_i)) \right) \quad (3.4)$$

$$= \sum_{i=1}^m w_i \cdot \exp(-y_i \alpha_T h_T(x_i)) \quad (3.5)$$

where w_i represents the weight of each example x_i at iteration $T - 1$ and is given by $w_i = \exp(-y_i \sum_{t=1}^{T-1} \alpha_t h_t(x_i))$. Let us define $W_0(h_T) = \sum_{i: h_T(x_i)=0} w_i$ i.e. the sum of weights of all examples on which the h_T abstains ($h_T(x_i) = 0$). Similarly, $W_+(h_T) = \sum_{i: h_T(x_i)=1, y_i=1} w_i$, represents the sum of all positively labeled examples for which $h_T(x_i) = 1$. $W_-(h_T) = \sum_{i: h_T(x_i)=1, y_i=-1} w_i$, represents the sum of all negatively labeled examples for which $h_T(x_i) = 1$.

The loss function can thus be simplified as:

$$L_T = W_0(h_T) + W_-(h_T) \cdot \exp(\alpha_T) + W_+(h_T) \cdot \exp(-\alpha_T) \quad (3.6)$$

Solving for α , L is minimized when

$$\alpha_T = \frac{1}{2} \ln \left(\frac{W_+(h_T)}{W_-(h_T)} \right) \quad (3.7)$$

Substituting in Eq. 3.6 we get

$$L_T = W_0(h_T) + 2 \sqrt{W_+(h_T) \cdot W_-(h_T)} \quad (3.8)$$

Hence, Adaboost begins by initializing the weights of all m examples to $1/m$.

For any weak rule h , we define

$$\begin{aligned} W_0(h) &= \sum_{i:h(x_i)=0} w_i \\ W_+(h) &= \sum_{i:h(x_i)=1, y_i=1} w_i \\ W_-(h) &= \sum_{i:h(x_i)=1, y_i=-1} w_i \end{aligned}$$

At every iteration t , the weak learner evaluates the loss function given by $L(h) = W_0(h) + 2 \sqrt{W_+(h) \cdot W_-(h)}$ for the entire set of weak rules $\{h\}$ and picks the weak rule h_t with the lowest loss. The coefficient for this weak rule is given by $\alpha_t = \frac{1}{2} \ln \left(\frac{W_+(h_t)}{W_-(h_t)} \right)$. The weights of the examples are then updated using $w_i = \exp(-y_i F_{t-1}(x_i))$.

Intuitively, at each iteration Adaboost increases the weights of examples misclassified by the prediction function and decreases the weights of examples correctly classified. In the following iteration, the weak learner is able to focus on the hard-to-classify examples and pick a weak rule accordingly.

One can prove that if the weak rules are all slightly correlated with the label, then the

strong rule learned by Adaboost will have a very high correlation with the label — in other words, it will predict the label very accurately. Freund and Schapire [100] prove that the training error at iteration T has an upper-bound given by

$$\text{training error} \leq \exp\left(-2 \sum_{t=1}^T (1/2 - \epsilon_t)^2\right) \quad (3.9)$$

where ϵ_t is the training error (fraction of misclassified training examples) of weak rule h_t . This shows that the training error drops exponentially fast. Adaboost has also been shown to have a bounded generalization error. It has been observed that the test error of the strong rule (percentage of mistakes made on test examples) continues to decrease even after the training error (fraction of mistakes made on the training set) reaches zero. This behavior has been related to the concept of a “margin”, which is simply the value $yF(x)$ [102]. While $yF(x) > 0$ corresponds to a correct prediction, $yF(x) > a > 0$ corresponds to a *confident* correct prediction, and the confidence increases monotonically with a . The performance of Adaboost depends primarily on the amount of training data. It also depends on the hypothesis space (set of weak rules). If the rules are too complex, Adaboost can overfit the training data.

3.3.2 Alternating decision trees

In machine learning, decision trees are commonly used to represent prediction rules that involve logical combinations of features. An *Alternating Decision Tree* (ADT) is a margin-based generalization of decision trees [31]. As shown in Figure 3.3 (A), an ADT consists of alternating levels of *prediction nodes* and *splitter nodes*. In terms of Adaboost, each splitter node represents a weak rule h_t and the associated prediction node below it represents its coefficient α_t . The tree is seeded with a prediction node α_0 that corresponds to the coefficient of a generic weak rule that always evaluates to 1. At every boosting iteration, a new splitter node together with its prediction node is introduced. The splitter node can be attached to

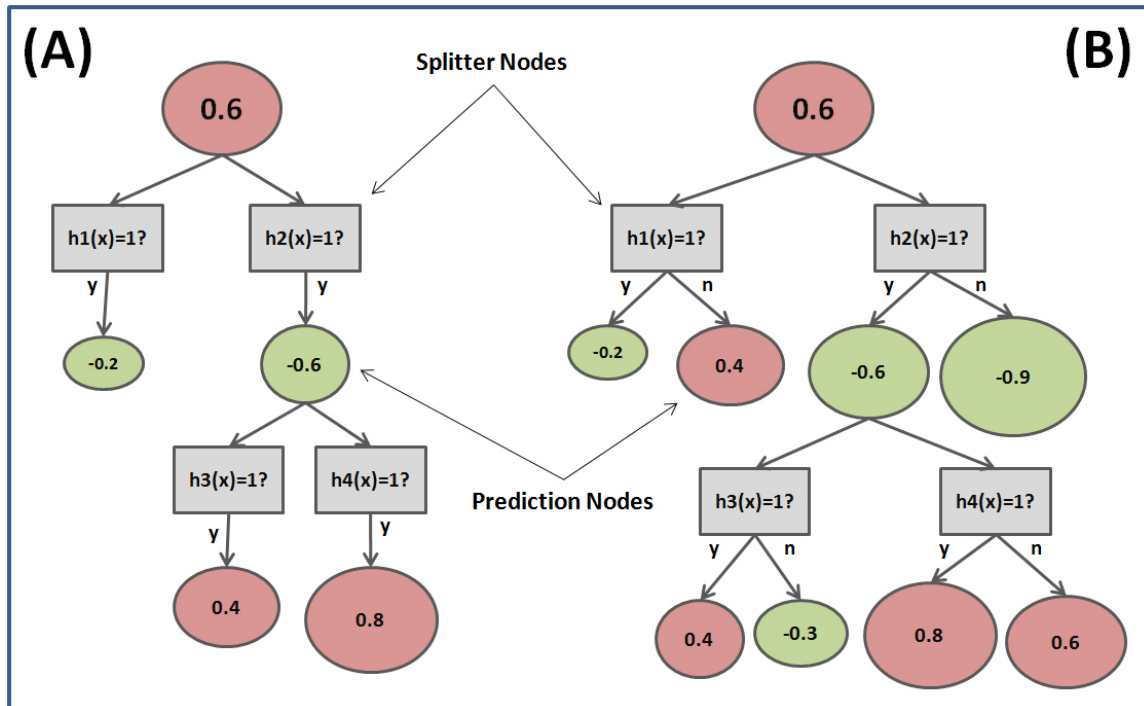


Figure 3.3: **Alternating decision trees:** ADTs are a margin-based generalization of decision trees that consist of alternating layers of splitter nodes (representing weak rules) and prediction nodes (representing the coefficient of each weak rule). Each path in the ADT consists of a conjunction of weak rules. (A) shows an ADT that uses abstaining weak rules which contribute to the prediction function only if the condition in the corresponding splitter node evaluates to “yes”. (B) shows an ADT that uses non-abstaining weak rules that contribute different values to the prediction function depending on the “yes” or “no” outcome of the condition in the corresponding splitter node.

any previous prediction node, not only leaf nodes. Hence, every splitter node is followed by a single prediction node. However, any prediction node can be followed by more than one splitter node.

Since the weak learner has to pick a weak rule and its position in the tree at every iteration, the search space grows according to the tree structure. The weak rule $h_t(x_i)$ can evaluate to 1 only for examples x_i that can reach the corresponding splitter node, and 0 for all other examples. A prediction node is said to be reachable if all weak rules in the path from the prediction node to the root node evaluate to 1. In order to calculate the value of the prediction function $F(x_i)$ for an example x_i , we sum the values in all the prediction nodes that are reachable. The predicted label for an example x_i is given by $\text{sign}(F(x_i))$. Intuitively, each splitter node in the tree consists of a question of the form “Is $h_t(x_i) = 1$ ”. If an example answers the question in the affirmative i.e. “yes” and the node is reachable, we add the value of the corresponding prediction node else we add nothing. Such weak rules are known as *abstaining weak rules*, since they abstain from contributing to the prediction score of an example if the condition in the corresponding splitter node is not satisfied by that example.

As shown in Figure 3.3 (B), ADTs can also contain *non-abstaining weak rules*. In this case, every splitter node is followed by two prediction nodes representing the “yes” and “no” outcomes of the condition. A non-abstaining weak rule contributes different values to the prediction function depending on whether the condition in the splitter node is satisfied or not.

A non-abstaining weak rule in a splitter node in the tree is equivalent to a pair of abstaining weak rules h_t and \hat{h}_t , such that $\hat{h}_t = 1 - h_t$ for all examples that can reach that splitter node and $\hat{h}_t = 0$ otherwise. The two prediction nodes represent two coefficients α_t and β_t for h_t and \hat{h}_t respectively.

Analogous to Equation 3.5, the loss function at iteration t can be expressed as

$$L_t = \sum_{i=1}^m w_i \cdot \exp(-y_i \alpha_t h_t(x_i) + \beta_t \hat{h}_t(x_i)) \quad (3.10)$$

$$= W((h_t = 0) \& (\hat{h}_t = 0)) + W_-(h_t) \cdot e^{\alpha_t} + W_+(h_t) \cdot e^{-\alpha_t} \quad (3.11)$$

$$+ W_-(\hat{h}_t) \cdot e^{\beta_t} + W_+(\hat{h}_t) \cdot e^{-\beta_t} \quad (3.12)$$

where $W((h_t = 0) \& (\hat{h}_t = 0))$ is the sum of weights of all examples that cannot reach the splitter node. $W_{\pm}(h_t)$ represent the sum of the weights of all examples with labels ± 1 that reach the splitter node and satisfy h_t . Similarly, $W_{\pm}(\hat{h}_t)$ represent the sum of the weights of all examples with labels ± 1 that reach the splitter node and satisfy \hat{h}_t .

Solving for α_t and β_t , L_t is minimized when

$$\alpha_t = \frac{1}{2} \ln \left(\frac{W_+(h_t)}{W_-(h_t)} \right) \quad (3.13)$$

$$\beta_t = \frac{1}{2} \ln \left(\frac{W_+(\hat{h}_t)}{W_-(\hat{h}_t)} \right) \quad (3.14)$$

Substituting back into Equation 3.12, we get

$$L_t = W((h_t = 0) \& (\hat{h}_t = 0)) + 2 \sqrt{W_+(h_t) \cdot W_-(h_t)} + 2 \sqrt{W_+(\hat{h}_t) \cdot W_-(\hat{h}_t)} \quad (3.15)$$

Learning regulatory programs: GeneClass

In this chapter, we present a novel classification-based algorithm called *GeneClass* which integrates regulatory sequence information and expression data to learn a genome-wide regulatory program that accurately predicts up/down regulation of genes in different experimental conditions. This chapter is based on work presented in [79], [80] and [62].

4.1 Introduction

We present an algorithm called GeneClass 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 sequence features – represented by the set of binding site patterns they contain (“motifs”) or transcription factor occupancy data from ChIP-chip assays – and the experiment-specific expression levels of a candidate set of regulatory proteins. The output is a prediction of the expression state of the regulated gene. Rather than trying to predict a real-valued expression level, we formulate the task as a binary classification problem, that is, we predict only whether the gene is up- or down-regulated. This reduction allows us to 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 present a robust variant of the

Adaboost algorithm that increases stability and computational efficiency, yielding a more scalable and robust predictive model. 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.

In computational experiments based on an yeast stress response and DNA damage datasets, we show that GeneClass predicts up- and down-regulation on held-out test data with high accuracy. We explore a range of experimental setups, and we retrieve important regulators, binding site motifs, and relationships between regulators and binding sites that are known to be associated to specific response pathways. Our method thus provides predictive hypotheses, suggests biological experiments, and provides interpretable insight into the structure of genetic regulatory networks. Finally, we present a detailed postprocessing framework for biological interpretation, including individual and group target gene analysis to reveal condition-specific regulatory programs and to suggest signaling pathways.

4.2 Related methods

Among recent statistical approaches, the most relevant method related to GeneClass is the REDUCE algorithm of Bussemaker *et al.* [20] for regulatory element discovery. Given gene expression measurements from a single microarray experiment and the regulatory sequence S_g for each gene g represented on the array, REDUCE proposes a linear model for the dependence of log gene expression E_g (or “motifs”) $E_g = C + \sum_{\mu \in S_g} F_\mu N_{\mu g}$, where $N_{\mu g}$ is a count of occurrences of regulatory subsequence μ in sequence S_g , and the F_μ are experiment-specific fit parameters.

REDUCE models the condition-specific activity of a motif by the experiment-specific

fit parameters F_μ . Thus, REDUCE is able to learn a prediction function over all genes in a single experimental condition using their sequence motif profiles. When presented with multiple experiments (microarrays), REDUCE learns multiple models, one for each experiment. On the other hand, GeneClass is able to learn a single global prediction function over all genes and all experimental conditions in a given dataset. In order to predict the expression levels of all genes in all experiments, we need our feature space to represent the gene context and the experiment context. We use regulatory sequence information such as regulatory sequence motif profiles and ChIP-chip data to represent the gene context. We use the expression levels of regulatory proteins to represent the experimental (cellular) context. In GeneClass, we infer condition-specific motif activity by associating motifs with regulatory proteins i.e. we learn paired (motif,regulator) features. The condition-specific expression levels of regulators thus model the condition-specific activity of associated motifs. The basic modeling assumption is that genes that have similar regulatory sequence features should have similar expression levels in experimental conditions that show coexpression of associated regulatory proteins. Due the linearity assumption, REDUCE is unable to model non-linear relationships between motifs. Since GeneClass learns a generalized decision tree as the regulatory model, it is able to model regulator and motif combinatorics more effectively. Also, while REDUCE learns a regression model, GeneClass discretizes gene expression data and uses classification instead.

Learning from a candidate set of potential regulatory proteins has also been used in the probabilistic model literature, including in the regression-based work of Segal *et al.* for partitioning target genes into *regulatory modules* for *S. cerevisiae* [104]. Here, each module is a coexpressed set of genes that is modeled as a probabilistic regression tree, where internal nodes of the tree correspond to states of regulators and each leaf node prescribes a normal distribution describing the expression of all the module's genes given the regulator conditions. The authors provide some validation on new experiments by establishing that the target gene sets of specific modules do have statistically significant overlap with the set

of differentially expressed genes; however, they do not focus on making accurate predictions of differential expression as we do here.

GeneClass retains the distinction between regulator genes and target genes, as well as a model that can capture combinatorial relationships among regulators; however, the margin-based GeneClass trees are very different from probabilistic trees. Unlike in [104], we learn from both expression and sequence data, so that the influence of a regulator is mediated through the presence of a regulatory sequence element. We note that in separate work, Segal *et al.* [105] present a probabilistic model for combining promoter sequence data and a large amount of expression data to learn transcriptional modules on a genome-wide level in *S. cerevisiae*, but they do not demonstrate how to use this learned model for predictions of regulatory response.

4.3 Learning regulatory programs as alternating decision trees

4.3.1 Feature space

A typical gene expression dataset measures the mRNA expression levels of several genes across different experimental conditions. We model the task of learning regulatory programs as a *classification problem* i.e. we try to learn a prediction function that can accurately classify the significantly upregulated data points (labeled +1) from the downregulated ones (labeled -1). We discretize the gene expression data into three levels: +1 representing significant up-regulation of expression, -1 representing significant down-regulation of expression and 0 representing no significant change in expression. Details of the discretization procedure are discussed in Section 4.5.2. Our goal is to learn a single global regulatory program that is able to predict differential gene expression for all genes in all experiments. Thus, every (gene,experiment) pair that is labeled +1 or -1 is used as a training or test

example in the learning procedure. The examples labeled 0 are not used in training since their labels are uncertain, in that the amount of up/down regulation is within the level of noise.

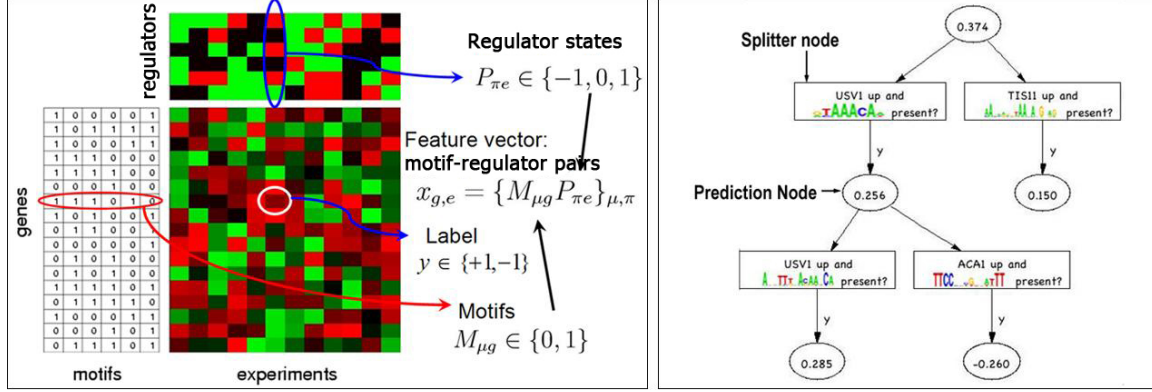


Figure 4.1: **Training data for GeneClass:** We show the data presentation for our GeneClass. Every (target gene, experiment) is assigned a label of +1 (up-regulated, in red) or -1 (down-regulated, in green) and represented by the genes vector of motif occurrences and the experiments vector of regulator expression states.

In order to predict the expression levels of all genes in all experiments, we need our feature space to represent the gene context and the experiment context. We start with a candidate set of M motifs $\{\mu\}$ representing known or putative transcription factor binding sites and a candidate set of R regulators $\{\pi\}$. Let $M_{\mu g} \in \{1, 0\}$ represent the presence or absence of a motif μ in the regulatory sequence of a gene g . Each gene g can then be represented by a vector $\{M_{\mu g}\}$ of motif occurrences. Let $P_{\pi e} \in \{-1, 0, 1\}$ represent the state of regulator π in an experiment e . The experimental context can then be represented by a vector $\{P_{\pi e}\}$ of the expression states of all the candidate regulators in that experiment. The feature vector for each training example x_{ge} is given by $\{\{M_{\mu g}\}, \{P_{\pi e}\}\}$. The hypothesis space (set of weak rules) on which the prediction function is defined can be written as $\chi = \{-1, 0, 1\}^R \times \{0, 1\}^M$. The set of weak rules represents all possible pairings of sequence motifs with regulators. Based on the nature of transcriptional regulation, we assume that a target gene's expression is dependent on the state of regulatory proteins and the presence or absence of sequence motifs. The data representation is depicted in Figure 4.1. The motif data can be replaced or augmented by ChIP-chip data that assays whether a gene's promoter

is bound by a particular regulatory protein.

4.3.2 GeneClass weak learner

GeneClass models regulatory programs as alternating decision trees (ADTs) (See Section 3.3.2). The nature of transcriptional regulation is such that different combinations of regulators and motifs regulate target genes in a context-specific manner. ADTs allow us to capture regulator and motif combinatorics. GeneClass uses the Adaboost algorithm, which we introduce in Sections 3.3.2 and 3.3.1, to learn the model. The algorithm maintains a weight distribution over the set of training examples. GeneClass iteratively calls a weak learner that picks a weak rule from a set of weak rules to minimize the exponential boosting loss. Examples are re-weighted at each iteration so that the algorithm is able to focus on hard-to-classify examples.

In GeneClass, the set of weak rules $\{h\}$ are boolean decision statements of the form “Is motif μ present in the regulatory sequence of a gene and is regulator π up-regulated in an experiment?” or “Is motif μ present in the regulatory sequence of a gene and is regulator π down-regulated in an experiment?”. For a particular (gene,experiment) example (g, e) , these statements are equivalent to “Is $M_{\mu g}P_{\pi e} = 1$?” or “Is $M_{\mu g}P_{\pi e} = -1$?”, respectively. We thus have a set of $2MR$ weak rules representing all possible motif-regulator pairs (μ, π_s) where the regulator π can be in state $s = \pm 1$. We do not use decision statements with regulators in state 0 based on the biological assumption that differential expression of a regulator triggers the differential expression of its targets.

The learning algorithm begins with a single prediction node that represents the class-bias in the dataset. At each iteration of boosting, the weak learner picks a weak rule and a position (prediction node) in the ADT to add the weak rule to. The position and (motif-regulator) pair (μ, π_s) are selected by minimizing the exponential boosting loss. The splitter nodes in our ADTs contain the the selected weak rules of the form “Is $M_{\mu g}P_{\pi e} = \pm 1$?”. Paths in the learned ADT correspond to conjunctions of these boolean (motif,regulator) conditions. It is

important to note that GeneClass does not restrict regulators to pair only with their DNA binding motifs. Hence, the relationship between a motif and a regulator is learned and not used a-priori in the learning algorithm.

We use ADTs with abstaining weak rules. As a consequence of the sparseness of the motif occurrences and discretized regulator expression, each weak rule evaluates false for a predominant part of the training data. By abstaining from predicting “no”, trained ADTs become shallower, and specific paths in the tree are statistically more significant and more easily interpretable in biological terms.

4.3.3 Predicting gene expression using a regulatory program

Figure 4.2 presents a simplified example to illustrate how a regulatory program generated by GeneClass computes a context-specific prediction score to predict the up/down regulation of target genes. In context *a* (Figure 4.2, top), the promoter of gene *a* contains a pair of sequence motifs associated by the regulatory program to a weak activator and a stronger repressor that are both expressed in the experimental condition. The regulatory program makes a moderately confident prediction that gene *a* will be downregulated, based on the sum of scores from the relevant pair of nodes in the tree. In context *b* (Figure 4.2, bottom), the promoter of target gene *b* contains binding sites for the weak activator but also for a co-factor, placed in a node below the weak activator in the tree. Both the activator and the co-factor are expressed in the condition shown, and the regulatory program computes a confident up prediction for gene *b* in this condition. In this way, GeneClass encapsulates a genome-wide and context-specific regulatory program, learned directly from motif data and expression data and without the introduction of additional prior assumptions.

4.3.4 Stabilized boosting

At each iteration, boosting adds the weak rule with the smallest exponential loss. The training examples are then reweighted such that they become decorrelated with the previously added

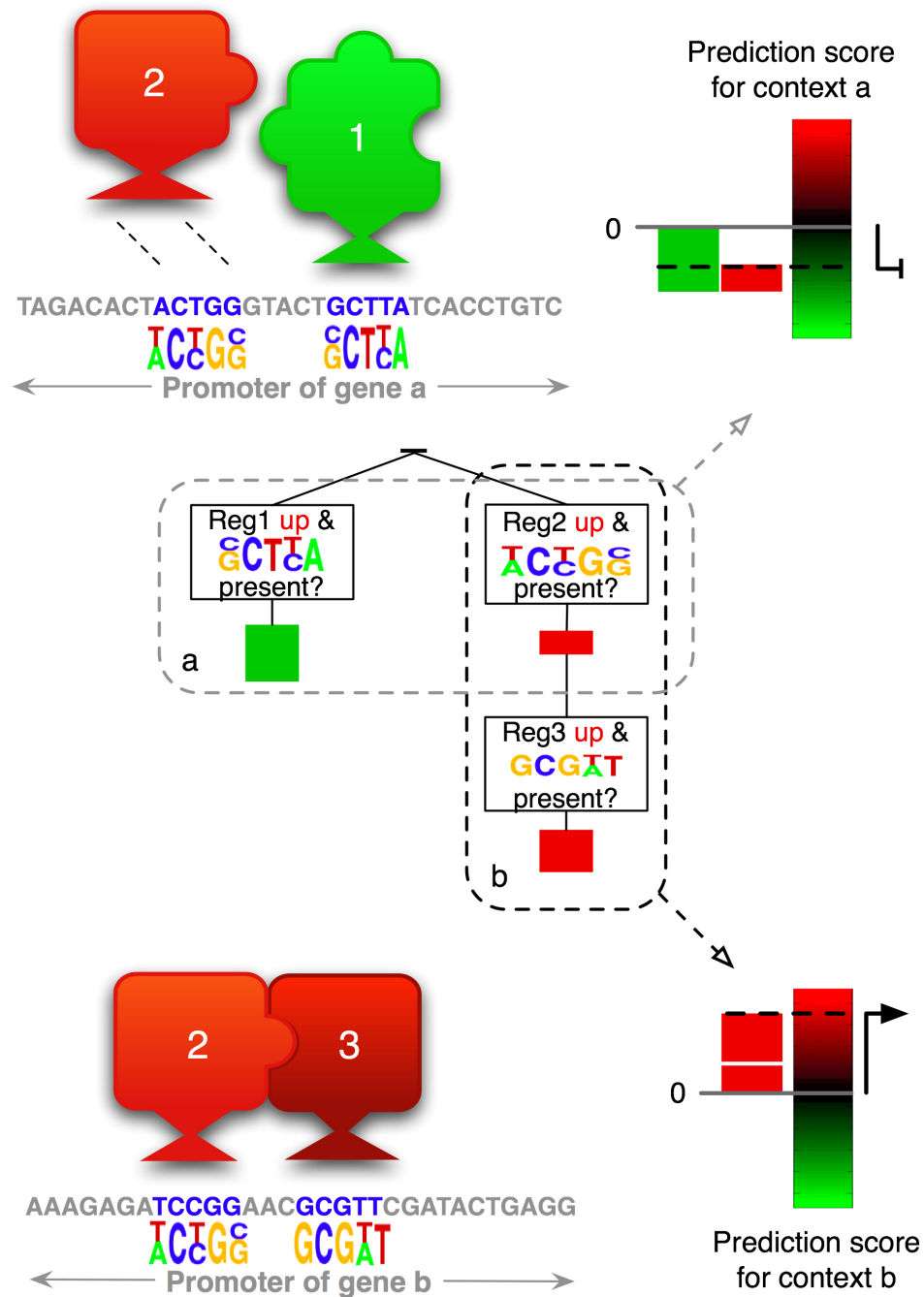


Figure 4.2: **Interpreting regulatory programs:** Simplified example showing how the regulatory program learned by GeneClass predicts context-specific up/down gene expression. GeneClass learns a global regulatory program described by an alternating decision tree. A simple regulatory program is shown, along with the prediction it makes in two contexts, indicated as context a (top-right) and context b (bottom-right).

rule. In a case where several weak rules are highly correlated with each other, only one of them will be added to the ADT. This means that correlated regulators and motifs might be missed. This presents an interpretability problem because important biologically meaningful features such as correlated regulators and motifs that co-occur in regulatory sequences could be missing in the ADT.

We solve this problem by averaging the prediction of several weak rules in the case where the rules with smallest boosting loss are highly correlated with each other. We determine whether the empirical correlation is statistically significant by comparing it with a threshold which is a function of the weights of the examples used for choosing the rules. As shown in the Section 4.6.6, this scheme not only finds biologically-meaningful, correlated features but also stabilizes the trees trained on different folds.

Let us consider two different weak rules h_1, h_2 . Let $A_{h_i} = \{\mathbf{x} | h_i(\mathbf{x}) = 1\}$ be the set of examples \mathbf{x} on which learner h_i predicts one. We define the symmetric difference of two sets of examples as the set of examples for which one but not both rules predict one i.e. the set of all examples \mathbf{x}_i for which $h_1(\mathbf{x}_i) + h_2(\mathbf{x}_i) = 1$ holds. The two weak rules, h_1 and h_2 , then have a highly correlated prediction if the total weight of the symmetric difference $A_{h_1} \ominus A_{h_2}$ is small. We denote this weight by $W(A_{h_1} \ominus A_{h_2})$.

In order to test the statistical significance of this correlation, we need to consider the distribution of the weights w_i of the examples in the two sets A_{h_1} and A_{h_2} . If the distribution is very skewed, small changes in the cardinality of the symmetric difference set can cause large changes in the corresponding weight. If the distribution is more uniform, then fluctuations in the size of the symmetrical difference set will cause appropriately scaled fluctuations in the weight of the symmetric difference set.

As shown in Appendix A, the function we use is motivated by Chernoff bounds [47]. We average over those weak rules h that obey

$$W(A_{h^*} \ominus A_h) \leq \eta_1 \sqrt{\frac{\sum_i w_i^2}{(\sum_i w_i)^2}} \quad (4.1)$$

where h^* is the weak rule with smallest loss at the considered boosting round, and $\eta_1 > 0$ is an empirically tuned parameter. The summation of weights is over all examples that reach the prediction node to which we want to add the stabilized weak rule. In the limit of equally weighted examples, this threshold evaluates to η_1 / \sqrt{N} where N is the total number of examples. For more skewed distributions the threshold becomes more permissive. This allows more weak rules to be averaged leading to a stable weak rule that is more resistant to fluctuations. In the extreme case, where all the weight is on a single example, the threshold evaluates to η_1 .

For a weak rule h , let us define $W_0(h) = \sum_{i, h(\mathbf{x}_i)=0} w_i$, $W_+(h) = \sum_{i, h(\mathbf{x}_i)=1, y_i=+1} w_i$, $W_-(h) = \sum_{i, h(\mathbf{x}_i)=1, y_i=-1} w_i$, where y_i is the label of example \mathbf{x}_i . We assume that the weights are normalized such that $W_0(h) + W_+(h) + W_-(h) = 1$. h has a corresponding coefficient $\alpha[h]$ given by

$$\alpha[h] = \frac{1}{2} \ln \frac{W_+(h)}{W_-(h)} \quad (4.2)$$

We average over a set of weak hypotheses $\{h\}$ by defining a new hypothesis h_{avg} that predicts 1 if all hypotheses have approximately equal coefficients $\alpha[h]$ and takes a majority vote over the set of hypotheses if about half of them predict 1. Otherwise, it predicts 0.

$$h_{\text{avg}}(x_i) = \begin{cases} 1, & |\sum_h \alpha[h] h(x_i)| > \theta \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

where we set

$$\theta = \frac{(\sum_h |\alpha[h]| - \min_h |\alpha[h]|)}{2} \quad (4.4)$$

The coefficient of this new weak rule is given by

$$\alpha[h_{\text{avg}}] = \frac{1}{2} \ln \left(\frac{W_+(h_{\text{avg}})}{W_-(h_{\text{avg}})} \right) \quad (4.5)$$

In some cases, we may obtain several highly correlated weak rules that have very low

predictive strength. These would most likely be the result of noise. In such a case, we want to avoid averaging weak rules. Thus, our algorithm abstains from stabilization if the weighted loss of the weak rule h^* is so close to $1/2$ that interpretability of the selected feature is questionable, even though the overall classification performance might still be improving [102].

The coefficient of the weak rule h^* is given by $\alpha[h^*] = \frac{1}{2} \ln \left(\frac{W_+(h^*)}{W_-(h^*)} \right)$. $\alpha[h^*]$ will be positive when $W_+(h^*) > W_-(h^*)$ and negative otherwise. Hence, h^* will correctly predict the weights of examples with total weight $\max(W_+(h^*), W_-(h^*))$. A weighted loss that approaches random guessing ($1/2$) can be defined as

$$\frac{1}{2} W_0(h^*) + \min(W_+(h^*), W_-(h^*)) \quad (4.6)$$

$$= \frac{1}{2} (1 - W_+(h^*) - W_-(h^*)) + \min(W_+(h^*), W_-(h^*)) \quad (4.7)$$

$$= \frac{1}{2} - \frac{(W_+(h^*) + W_-(h^*))}{2} + \min(W_+(h^*), W_-(h^*)) \quad (4.8)$$

$$= \frac{1}{2} - \frac{1}{2} |W_+(h^*) - W_-(h^*)| \quad (4.9)$$

The advantage over random guessing is thus $\frac{1}{2} |W_+(h^*) - W_-(h^*)|$. We want this term to be large. Hence, we introduce another threshold η_2 and perform stabilization only if

$$\frac{1}{2} |W_+(h^*) - W_-(h^*)| \geq \eta_2 \sqrt{\frac{\sum_i w_i^2}{(\sum_i w_i)^2}} \quad (4.10)$$

For all experiments in this thesis we use $\eta_1 = \eta_2 = 0.1$.

The pseudocode for the GeneClass algorithm is presented in Appendix B.

4.4 Extracting predictive features from regulatory programs

In this section, we introduce a post-processing framework for extracting context-specific regulatory features from our learned models, that answer specific biological queries. GeneClass

learns a global regulatory program over all genes and all experiments in a dataset. Each node in the ADT consists of a weak rule defined on a set of motif-regulator pairs. Note that the association of a motif and a regulator in a weak rule does not necessarily imply a direct binding relationship between the regulator and the motif. Such a pair could represent an indirect regulatory relationship such as an upstream signaling regulator regulating targets through another transcription factors binding site. It could also represent co-occurrence of the true binding site of the regulator with another motif.

4.4.1 Extracting global features

In order to identify regulators and motifs that are globally predictive over all examples, we introduce two scoring metrics for ranking motifs and regulators.

The *iteration score (IS)* of a weak rule (motif-regulator pair) is the boosting iteration during which it first appears in the ADT. Weak rules that are learned in early boosting rounds tend to predict on large sections of the data. Hence, weak rules with low IS, tend to be globally significant.

We define the *abundance score (AS)* of a regulator/motif as the number of splitter nodes in the ADT that include the regulator/motif as part of its weak rule. A regulator/motif with a large abundance score will affect a large number of paths through the ADT and hence affect a large number of target genes. If the state of a regulator is changed, its predicted effect on target genes will depend on its abundance in the ADT.

4.4.2 Gene set analysis: Extracting context-specific features

A biologist would also be interested in identifying regulators and motifs that regulate different subsets of genes in various subsets of experiments. Below, we present scores to rank regulators and motifs relevant to regulation of specific gene sets.

To rank motifs and regulators that are predictive of 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) examples (g, e) in question. We then rank motifs and regulators using AS and IS in the extracted subtree.

To study the regulation of a gene set in multiple experiments, we rank motifs and regulators using a *frequency score (FS)*. The frequency score for a motif/regulator over a set of $B = \{(g, e)\}$ examples is defined as the number of correctly predicted examples in B that pass through all splitter nodes containing the motif/regulator.

4.4.3 Signaling pathways and regulatory cascades

Different signaling pathways are activated under different experimental conditions, and these highly interconnected pathways affect regulation via activation or repression of sets of transcription factors. Since many kinases and phosphatases are auto-regulated or are in tight positive and/or negative feedback relationships with the transcription factors that they regulate [37], we hypothesize that mRNA levels of signaling molecules in particular pathways might be predictive of expression patterns of targets genes of downstream transcription factors. We use two methods to identify regulatory cascades. First, we use gene set analysis on individual transcription factors to identify predictive regulators that might be act upstream. Second, we use ChIP-chip data to identify regulators that co-associate with transcription factor binding profiles in high ranking weak rules.

4.5 Datasets

4.5.1 Expression data

We present analysis of two gene expression datasets: The environmental stress response (ESR) dataset [37], consists of 173 dual-channel cDNA microarray experiments measuring the genome-wide response (6110 genes) of the yeast (*S. cerevisiae*) to 13 different environmental stresses. These include heat shock, hyper-osmolarity, hypoosmolarity and

simultaneous heat shock and osmotic stress; peroxide stress; oxidative stress due to menadione, diamide and dithiothreitol (DTT), amino acid starvation, nitrogen starvation, diauxic shift, entry into stationary phase, steady state growth and growth on alternative carbon sources.

The DNA damage dataset [36], consists of 53 experiments measuring expression patterns (6167 genes) of wild-type and knock-out mutant yeast (*S. cerevisiae*) cells exposed to various DNA damaging agents.

Both datasets also include control experiments in which both channels in the microarray assay the gene expression of replicate samples. We use these arrays to empirically estimate the noise in the dataset (See Section 4.5.2). We use the noise model to discretize the expression data.

All measurements are given as \log_2 fold changes with respect to a reference expression value.

4.5.2 Discretization of expression data

To model the learning problem as a classification task, we 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). The noise model incorporates the dependence of noise on fluorescence intensities.

In order to estimate the null model, we use control experiments [36] for the ESR and the DNA damage datasets, in which both channels of a microarray assay replicate samples. Let R and G represent the values for the red (Cy3) and green (Cy5) channels for each spot (gene) in the microarray. For single channel microarrays, R and G represent expression levels of a gene in replicate experiments. We plot the log ratio of the two channels $M = \log_2(\frac{R}{G})$ versus the average log-intensity $A = \log_2(\sqrt{RG})$ for all genes, as shown in Figure 4.3). This displays the intensity specific distribution of the noise in log-scale. Ideally all these points

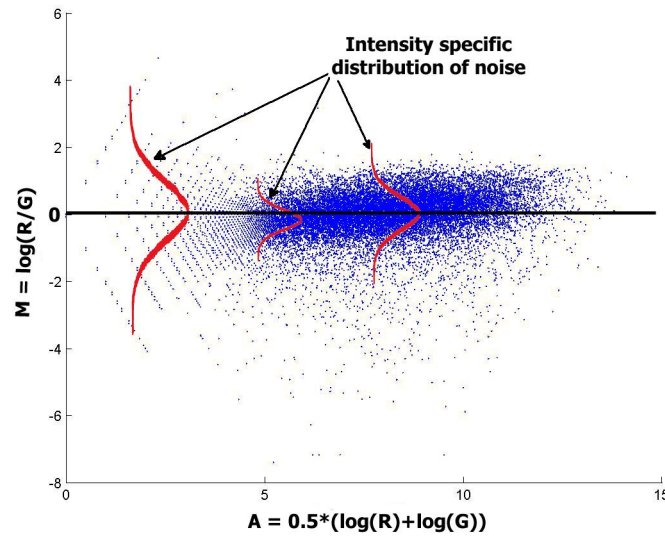


Figure 4.3: **Noise model for discretizing data:** We plot the log ratio of the two channels M versus the average log-intensity A for all genes in microarrays assaying replicate samples. We use this to develop an average intensity-dependent noise model to discretize expression data.

should have a value of 0 for the M axis and any deviations are due to noise. It is clear that the noise is higher at lower intensities.

We compute the cumulative empirical null distribution of M conditioned on A by binning the A variable 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. The noise model is based on work of Tu *et al.* [119].

4.5.3 Candidate set of regulators

Our candidate set of regulatory proteins consists of 475 genes consisting of 237 known and putative transcription factors and 250 known and putative signaling molecules, with an overlap of 12 genes of unknown function. Of these, 466 are from Segal *et al.* [104] and 9 generic (global) regulators are obtained from Lee *et al.* [68].

4.5.4 Motif data

The TRANSFAC database [128] provides a library of known and putative transcription factor binding sites, some of which are represented by position-specific scoring matrices (PSSMs) and consensus sequences. In order to identify target genes of these motifs, we obtain the 500 bp upstream promoter sequences of all *S. cerevisiae* genes from the Saccharomyces Genome Database (SGD). For each of these sequences, we search for transcription factor binding sites using the PATCH software licensed by TRANSFAC [128]. A total of 354 binding sites are used after pruning to remove redundant and rare sites.

We also use motif data provided by Pilpel *et al.* [92]. 356 PSSMs are obtained using AlignACE [53] which is a “cluster-first” motif-discovery method based on detecting over-represented sequence patterns in promoter sequences of different gene sets. These PSSMs are matched to promoters of 5651 genes in the genome using ScanACE [53].

4.5.5 ChIP-chip data

Lee *et al.* [68] use 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 yeast 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 [68] is used to assign p -values to these measurements. We use the ChIP-chip data as a binary “motif” matrix by thresholding the p -values, so that each target gene’s motif vector is replaced or augmented by a transcription factor occupancy vector for the set of transcription factors. We tried different thresholds of 0.001, 0.05 and 0.1 and found the best prediction accuracy on test data with a p -value threshold of 0.1. Although, this value might seem very permissive, recent work has shown that weak binding of transcription factors has a significant effect on gene regulation. Lee *et al.* [68] suggest to use a p -value of 0.005 in order to reduce false positives. However, this causes a large number of false negatives.

GeneClass has no way of accounting for lack of motif data. However, it can potentially filter out false positives. Hence, we use a more permissive p -value to allow a lower number of false-negatives.

4.6 Statistical validation

4.6.1 Prediction accuracy in cross-validation experiments

In order to assess the predictive ability of our algorithm we perform 10-fold cross-validation on the datasets 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 average the test-loss (percentage errors in prediction) over the 10 folds.

When we use the motif list from the TRANSFAC database [128] with the set of 475 candidate regulators for the ESR dataset, we get an average test loss of $20.8\% \pm 2.8\%$. The AlignACE [92] motif-data gives us an average test-loss of $16.2\% \pm 4.0\%$ on the ESR data set. This result clearly shows that the set of hand-curated motifs in the TRANSFAC database are incomplete and motifs learned using motif-discovery tools have better predictive performance.

On using the motif data from AlignACE [92] with the DNA damage dataset, we obtain an average test loss of $23.4\% \pm 6.3\%$. The DNA damage dataset is an older dataset than the ESR dataset where the observed noise levels are higher. The dataset also contains several knockout phenotypes which are known to have secondary responses.

4.6.2 Motif data versus ChIP-chip data

In order to study the effect of the different types of regulators and motif data, we use a single random held-out test set consisting of one tenth of the examples labeled ± 1 in order to get an estimate of test error. We compare performance of the AlignACE [92] motif set against

ChIP-chip data. We also compare prediction accuracy when we use only transcription factors (TF), only signaling molecules (SM) and all regulators as the candidate set of regulators. The experiments are summarized in Table 4.1. Results show that the motif data outperforms the ChIP-chip data and the different sets of candidate regulators give similar test errors.

Experiment	Motif data	Regulators	Targets	Error on ESR	Error on DNA damage
<i>ChIP+all</i>	ChIP	475 SM+TF	6102	17.29%	23.56%
<i>ChIP+SM</i>	ChIP	250 SM	6102	16.7%	23.33%
<i>AlignACE+all</i>	AlignACE motifs	475 SM+TF	5579	12.7%	18.87%
<i>AlignACE+TF</i>	AlignACE motifs	237 TF	5579	13.7%	20.34%
<i>AlignACE+SM</i>	AlignACE motifs	250 SM	5579	14.11%	19.58%

Table 4.1: Different experimental setups and their performance

4.6.3 Prediction scores show correlation with expression data

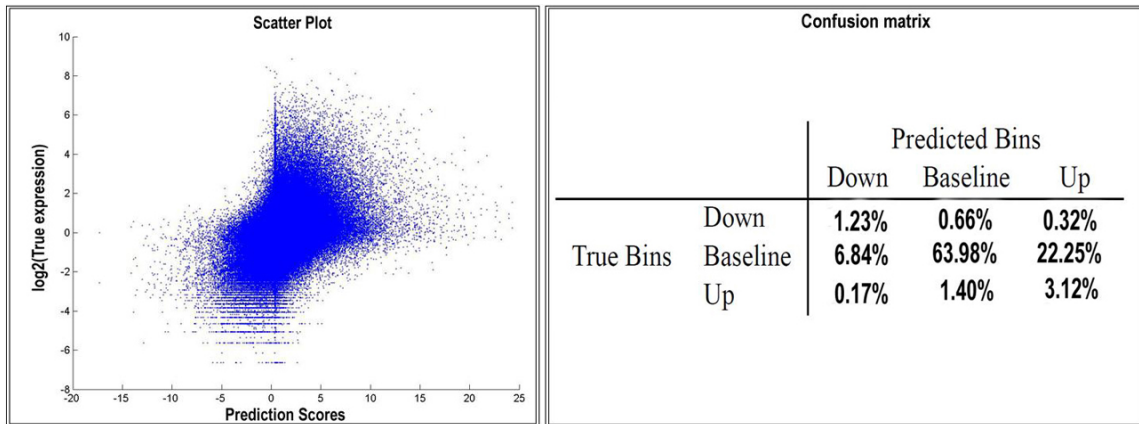


Figure 4.4: **Correlation of prediction scores with expression data** (Left) The scatter plot shows the correlation between prediction scores (x-axis) and true log expression values (y-axis) for genes on held-out experiments. (Right) Confusion matrix: truth and predictions for all genes in the held-out 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.

Although, GeneClass is a binary classification algorithm, the output is a real-valued prediction function for all genes and experiments in the form of an alternating decision

tree (See Section 3.3.1). The sign of the prediction score gives the predicted label and the absolute value represents a confidence level for the prediction.

Figure 4.4 (Left) shows real-valued expression data versus prediction scores for all examples (up, down, and baseline) from the held-out experiments using 10-fold cross-validation on the entire ESR data set, where baseline examples are randomly divided among the 10 folds. 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.

Let $F(x_{ge})$ represent the prediction function for any gene-experiment example (x_{ge}, y_{ge}) with feature vector x_{ge} and label $y_{ge} \in \{-1, 0, 1\}$. We can make 3-class predictions by thresholding on the confidence levels of up and down predictions, that is, we predict examples to be up- or down-regulated if $F(x_{ge}) > a$ or $F(x_{ge}) < -b$, and to be baseline if $-b \leq F(x_{ge}) \leq a$ where $a, b > 0$. By assigning thresholds to expression and prediction scores ($a = 0.95, b = 0.2$), we bin the examples into up, down and baseline to obtain the confusion matrix in Figure 4.4 (right). We see a good separation between classes, represented by the strong diagonal elements in the confusion matrix. 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.

4.6.4 Comparison to a baseline classification method

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%).

4.6.5 Randomization experiments

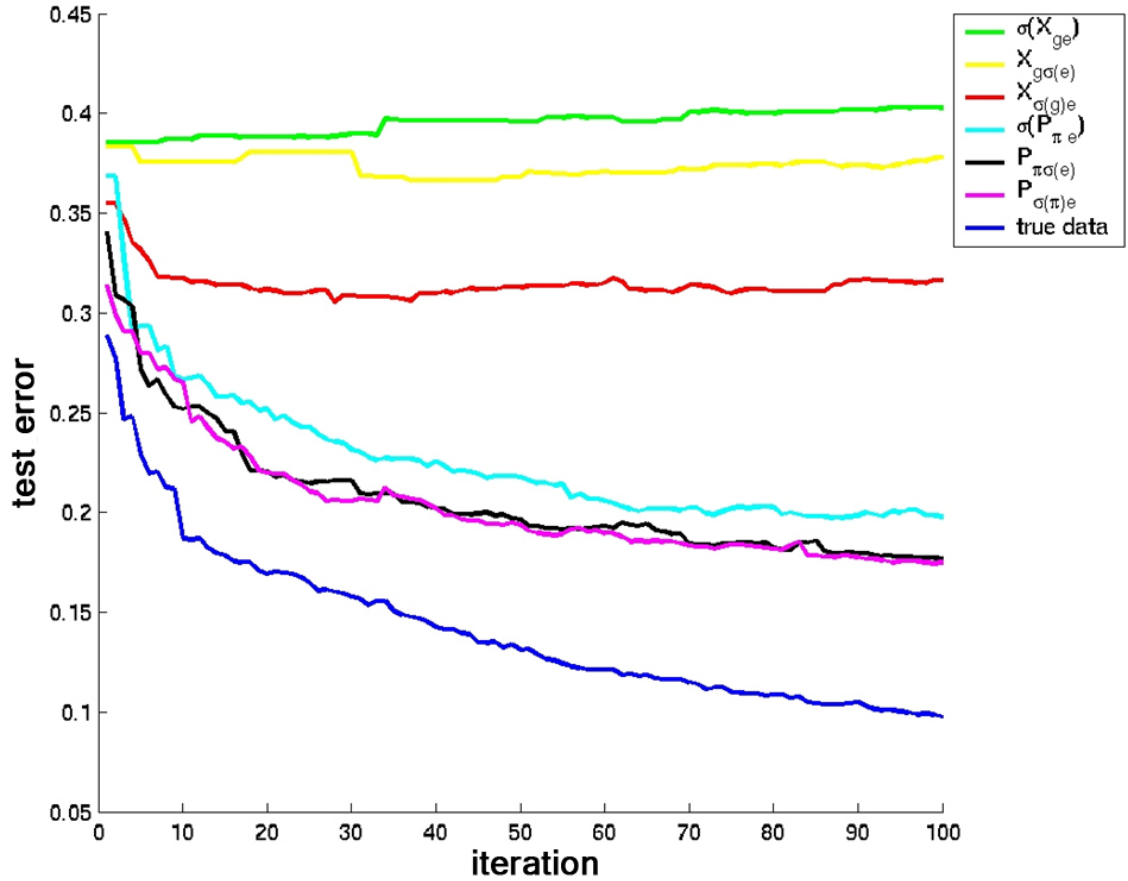


Figure 4.5: **Test error for GeneClass with randomized data:** The figure shows that test error for GeneClass runs using randomized target gene expression data, randomized regulator expression and randomized motif data are significantly higher than experiments on true data.

In order to evaluate the predictive relevance of the regulator expression data and the

motif data, we use GeneClass to learn models using partially randomized version of these data and compare the test error to models learned using the true data. Figure 4.5 shows six sets of randomization experiments on a subset of 1400 high variance target genes in the heat shock experiments of the ESR dataset. The blue curve labeled “true data” shows the test error for the actual expression and motif data. We see that the test error decreases exponentially with each boosting iteration. We now describe each of the randomization experiments.

Let σ represent a randomization function. Given a target gene expression matrix, we can randomize the expression data in three ways i.e. shuffle all entries in the matrix or shuffle only rows (genes) or columns (experiments). The curves labeled $\sigma(X_{ge})$ (green), $X_{g\sigma(e)}$ (yellow) and $X_{\sigma(g)e}$ (red) show the test error for these three experiments. We clearly see that randomizing the labels leads to overfitting behavior i.e. after an initial drop, the test error begins to increase with the number of iterations. However, it is interesting to note that the test loss is more sensitive to shuffling experiments than shuffling genes.

Similarly to assess the effect of regulator expression on predictive performance, we randomize the regulator expression matrix in three ways. For the curve labeled $\sigma(P_{\pi e})$ (light blue) we shuffle all the values in the regulator expression matrix. For the curve labeled $P_{\pi\sigma(e)}$ (black) we shuffle columns of the regulator expression matrix. For the curve labeled $P_{\sigma(\pi)e}$ (magenta) we shuffle rows of the regulator expression matrix. We see that in all three cases, the algorithm is able to learn with reasonable accuracy. However, the test error is still significantly poorer than experiments using true data. This shows that information provided by the motif data is sufficient but not adequate to predict expression.

4.6.6 Stabilized boosting results in robust ADTs

Table 4.2 shows how GeneClass stabilizes trees trained on different folds. We rank regulators based on an iteration score (IS) which is the boosting iteration at which the regulator first appears in the ADT. We compare the 20 top-ranking regulators for 10-fold cross-validation

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

Table 4.2: **The effect of stabilized boosting on ranking of predictive features:** Top ten parents ranked by iteration score (IS) for alternating decision trees learned with and without stabilization. The stabilization uses parameters $\eta_1 = 0.01$ and $\eta_2 = 0.03$

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 regulators (rank > 6). By including more complete information about predictive features, we obtain more stable and interpretable trees.

We also find that using abstaining weak rules instead of non-abstaining weak rules leads to a 4-fold reduction in running time on the ESR dataset for 1000 iterations without a significant change in the prediction accuracy. For abstaining weak rules, only a single prediction node is added to the ADT at each boosting iteration. For non-abstaining weak rules, two prediction nodes are added at each iteration. Hence, the search space is reduced by half for the abstaining case. Abstaining also leads to shallower and more interpretable trees.

4.7 Biological validation

We use the different feature extraction methods and scores introduced in Section 4.4 to identify predictive regulators and motifs. Below, we present a few illustrative examples.

4.7.1 Globally predictive regulators and motifs

For the ESR dataset, the first weak rule added to the tree contains the STRE motif and a regulator Tpk1. The STRE motif is the binding site of the *MSN2/4* transcription factor which is known to be the most important general stress response regulator and affects the expression levels of over 900 genes [37]. It is known that the transcriptional activity of the Msn2/4 protein is regulated by the protein-kinase Tpk1 via cellular localization. *MSN2/4*'s expression levels are found to not change significantly as it is mainly regulated post-transcriptionally. Thus, we see an interesting indirect relationship between the motif

and its associated regulator.

While the binding sites of some very important stress factors like Msn2/4 and Hsf1 (heat shock factor) have high iterations scores in the ADT, the mRNA expression levels of these regulators do not seem to be very predictive. Hsf1 does not appear as a regulator in the tree and Msn2/4 gets low abundance and iteration scores as a regulator, despite their importance as heat-shock and general stress response regulators respectively. Similar results are observed in the modules of Segal *et al.* [104], where Hsf1 is not found in any of the regulatory programs and Msn2/4 is found in only three of the fifty regulatory programs but with low significance. We find that the mRNA expression levels of *MSN2* and *HSF1* are mostly in the baseline state within the limits of experimental noise due to which it is difficult to identify these as important regulators based simply on their expression profiles. Thus, we see the advantage of using complementary sources of regulatory information namely motif data along with mRNA expression levels of regulatory proteins.

4.7.2 Regulators of functionally related genes

It has been observed that a subset of approximately 26 protein folding chaperones is induced by a variety of stress conditions [37]. The Hsf1 transcription factor along with Msn2/4 are known to be the prime regulators for this set of genes. We first use the gene set analysis framework to analyze this set of genes across all 173 ESR experiments. We then analyze specific sets of experiments — *MSN2/4* deletion mutants, *MSN2* and *MSN4* over-expression mutants, *YAP1* deletion mutant and *YAP1* over-expression mutant— as well as study the regulatory machinery in opposite polarities of heat shock, i.e. temperature increase versus temperature decrease (see Figure 4.6).

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). We find Cmk2 and Slt2 among the top scoring regulators. 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

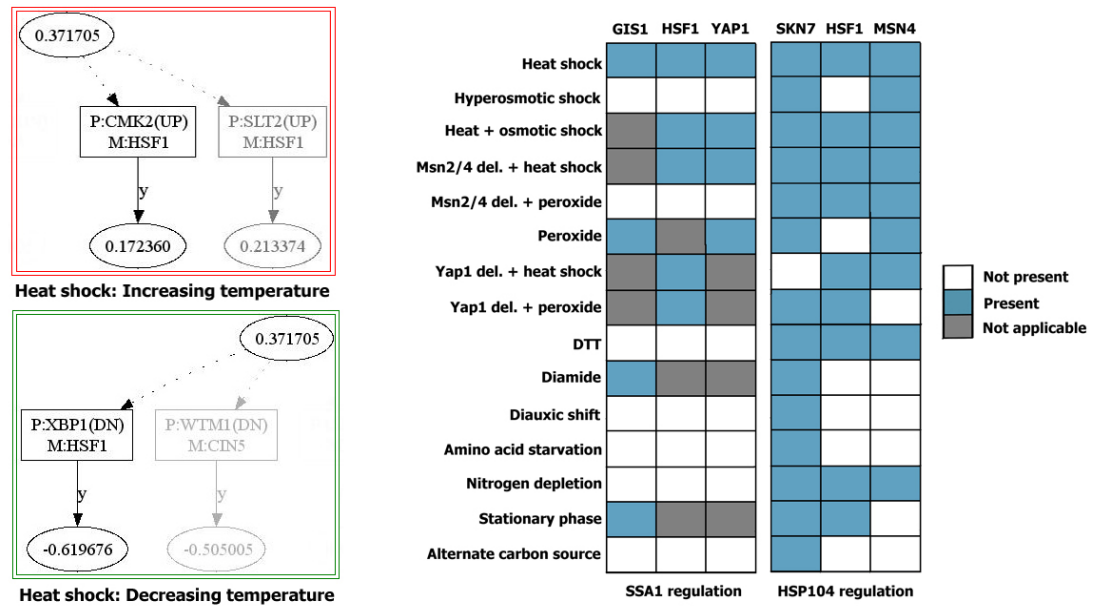


Figure 4.6: **Regulation of protein-folding chaperones** (Left) Comparison of trees for response of heat shock proteins to increasing and decreasing heat shock. Xbp1 and Wtm1 are both repressors. (Right) Condition-specific regulation of SSA1 and HSP104. The grey squares indicate that the target gene was in the baseline state for those experiments.

cell growth and response to nutrient availability [137]. In both experimental setups, Slt2 is found associated in a motif-regulator 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 binding occupancy data. In mammalian cells, CaMKII which is an ortholog of Cmk2 has been found to significantly affect Hsf1 function [49], and the association between Cmk2 and the Hsf1 motif might indicate a similar relationship in yeast. Other high scoring regulators include Usv1 and Tpk1. The high scoring Msn2/4 motif is found to be associated with regulators Slt2 and Ptp2, the latter being part of the HOG MAPKinase pathway. Ptp2 can inactivate Slt2 via phosphorylation and a Ptp2 mutant has been found to be hyper-sensitive to heat [82]. Msn2/4 could thus be a downstream target of pathways involving these signaling molecules. A weak rule containing Tpk1 as the regulator and Skn7 binding occupancy is 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. We note that neither Hsf1 nor Msn4 are found to be high scoring

regulators. Without the use of motif data we would not be able to identify these as the key regulators of this set of protein folding chaperones.

We also use the gene set analysis framework in sets of experiments consisting of specific stresses, again ranking features by frequency score, to examine regulatory phenomena unique to these stress responses. Figure 4.7 shows the regulators and motifs that are predictive of the differential expression of the heat shock genes in the simultaneous heat and osmolarity shock experiments. The predictive regulators 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 weak rule containing the Snf3 regulator and Hap4 binding occupancy data to be the highest scoring feature. Snf3 is part of the glucose sensor family and Hap4 is a transcription factor involved in regulating growth in non-fermentable carbon sources [96]. Similarly, the Ptp2-Msn4 regulator-motif pair is particularly prominent in the hyperosmotic stress indicating possible activation of the HOG1 pathway. Skn7 and Hsf1 have been shown to induce several heat shock proteins in response to oxidative stress [94]. We observe this phenomenon in the features extracted for response to oxidative stress due 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 regulators. Both these regulators control exit from mitosis, and Clb2 (necessary for G2 repression of the SCB factor) is 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 for gene sets.

4.7.3 Regulation of individual target genes

We now focus on regulation of two specific heat shock genes, *SSA1* and *HSP104*. Interesting aspects of condition-specific regulation, are summarized in Figure 4.6 (right). 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

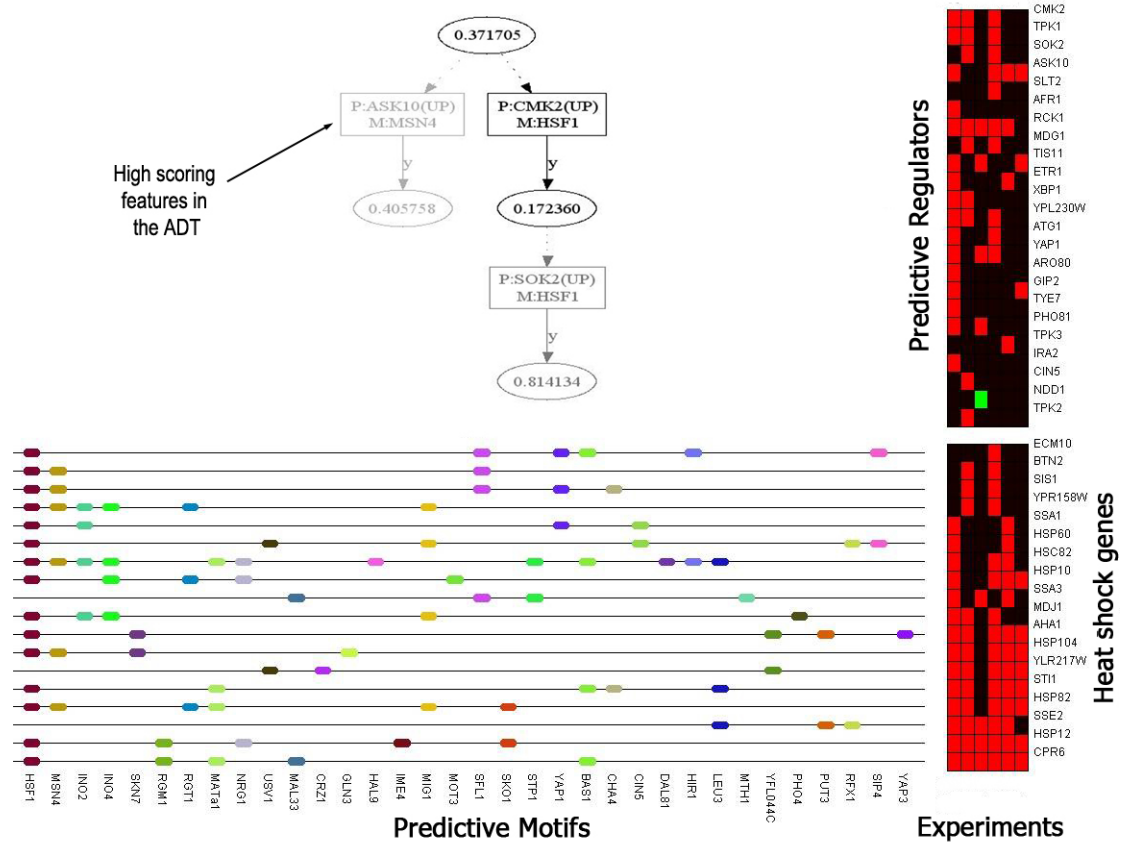


Figure 4.7: Regulation of heat shock proteins in heatshock and osmotic stress The figure shows the regulatory motifs and regulators that are predictive of the expression of the heat shock proteins in the simultaneous heat and osmolarity shock experiments. The bottom right rectangle represents the discretized 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 regulators. The regulators 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.

in experiments involving 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 in the diamide response and stationary phase response. Gis1 is known to regulate some heat shock proteins [85]. It is not known if Gis1 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 Msn4, Hsf1 and Skn7 binding sites in its upstream region and appears to be actively regulated by these transcription factors in a stress specific manner. All three factors appear to jointly control regulation in almost all stresses. The exceptions are hyper-osmotic stress and peroxide stress where only Skn7 and Msn4 seem to be active and the response to stationary phase induction where Skn7 and Hsf1 seem to be active.

4.7.4 Identifying signaling pathways

We focus on the Hsf1 transcription factor and extract the signaling molecules that are predictive of its activity and its targets. We extract all signaling molecules that associate with Hsf1 binding occupancy as regulators in the entire regulatory program. We find an important section of the PKA signaling pathway (Tpk1, Bcy1, Pde1, Yak1) as well as parts of the PKC pathway (Wsc4, Slf2 and Cdc28). We also find Sds22, Gip2 and Gac1, all of which are subunits of the Glc7p protein phosphatase, which has been identified as an Hsf1p binding protein [72].

4.7.5 In silico knockouts to identify transcription factor targets

The target genes of several transcription factors and regulatory proteins are unknown and so it remains an interesting topic of research to identify their putative functions. A common wet lab technique that is used to understand the global regulatory effect of a particular gene is to create a mutant (a knockout) in which the protein of interest is not in a functional state. The genome wide response of the knockout is compared to that of the normal wild-type

strain and genes that change their expression significantly are identified as potential primary or secondary targets of a regulator. By analyzing the functions of the targets, one can get a rough estimate of the biological processes that the protein of interest might regulate. We decided to try a computational analog of a wet lab knockout to check if we could recover functional information about predictive regulators.

In the GeneClass framework, by removing a candidate from the regulator list and retraining the ADT, we can evaluate whether test loss significantly increases with omission of the regulator and identify other weaker regulators that are also correlated with the labels. We perform an *in silico knockout* of the regulator Usv1 in the heat shock experiment, and observed a small but significant increase of 4% in test error. Regarding structural changes in the ADT, we observe that the overall hierarchy of the features does not change significantly: Tpk1, Xbp1, Ppt1 and Gis1 remain the highest scoring regulators. However, we find that on retraining the ADT, we make errors on 305 target genes whose expression profiles were previously correctly predicted. We use the Gene Ontology database and identify functional terms that are enriched in this set of genes using a hypergeometric p -value. The significant terms include cell wall organization and biogenesis, heat-shock protein activity, galactose, acetyl-CoA and chitin metabolism and tRNA processing and cell-growth. These match many of the terms enriched by analyzing Gene Ontology annotations of genes that changed significantly in a microarray experiment by Segal *et al.* [104] with stationary phase induced in a true *USVI* knockout [104].

4.8 Conclusions

Our work on the 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, may tell 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 [87] 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 [86]. 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. We present a stabilized version of boosting to increase the stability of features included in the prediction tree and to enable the detailed target gene analysis that we present in our post-processing framework. As we show in Table 4.2, 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. 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. The accuracy/interpretability trade-off in 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 the 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. The post-processing method described in this chapter addresses condition-specific regulation by extracting and analyzing subtrees corresponding to related sets of experiments.

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. 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.

Learning cis regulatory motifs: MEDUSA

In this chapter, we present MEDUSA, an integrative method for learning motif models of transcription factor binding sites by incorporating promoter sequence and gene expression data. This chapter is based on work presented in [78].

5.1 Introduction

One of the central challenges in computational biology is the elucidation of mechanisms for gene transcriptional regulation using functional genomic data. The identification of binding sites and targets of transcription factors is a key component in these computational efforts.

In Chapter 4, we introduce a predictive framework for modeling gene regulation and describe the GeneClass algorithm for learning gene regulation programs from expression data and regulatory motif data. However, the GeneClass algorithm uses a fixed set of candidate motifs as an input to the algorithm and cannot discover unknown motifs. In many organisms, the binding sites of most transcription factors are not known. Even in well-studied organisms such as yeast, compendia of known DNA binding sites are incomplete and most binding sites are poorly characterized as deterministic sequences or consensus sequences due to limited number of experimentally confirmed target sites. As described in Section 2.4 transcription factors bind stochastically to different regulatory sequences with

different affinities. Hence, it is important to develop motif-discovery methods that catalog all the variants of a binding site in order to identify targets of the corresponding transcription factor.

In this chapter, we present *MEDUSA* (*Motif Element Detection Using Sequence Agglomeration*), an integrative method for learning motif models of transcription factor binding sites by incorporating promoter sequence and gene expression data. As in GeneClass, we use boosting with alternating decision trees (see Section 3.3), to enable feature selection from the high-dimensional search space of candidate binding sequences while avoiding overfitting. MEDUSA searches through the massive space of all possible subsequences in the promoter sequences of genes and builds a motif model whose presence in the promoter region of a gene, coupled with activity of a regulator in an experiment, is predictive of differential expression. Each motif model is either a k -length sequence, a dimer, or a position-specific scoring matrix (PSSM) (see Section 2.4) that is built by agglomerative probabilistic clustering of sequences with similar boosting loss. In this way, we learn motifs that are functional and predictive of regulatory response rather than motifs that are simply overrepresented in promoter sequences. Also, unlike GeneClass we do not use a set of candidate motifs or known transcription factor binding sites. We learn sequence motifs directly from raw promoter sequence data. Moreover, MEDUSA produces a model of the transcriptional control logic that can predict the expression of any gene in the organism, given the sequence of the promoter region of the target gene and the expression state of a set of known or putative transcription factors and signaling molecules.

We apply MEDUSA to various datasets of different sizes in yeast, worm and human B-cells. We learn yeast motifs whose ability to predict differential expression of target genes outperforms motifs from a compendium of known binding sites and from a previously published candidate set of learned motifs. We also show that MEDUSA retrieves many experimentally confirmed transcription factor binding sites. We also introduce a novel margin-based score to extract context-specific regulators and motifs.

5.2 Related methods

While there is a vast literature on the subject of motif discovery, only a few different conceptual approaches have been tried, and each of these standard approaches has its limitations. We briefly described some of these approaches in Section 3.1. Here we recapitulate the main differences between MEDUSA and other common motif discovery methods.

As discussed in Section 3.1, most methods for discovering transcription factor binding sites rely on first clustering genes (based on expression profiles, annotations, or both) and then looking for overrepresented patterns in the regulatory sequence for these genes. These methods tend to learn motifs specific to static clusters of genes and it is unclear how one objectively generalizes these motifs to the entire genome to identify other targets of the motif. These methods also fail to account for the transcription factor concentration and its effect on target expression.

The REDUCE method of Bussemaker *et al.* [20] discovers motifs whose presence individually correlates with differential mRNA expression in a single microarray experiment. Also, motifs are learned as deterministic subsequences and variation in the motifs can exist by selecting close variants of the same subsequence.

The motif-discovery algorithms by Beer and Tavazoie [10] and Segal *et al.* [106] force genes into static modules and do not model dynamic motif activity. The learning algorithms are highly sensitive to initialization procedure and can suffer from local minima problems. For statistical validation, these methods do not specifically predict the gene expression data directly. Rather, they try to predict module assignment of genes.

In MEDUSA, we adopt an alternative approach. Rather than trying to learn structure in promoter sequence data relevant to specific modules of genes, we learn a single context-specific regulation program for all genes. The model predicts the differential expression of target genes as a function of biologically meaningful regulatory inputs, including the expression levels of regulatory proteins and promoter sequence data. We view the learning

task as a prediction problem rather than a model selection problem, and we seek to learn regulatory programs that will make accurate predictions of differential expression in new or held-out experiments. MEDUSA learns probabilistic representations of motifs and automatically learns all the genes targeted by the motifs. The MEDUSA algorithm does not require complex initialization and has very few parameters, making it easy to run “out-of-the-box”.

5.3 Learning cis regulatory motifs from regulatory sequence and expression data

The MEDUSA learning algorithm is an extension of the GeneClass algorithm presented in Chapter 4. We once again model the learning task as classification problem i.e. we use boosting with alternating decision trees to learn a single regulation program that predicts the up and down regulation of all genes in all experiments of a given expression dataset. The primary difference between GeneClass and MEDUSA is the construction of the regulatory sequence feature space and the manner in which the weak learner picks a weak rule at each boosting iteration. We begin with an overview of the learning process.

5.3.1 Overview of the MEDUSA learning algorithm

Figure 5.1 illustrates the major steps and data used in the MEDUSA learning algorithm. As in GeneClass, the gene expression data is discretized into three states, up (over-expressed), down (under-expressed), and baseline (not significantly differentially expressed), and genes are partitioned into potential regulators (transcription factors and signal transducers) and targets. The regulators are also included in the list of target genes so that their transcriptional regulation can be modeled. The MEDUSA learning algorithm is presented with the promoter sequences of target genes, the discretized expression profiles of the regulators across multiple conditions, and the differentially expressed (up and down) target gene examples from these

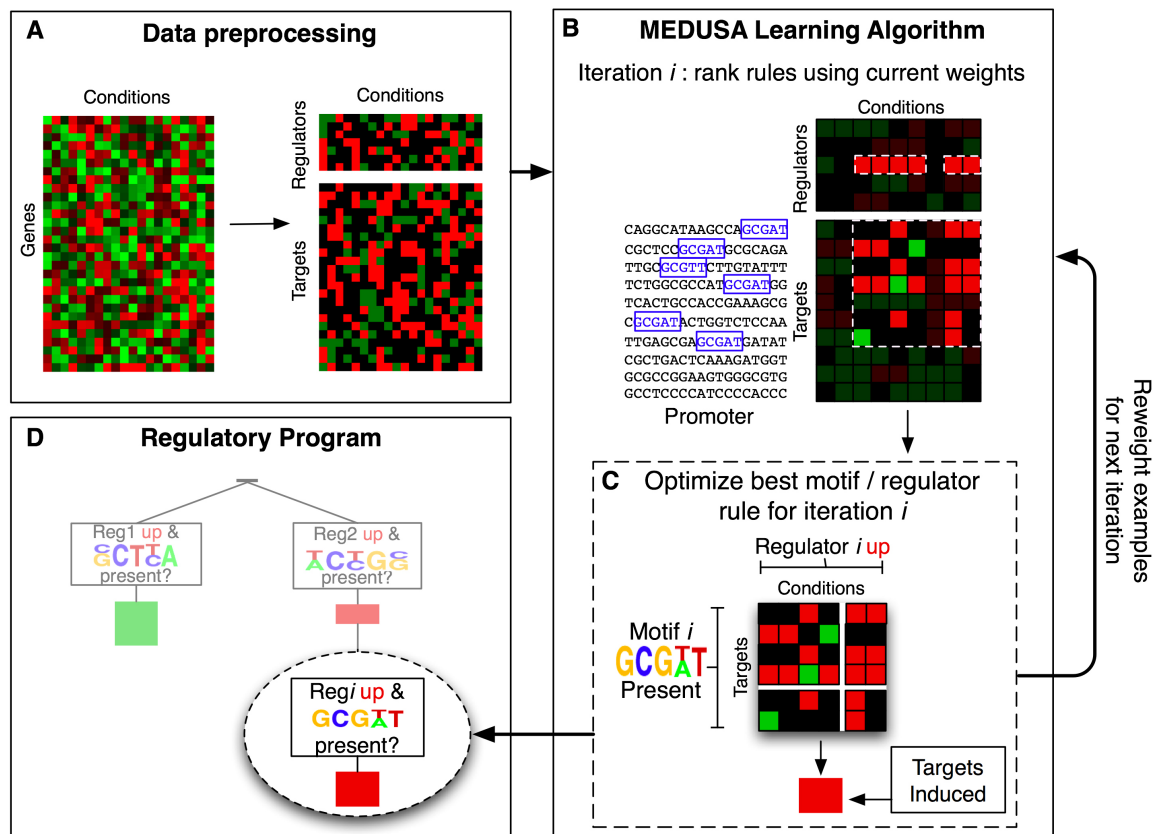


Figure 5.1: **Overview of the MEDUSA learning algorithm:** Part (A) shows the data-preprocessing steps. Part (B) shows the representation of the training data. Part (C) shows an example of a weak rule. Part (D) shows the regulation program as an alternating decision tree

experiments. Baseline examples are not used to train MEDUSA. MEDUSA considers rules based on promoter sequence data and regulator expression states. MEDUSA uses a boosting strategy to avoid overfitting over many rounds of the algorithm. At each iteration i , a motif-regulator rule is chosen based on the current weights on the training examples. This rule predicts that targets whose promoters contain the motif will go up (or down) in experiments where the regulator is over- (or under-) expressed. Before the next iteration, the examples are reweighted to emphasize the ones that are difficult to predict. To learn the sequence motif, the algorithm agglomerates predictive subsequences to produce candidate PSSMs, and it optimizes both the choice of PSSM and the probabilistic threshold used to determine where the hits of the motif occur. At the end of each round of training, motif-regulator rules are placed into an alternating decision tree, building a global regulation program. The regulation program asks questions such as, "Is the mRNA level of regulator i up (or down) in the experiment, and is the motif j present in the upstream region of the gene?" The control logic of the regulatory program is described by an alternating decision tree (See Figure 5.1(D) and Figure 4.2), which encodes how the overall up or down prediction score for a target gene in a given experimental condition results from the contribution and interaction of multiple regulators and motifs.

5.3.2 Feature space

The discretization of expression data (see Section 4.5.2) into up- and down-regulated expression levels allows us to formulate the problem of predicting regulatory response of target genes as the *binary classification* task of learning to predict up and down examples. Rather than viewing each microarray experiment as a training example, MEDUSA considers all genes and experiments simultaneously and treats every gene-experiment pair as a separate instance, dramatically increasing the number of training examples available. For every gene-experiment example, the gene's expression state in the experiment (up- or down-regulation) gives the output label $y_{ge} = \pm 1$. As in GeneClass, baseline examples labeled 0 are not used

in training since their labels are uncertain, in that the amount of up/down regulation is within the level of noise.

The inputs to the algorithm are (i) the promoter sequences of the target genes and (ii) the discretized expression levels of a set of candidate regulator genes. The sequence data is represented only via occurrence or non-occurrence of motifs represented by all possible length- k words ($k = 3 \dots 7$) known as k -mers and dimers with specific gaps and orientations. A full discussion of how MEDUSA determines the set of motifs at each round of boosting is given in Section 5.3.4. Let $M_{\mu g}$ indicate the presence ($M_{\mu g} = 1$) or absence ($M_{\mu g} = 0$) of a motif μ in the promoter sequence of gene g , and let $P_{\pi e}^s$ indicate the up-regulation ($s = +1$) or down-regulation ($s = -1$) of a regulator π in experiment e ($P_{\pi e}^s = 1$, if regulator π is in state s in experiment e , and $P_{\pi e}^s = 0$, otherwise). Hence, the feature vector for a gene g in an experiment e is given by $\{\{M_{\mu g}\}, \{P_{\pi e}^s\}\}$.

5.3.3 MEDUSA weak learner

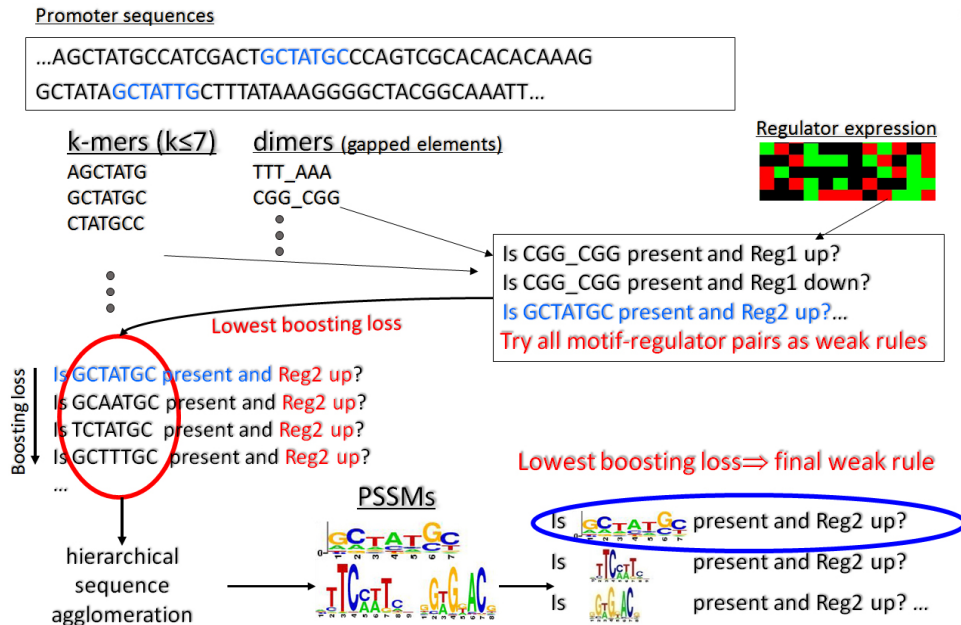


Figure 5.2: Overview of the MEDUSA weak learner

As shown in Figure 5.2, at each iteration of boosting, MEDUSA invokes a weak learner that picks a weak rule h^* that optimizes the exponential Adaboost loss function (See Section 3.3.1) given by

$$L(h) = W_0(h) + 2 \sqrt{W_+(h) \cdot W_-(h)} \quad (5.1)$$

Our weak rules split the gene-experiment examples in the training data by asking questions of the form “Is $M_{\mu g} P_{\pi e}^s = 1$?”; i.e., “Is motif μ present, and is regulator π in state s ?”. In this way, each rule introduced corresponds to a putative interaction between a regulator and some sequence element in the promoter of the target gene that it regulates.

However, since binding sites of transcription factors are rarely deterministic sequences we augment this weak learner with a hierarchical agglomerative procedure over motifs to learn PSSMs. The sequence agglomeration procedure is equivalent in principle to the stabilization procedure used in GeneClass (See Section 4.3.4). The main idea is to agglomerate motifs with similar boosting loss into a more predictive PSSM. This procedure serves two purposes. First, it stabilizes the learned model thus improving prediction accuracy. Second, it avoids masking of equally predictive k -mers or dimers allowing us to learn a richer motif representation. Thus a weak rule can contain a motif which can be a k -mer, a dimer or a probabilistic PSSM. Details of the agglomeration procedure are presented in Section 5.3.4.

The weak rules are combined by weighted majority vote using the structure of an alternating decision tree [31, 79] (See Section 3.3.2). If the {motif presence, regulator state} condition for a particular rule holds in the example considered, the coefficient of the rule is added to the final prediction score. This coefficient can be either positive or negative, contributing to up- or down-regulation respectively. Rules that appear lower in the tree are conditionally dependent on the rules in ancestor nodes. The tree structure is thus able to reveal combinatorial interactions between regulators and/or motifs. The sign of the final prediction score gives the prediction, and the absolute value of the score indicates the level of confidence.

Each iteration of the boosting algorithm results in the addition of a new splitter node (corresponding to a new weak rule) and its corresponding prediction node to the tree. The weak rule and its position in the tree at which it is added are chosen by minimizing the boosting loss over all possible combinations of motifs, regulators, and regulator-states, and over all possible positions (“preconditions”) in the current tree. A pseudo-code description is given in Appendix C.

The implementation uses efficient sparse matrix multiplication in MATLAB, exploiting the fact that our motif-regulator features are outer products of motif occurrence vectors and regulator expression vectors, and allows us to scale up to large datasets and the high-dimensional feature space.

5.3.4 Learning PSSMs using hierarchical sequence agglomeration

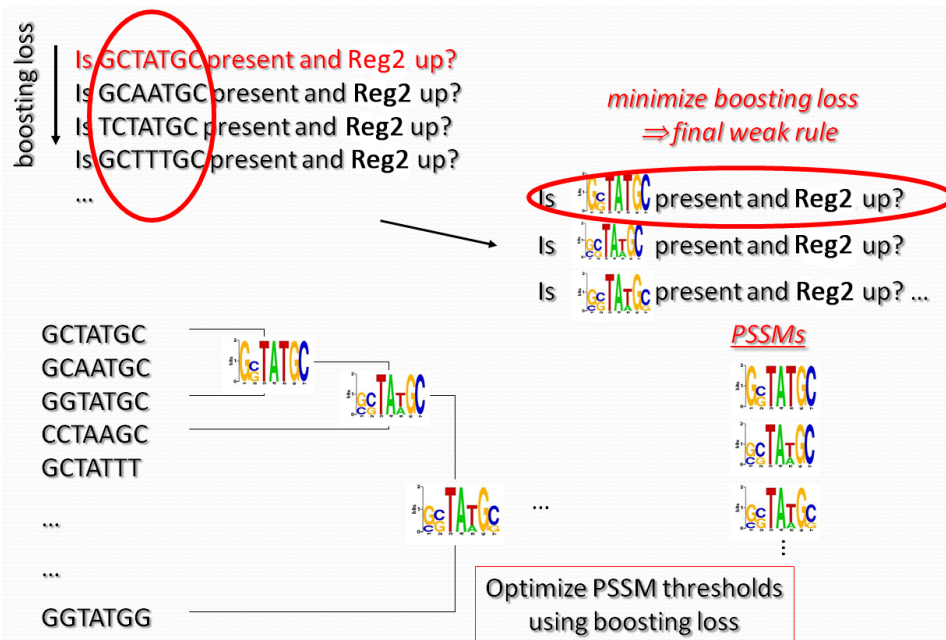


Figure 5.3: Overview of hierarchical sequence agglomeration in MEDUSA

At each boosting iteration, MEDUSA considers all occurrences of k -mers and gapped homodimers in the promoter sequence of each gene as candidate motifs. We restrict the set of

all dimers to those whose two components (monomers) have specific relationships, consistent with most known dimer motifs: equal (e.g. ACG_ACG), reversed (e.g. ACG_GCA), complements (e.g. ACG_TGC), or reverse complements (e.g. ACG_CGT). Since slightly different sequences might in fact be instances of binding sites for the same regulator, MEDUSA uses a hierarchical motif clustering algorithm to generate more general candidate PSSMs as binding site models (see Figure 5.3). The motif clustering uses k -mers and dimers associated with low boosting loss as a starting point to build PSSMs: these sequences are viewed seed PSSMs, and then the algorithm proceeds by iteratively merging similar PSSMs, as described below. The generated PSSMs are then considered as additional putative motifs for the learning algorithm.

As discussed in Section 2.4, for a binding site of length k , a position-specific scoring matrix (PSSM) is a $k \times 4$ matrix that assigns a probability $p_i(x)$, for each position $i = 1 \dots k$ and nucleotide $x \in \{A, C, G, T\}$. In order to search for hits of a binding site of length k in a longer sequence, we score all overlapping subsequences of length k using a *log-odds score*. For a subsequence $a_1, a_2 \dots a_k$ where $a_i \in \{A, C, G, T\}$, and a PSSM as defined above, the log-odds score is defined as $S = \sum_{i=1 \dots k} \log_2(p_i(x = a_i)/p^{bg}(x = a_i))$. The background probability of nucleotide x is given by $p^{bg}(x)$. This log-odds score is compared to some threshold θ to determine if the subsequence is a hit. For PSSMs representing gapped dimers, the part of the subsequence corresponding to the gap in the dimer is not used for scoring the subsequence.

When comparing two PSSMs, we allow possible offsets between the two starting positions. In order to give them the same lengths, we pad either the left or right ends with the background distribution. We then define a distance measure $d(p, q)$ as the minimum over all possible position offsets of the JS entropy [23] between two PSSMs p and q .

$$d(p, q) \equiv \min_{\text{offsets}} [b_p D_{KL}(p \| b_p p + b_q q) + b_q D_{KL}(q \| b_p p + b_q q)],$$

where D_{KL} is the Kullback-Leibler (KL) divergence [23] and is given by

$$D_{KL}(p||q) \equiv \sum_{a_1, \dots, a_n} p(a_1, \dots, a_n) \ln \frac{p(a_1, \dots, a_n)}{q(a_1, \dots, a_n)}$$

By using $p(a_1 \dots a_n) = \prod_{i=1}^n p_i(a_i)$ and $\sum_{a_i} p_i(a_i) = 1$ (and the analogous equations for q) one can easily show that $D_{KL}(p||q) = \sum_{i=1}^n D_{KL}(p_i||q_i)$. The relative weights of the two PSSMs, b_p and b_q , are here defined as $b_{p,q} = G_{p,q}/(G_p + G_q)$, where G_p, G_q are the number of genes whose promoter sequences have hits for PSSMs p and q respectively. Note that this distortion measure is not affected by adding more “padded” background elements either before or after the PSSM. Our merge criterion is similar to the one used in the agglomerative information bottleneck algorithm [110], though we also consider offsets in our merges.

At every boosting iteration, we first find the weak rule h^* among all possible combinations of regulators, regulator-states and k -mers/dimers, that minimizes boosting loss. The input to the sequence agglomeration procedure are a set of K motifs with lowest loss appearing with the same regulator, regulator state, and tree-position as h^* . Sequence motifs can be regarded as PSSMs with 0/1 emission probabilities, smoothed by background probabilities. By iteratively joining the PSSMs with smallest $d(p, q)$, the clustering proposes a set of $K - 1$ PSSMs from the various stages of the hierarchy. The threshold θ_p for a PSSM p determines the set of genes G_p targeted by the PSSM. The boosting loss $L(h)$ associated with a regulator-PSSM pair depends on the weights w_i of training examples (g, e) for all $g \in G_p$ (See Equation 5.1). Hence, the threshold θ_p determines the boosting loss. At every merge of two PSSMs, the optimal score threshold θ_p associated with the new PSSM is found by minimizing the boosting loss over possible values for the threshold.

Note also that the new PSSM can be longer than either of the two PSSMs used in the merge, due to the procedure of merging with offsets; in this way, we can obtain candidate PSSMs longer than the maximum seed k -mer length of 7. The number of target genes, which determines the weight of the PSSM for further clustering, is calculated by counting

the number of promoter sequences which score above the threshold. The total number of possible weak rules is equal to $2R(N_{k\text{-mer}} + N_{\text{dimer}}) + K - 1$ where R is the number of candidate regulators and $N_{k\text{-mer}}$ and N_{dimer} are the numbers of k -mers and dimers, respectively. This is of the order of 10^7 . Our results show that even in this high-dimensional space boosting does not lead to overfitting. The new node that is added to the alternating decision tree is the weak rule that minimizes boosting loss considering all sequence motifs and PSSMs.

5.4 Extracting predictive features

In order to extract context-specific regulators and motifs that are predictive of a subset of genes in a set of experiments, we introduced the frequency score in Section 4.4.2. The frequency score for a motif/regulator over a set, $B = \{(g, e)\}$, of examples is defined as the number of correctly predicted examples in B that pass through all splitter nodes containing the motif/regulator.

However, this score does not take into account the relative contribution of the motif/regulator to the prediction scores of these examples. We thus, introduce a *margin-score* for ranking the regulators and motifs affecting target genes based on the theoretical idea of a margin [61]. In large-margin techniques like boosting and SVMs, the margin for an example x_{ge} with label $y_{ge} = \pm 1$ and prediction function F is given by $y_{ge}F(x_{ge})$. If the margin is positive, the prediction is correct, and the size of the margin gives a measure of confidence in the prediction. If we remove, for example, regulator R from the regulatory program (i.e. delete nodes containing R and their subtrees from the ADT learned by boosting), we denote F^{-R} as the modified prediction function and define the following score:

$$S_R = \sum_{\{(x_{ge}, y_{ge})\} \in T} y_{ge} (F(x_{ge}) - F^{-R}(x_{ge})) \quad (5.2)$$

where T is a set of training examples of interest, e.g. a particular set of genes restricted to some subset of experiment. The score S_R will again be positive if on average R is important

for making predictions, and its size measures its importance to the target set.

5.5 Datasets

5.5.1 Yeast (*S. cerevisiae*) datasets

We use MEDUSA to analyze three gene expression datasets of different sizes in the yeast *S. cerevisiae*.

5.5.1.1 Gene expression data

Section 4.5.1 introduced two yeast gene expression datasets sets namely *the environmental stress response dataset* [37] (*ESR*) and the DNA damage dataset [36]. The ESR dataset assays the genome-wide gene expression response in 173 cDNA microarray experiments spanning 13 different environmental stresses. We discretize the gene expression data as explained in Section 4.5.2.

The DNA damage dataset is a moderately sized dataset consisting of 53 microarrays assaying the mRNA levels of all yeast genes in response to different DNA damaging agents. We discretize the gene expression data as explained in Section 4.5.2.

We also analyze a new unpublished yeast dataset from our collaborator Dr. Li Zhang. We refer to this dataset as the *hypoxia dataset*. We use RNA samples from 8 different experimental conditions. These include aerobic response, aerobic response of a $\Delta hap1$ knockout, early hypoxia response, late hypoxia response, late hypoxia response of a $\Delta hap1$ knockout, response to cobalt chloride (Co^{2+}), response to heme sufficiency and heme deficiency. We use 24 single channel Affymetrix oligonucleotide microarrays to assay the gene expression. Details of the yeast samples used, RNA preparation and microarray setups are provided in Section 6.2. We use this dataset to study the role of oxygen, heme, Hap1, and Co^{2+} in oxygen sensing and regulation [61]. Since Affymetrix microarrays are single channel arrays, we compare each of the knockout, stress or perturbation microarray

experiments to a corresponding reference microarray. The expression fold-changes are converted to p -values using an intensity-specific noise model obtained from replicate data (See Section 4.5.2). The fold-changes are then discretized into +1, 0 or -1 labels using a p -value threshold of 0.05. A label of ± 1 indicates up/down-regulation beyond the threshold level of noise. In Affymetrix arrays several genes have multiple probes on the gene chip. In such cases, we discretize each probe reading independently and used a majority vote over the +1 and -1 discretized values to obtain a final label for the gene. In cases where a majority vote is not possible (due to equal number of probes with +1 and -1 values), we use the label corresponding to the reading with the lowest p -value. All the replicate experiments are used as input to MEDUSA. However, in order to remove inconsistency, for each gene, we further filtered out expression values that did not agree with the consensus label (+1 or -1) across replicates of a particular experimental condition.

5.5.1.2 Candidate set of regulators

Our candidate set of yeast regulatory proteins consists of 475 genes consisting of 237 known and putative transcription factors and 250 known and putative signaling molecules, with an overlap of 12 genes of unknown function. Of these, 466 are from Segal *et al.* [104] and 9 generic (global) regulators are obtained from Lee *et al.* [68].

5.5.1.3 Promoter sequences

We use 1000 bp nucleotide sequences upstream of the transcription start site (TSS) of all *S. cerevisiae* genes that we obtain from the Saccharomyces Genome Database (SGD, <ftp://ftp.yeastgenome.org/yeast/>, Jan 2006). We scan these sequences for all occurring k -mer motifs ($k = 3 \dots 7$) as well as 3 – 3 and 4 – 4 dimer motifs allowing a middle gap of up to 8 bp. We restrict the set of all dimers to those whose two components have specific relationships, consistent with most known dimer motifs: equal, reversed, complements, or reverse complements.

5.5.1.4 Gene annotations

We obtain gene annotations and functional associations from *Saccharomyces* Genome Database (SGD, <ftp://ftp.yeastgenome.org/yeast/>, Jan 2006). The gene ontology (GO) tree structure was downloaded from the Gene Ontology Consortium [1]. To identify statistically enriched terms associated with sets of genes, we calculate p -values using the cumulative hypergeometric null distribution on the basis of the number of genes in the set, the number of genes in that set that are annotated with each GO term, and the number of genes in the genome that are annotated with that GO term. We then filter terms using a threshold.

5.5.2 Worm (*C. elegans*) dataset

We also use MEDUSA to analyze early embryonic development of the worm *Caenorhabditis elegans*. We use the expression dataset by Baugh *et al.* [8]. It consists of a finely sampled time course that commences with the zygote and extends into mid-gastrulation, spanning the transition from maternal to embryonic control of development and including the presumptive specification of most major cell fates. The dataset consists of 7 time points with multiple replicates for each experiment. The gene expression data is assayed using single channel affymetrix microarrays. We transform the data into fold changes using the *PC32* time point (32 minutes after pseudo-cleavage) as control. We use replicate data to estimate an intensity-dependent noise model. We discretize the data into 3 levels — ± 1 representing significant up(down)regulation and 0 representing expression measurements within the level of noise using a p -value of 0.01. This gives us a total of 9135 genes that significantly change expression in at least one time point.

Our candidate regulator set consists of 1370 genes consisting of transcription factors, kinases, phosphates and signaling molecules from WormBook, <http://www.wormbook.org>, TRANSFAC [77] and WormPD [22]. We obtain promoter sequences spanning 1000 bp upstream of the genes from Wormmart (<http://www.wormbase.org/biomart/martview>).

5.5.3 Human B-cell dataset

The human B-cell gene expression dataset [6] consists of 336 samples including: normal purified cord blood (5 samples), germinal center (10), memory (5) and naive (5) B cells; 34 samples of B cell chronic lymphocytic leukemia (B-CLL), 68 of diffuse large B cell lymphomas (DLBCL) including cases further classified as immunoblastic or centroblastic, 27 of Burkitt lymphoma (BL), 6 of follicular lymphoma (FL), 9 of primary effusion lymphoma (PEL), 8 of mantle cell lymphoma (MCL), 16 of hairy cell leukemia (HCL), 4 cell lines derived from Hodgkin disease (HD), 5 B-cell lymphoma cell lines and 5 lymphoblastic cell lines. We restricted our analysis of the B cell data set to the well-studied Burkitt lymphoma cell line (Ramos) treated in vitro to activate CD40 or B-cell receptor (antiIgM) signaling and cell lines engineered to stably express BCL6 and BCL6(Δ PEST) mutant [6]. We use several sets of control conditions to obtain a data set of 102 experiments, grouped into 12 sets, each of which probe the genome-wide response to specific treatments.

We use the discretization procedure explained in Section 4.5.2. This gives us a set of 8500 genes that show significant up/down regulation in atleast one experiment.

We use the BioMart database (<http://www.biomart.org/>) to compile a candidate set of 3800 regulators. These regulators are obtained by searching for genes whose functional annotations contain keywords such as “DNA-binding”, “transcription factor”, “signal transduction”, “kinase”, “phosphatase”, “receptor”, “cofactor” and “regulation of transcription”. We also use BioMart to obtain promoter sequences spanning 2000 bp upstream of the 8500 target genes.

5.6 Statistical validation

5.6.1 Prediction accuracy in cross-validation experiments

We report prediction accuracy in cross-validation experiments for MEDUSA on the five gene expression data sets described in Section 5.5. These datasets are of different sizes and involve very different biological processes in the yeast *S. cerevisiae*, the worm *C. elegans*, and human B-cells: a large data set measuring yeast stress response to diverse environmental stress (ESR) [37], a moderate sized dataset measuring yeast DNA damage response, a small unpublished data set from our collaborator Dr. Li Zhang studying the role of oxygen, heme, Hap1, and Co^{2+} in oxygen sensing and regulation (Hypoxia), a developmental time course from the early worm embryo [9] (Worm), and part of a human B cell expression atlas [6] (B Cell).

Data set	#expts / #genes	cross-validation set-up	algorithm	sequence features	error rate
ESR	173 / 475 regs ~6000 targets	5-fold c.v., held-out expts, replicate expts grouped	MEDUSA	promoters	13.4%
				AlignAce motifs	16.1%
			<i>k</i> -nearest neighbor	TRANSFAC motifs	20.8%
				TRANSFAC motifs	31.3%
DNA damage	52/ 475 regs ~6000 targets	10-fold c.v., held-out examples	MEDUSA	promoters	20.7%
Hypoxia	18 (6 conditions) / 475 regs ~3500 targets	10-fold c.v., held-out examples	MEDUSA	promoters + ChIP	8.0%
		10-fold c.v., held-out examples replicate gene-expts grouped	MEDUSA	ChIP only	26.0%
				promoters + ChIP	23.9%
Worm	22 (6 time points) / 1390 regs ~8500 targets	10-fold c.v., held-out examples replicate gene-expts grouped	MEDUSA	promoters	17.0%
B Cell	102 (23 conditions) / 3291 regs ~7500 targets	10-fold c.v., held-out examples replicate gene-expts grouped	MEDUSA	promoters	25.7%

Table 5.1: Prediction performance for MEDUSA across multiple yeast, worm and human data sets:

For the ESR data set, we compare to a baseline method, *k*-nearest neighbor, to assess the difficulty of the prediction problem.

Results showing that MEDUSA achieves low prediction error rates across all the data sets and in different cross-validation experiments are summarized in Table 5.1. In each case, we report the error rate of MEDUSA’s up/down predictions on the differentially expressed test examples, i.e. baseline examples are not used for training nor are they included in this

evaluation.

5.6.1.1 Prediction accuracy for yeast datasets

Our largest scale experiments are performed on the ESR data set. Here we divide experiments into 5 folds and group replicate experiments together within folds for 5-fold cross-validation. This procedure ensures that replicates of an experimental condition are never in both training and test sets, making the prediction task more difficult. In this setting, learning motifs directly from promoter sequences, MEDUSA achieves an impressively low error rate of 13.4%.

For the smaller DNA damage data set, we performed 10-fold cross-validation experiments on held-out gene-experiment examples instead of held-out experiments, and we ran MEDUSA for 300 iterations of boosting. DNA damage is both smaller and more diverse than ESR, with many of the experiments involving gene knockout strains and mutants, so that the reference conditions vary across the data set; our noise modeling also revealed that the data were noisier than in ESR. For these reasons, the error rate of 20.7%, while not as good as ESR, still represents significant generalization performance.

Since the hypoxia data set is too small to allow evaluation of test-loss on held-out experiments, we perform 10-fold cross-validation experiments on held-out gene-experiment examples in this case. Here, we achieve a very low error rate of 8.0%, but the prediction task is made easier by the presence of replicates in the data set: each of 6 experimental conditions are represented by 3 replicates, though one replicate is from a different yeast strain. We repeat the 10-fold cross-validation but group replicate measurements (replicate expression values for the same gene and in the same condition) within folds, so that we never see the same gene in the same condition in both training and testing. In this more difficult setting, MEDUSA still achieves significant prediction performance, with an error rate of 23.9%.

5.6.1.2 Prediction accuracy for worm and human datasets

To test MEDUSA's statistical performance in higher eukaryotes, we perform the same 10-fold cross-validation set-up with held-out examples and replicate examples grouped in folds on the Worm and B-cell data sets.

On the Worm data set, an early embryonic expression time course, we obtain a test error of 17.0% which is comparable to our yeast results and is in fact much better than results on the hypoxia data set, which is about the same size but more diverse and noisier.

In the B-cell data set, despite the complexity of mammalian gene regulation and the larger number of regulators and targets, the statistical performance is still good, with test error of 25.7% – comparable to a that of the DNA damage stress response in yeast.

5.6.2 Comparison to GeneClass

We also compare the prediction accuracy of MEDUSA to that obtained using GeneClass with a fixed set of database motifs as sequence features. As seen in Table 5.1 for the ESR dataset using either TRANSFAC [128] or AlignAce [92] motifs results in a significantly higher prediction error rate of 20.8% and 16% respectively as compared to MEDUSA's error rate of 13.4%, suggesting that MEDUSA can extract sequence information that is not represented in these databases.

Also, for the hypoxia dataset using GeneClass with transcription factor occupancy data from ChIP-chip experiments [42] leads to a much higher error rate of 26.0% as compared to MEDUSA's 8% error rate, suggesting that the problem is not trivial (and that the conditions under which the ChIP chip experiments were performed are not relevant to hypoxia).

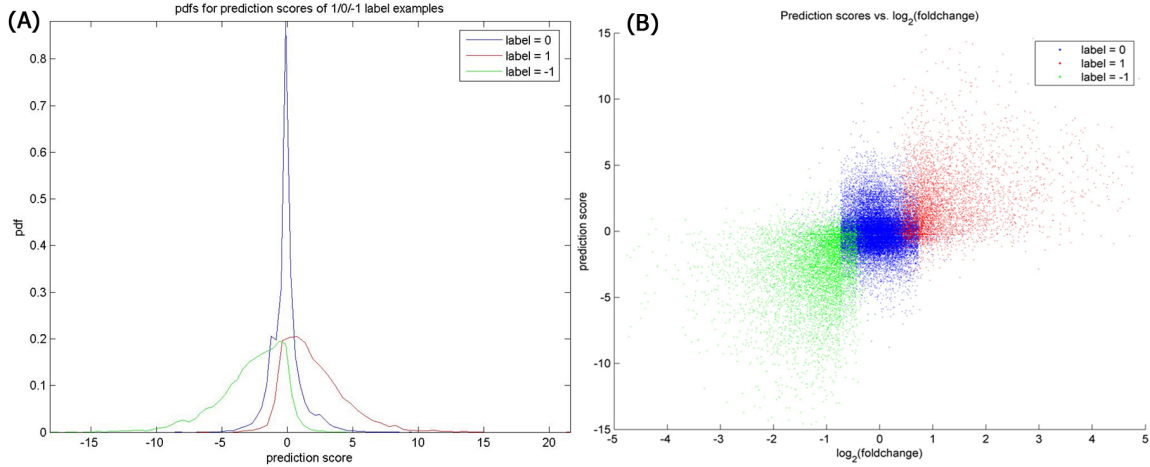


Figure 5.4: **Three-class prediction performance on the hypoxia dataset:** (A) shows the distribution of the prediction scores for the three classes. (B) shows a scatterplot of the true log₂ expression values versus prediction scores for all examples in 10-fold cross validation.

5.6.3 Prediction accuracy for the three-class (up/down/baseline) prediction problem

MEDUSA uses only the up/down-regulated examples for training and testing. This generally constitutes a small fraction of all the expression data (10-15%). However, it is possible to make three-class predictions (up, down, or baseline) by thresholding MEDUSA's prediction scores, and in this way we can report 3-class cross-validation accuracy across all examples, including those that are labeled baseline.

The output of the MEDUSA learning algorithm is a real-valued prediction function $F(x_{ge})$ for all genes and experiments in the form of an alternating decision tree. The sign of the prediction score gives the predicted label and the absolute value represents a confidence level for the prediction. We can make 3-class predictions by thresholding on the confidence levels of up and down predictions, that is, we predict examples to be up- or down-regulated if $F(x_{ge}) > a$ or $F(x_{ge}) < -b$, and to be baseline if $-b \leq F(x_{ge}) \leq a$ where $a, b > 0$.

We reexamine our 10-fold cross-validation results on the hypoxia dataset using held-out examples to evaluate three-class prediction performance, where baseline examples were randomly divided among the 10 folds for the purpose of reporting results. Figure 5.4 (A)

			Predicted labels	
True label		+1	0	-1
	+1	71.6%	26.5%	2.0%
	0	18.3%	69.9%	11.8%
	-1	3.3%	27.8%	68.9%

Figure 5.5: **Confusion matrix for three-class classification on the hypoxia dataset:** Truth and predictions for all genes in the held-out experiments, including those expressed at baseline levels. Examples are binned by assigning thresholds $a = 0.49$ and $-b = -1.21$ for prediction of positive and negative labels, respectively.

shows the distribution of the prediction scores for the three classes, and Figure 5.4 (B) shows a scatterplot of the true \log_2 expression values versus prediction scores for all examples. In both figures, we see a good separation between classes, and in the scatterplot, we see a significant correlation between true expression level and real-valued prediction score. We choose the confidence thresholds a and b so as to optimize the mean balanced accuracy (average accuracy over the three classes) over the 10 folds. We find the minimum balanced accuracy to be 70.1% for $a = 0.49$ and $-b = -1.21$. Performance of a random classifier would yield a balanced accuracy of 33.3%. While it would be more correct to choose the thresholds based on a separate cross-validation scheme, we observe that the balanced error is fairly stable across folds, suggesting that our choice does not lead to overrated performance. For these thresholds, we obtain the confusion matrix shown in Table 5.5, demonstrating strong diagonal entries and reasonable accuracy on the baseline examples, despite the fact that these examples are omitted from training.

5.6.4 Comparison to simple methods based on clustering or correlation

We have already shown that GeneClass strongly outperforms a baseline approach, based on k -nearest neighbor, when both methods are tested on the same prediction problem: predict up/down target expression from regulator expression states and promoter motif content,

using database motifs (Section 4.6.4). We now consider a different question, which is whether MEDUSA outperforms “strawman” methods based on clustering or correlation which make no attempt to use sequence information. We consider two approaches: a *clustering strawman*, where genes are clustered across the training experiments, each cluster is assigned a “nearest regulator”, and this regulator’s expression state or fold-change is used to predict the up/down state for all the cluster’s genes; a *nearest regulator strawman*, which similarly assigns a nearest regulator to every target gene whose state or fold-change is used to predict the target’s state. To emphasize that the strawmen solve different prediction problems than MEDUSA, with different implied models of regulation, we contrast the methods in Table 5.2.

prediction method	gene-specific information used	implied regulation model
clustering strawman	gene’s cluster membership	all genes in cluster are controlled by one regulator (or possibly k highly correlated regulators) under all experimental conditions
nearest regulator	gene’s identity	each gene is controlled by one regulator (or possibly k highly correlated regulators) under all experimental conditions
MEDUSA	gene’s promoter sequence	genes are controlled by a set of regulators; different regulators act in different experimental conditions and are not necessarily well-correlated with each other

Table 5.2: Strawman methods for comparison to MEDUSA: The strawmen methods and MEDUSA use different information in order to make predictions and make different assumptions about gene regulation.

We test the clustering strawman on the ESR data set. We use exactly the same 5-fold cross-validation set-up with held-out experiments as in our MEDUSA results. Prediction results are reported on differentially expressed (significantly up or down) test examples only, as before. We use k -means clustering on training experiments to assign genes to clusters (using hierarchical clustering led to similar results), and we use 50 clusters, which is typical for this dataset (e.g. [104]), though other choices also leads to similar results. We also try three choices of cluster representative used to make predictions for the cluster’s genes in the test experiments: a randomly chosen gene from each cluster, the gene in the cluster whose expression profile correlates best with the mean cluster profile, and the regulator that correlates best with the cluster profile.

Results of the cluster-representative approach are reported in Table 5.3. We try using both discrete expression or real-valued expression data in order to perform the clustering on the training set examples. However, what matters most critically is whether the cluster strawman is required to make predictions based on the discretized expression state of the regulator/representative, as MEDUSA does, or if it can use the sign of the log fold change. When restricted to using discrete (up or down) states to make predictions, the cluster strawman gives high error in cross-validation for all variants of the method, since fairly often the chosen cluster representative is in a baseline state in the test experiments. In this case, results are somewhat better if the clustering is performed on discrete rather than real-valued expression data, so we report these results only in the table. We note that in the ESR data set, the ratio of up to down differentially expressed examples is about 60 to 40, so that the best guessing strategy of always guessing “up” leads to 40% error. The cluster strawman with discrete regulator states performs worse than this guessing strategy.

If the cluster strawman is allowed to use the fold change of the cluster representative in order to make up/down predictions in the test experiments, prediction performance is much better, with the nearest regulator representative leading to 16.0% test error, slightly higher than but comparable to MEDUSA’s 13.4% accuracy. Note that in this situation, the cluster representative is making up/down predictions for significantly differentially expressed targets even though its own fold change may not be significant.

Is the clustering strawman in fact identifying important regulators, based on its good prediction performance when used with real-valued cluster representative levels? We first note that all choices of cluster representative – nearest regulator, nearest gene, random gene – give the same prediction performance. This result suggests that the cluster strawman is learning mainly about the correlation structure of the data, and it will only be successful in identifying regulators with a strong correlation signal. To examine this issue, we rank regulators by the number of clusters to which they are assigned as “nearest regulators” across all folds in the cross-validation study. We observe that the clustering strawman is able to

identify some of the known stress response regulators such as Tpk1, Usv1 and Xbp1, which are also picked by MEDUSA and a previous study based on module networks [104]. These regulators have a uniform, strong signal across all the experiments and are hence easily identified as predictive regulators by both techniques. However, more subtle regulatory signals such as the regulatory activity of Msn2/4 and Hsf1, which MEDUSA identifies, are not identified by the strawman. Also, since the strawman solely relies on expression data, it misses out on regulators whose activity is controlled post-translationally. An interesting example is the universal stress response regulator Msn2/4 whose regulatory activity is controlled by translocation into and out of the nucleus. Its mRNA expression does not change significantly in most of the ESR experiments, and hence it is difficult to identify as a significant regulator based on its expression profile. MEDUSA, however, identifies the Msn2/4 binding site as the top scoring motif. In this way, MEDUSA is able to identify a richer set of biologically relevant features as compared to the strawman techniques that use expression data alone.

prediction method		5-fold c.v. error rate held-out experiments replicates grouped
MEDUSA (discrete regulator states)		13.4%
Cluster strawman, using representative's discrete state for prediction	Representative target gene chosen randomly from cluster	63.6%
	Representative target gene closest to cluster mean	41.3%
	Representative regulator closest to cluster mean	51.4%
Cluster strawman, using representative's fold change for prediction	Representative target gene chosen randomly from cluster	19.0%
	Representative target gene closest to cluster mean	15.0%
	Representative regulator closest to cluster mean	16.0%

Table 5.3: Comparison of MEDUSA prediction performance to simple clustering-based methods on the ESR data set:

We report results for a clustering-based approach, where training set examples are clustered and a representative gene from each cluster, or a representative regulator for each cluster, is chosen. This gene's expression level on each test experiment is used to predict the up/down expression state for all other cluster members.

Due to the small number of experiments in the hypoxia data set, we use a cross-validation set-up of held-out examples, making it difficult to perform the clustering method. Instead, we compare the prediction performance of MEDUSA against a simple correlation-based approach, where we identify the regulator that best correlates with each target gene across

prediction method	10-fold c.v. error rate held-out examples	10-fold c.v. error rate held-out examples, replicates grouped
MEDUSA	8.0%	23.9%
Discretize nearest regulator	40.2%	69.0%
Majority vote of discretize $k = 10$ nearest regulators	11.1%	42.5%
Real-valued nearest regulator	56.7%	63.0%
Majority vote of real-valued $k = 10$ nearest regulators	13.6%	23.0%

Table 5.4: Comparison of MEDUSA prediction performance to simple correlation-based methods on the hypoxia data set: We report results for both the Pearson correlation over real-valued expression data (including baseline examples) and the normalized Hamming distance (excluding baseline examples) for discretized expression data, where the inclusion/exclusion of baseline examples was chosen in order to report the better results.

the training examples and use this regulator to predict the target’s expression level on the test examples. We also consider taking a majority vote of k -nearest regulators. We note that this approach necessarily uses the target gene’s identity rather than learning a single model that can be applied to all genes. In this data set, the ratio of up to down target gene examples is about 40 to 60; therefore, the best guessing strategy of always guessing the larger class has an accuracy of 60% and an error rate of 40%.

Results of our experiments are summarized in Table 5.4. Here we again choose nearest regulators based on both discrete and real expression profiles, but prediction is based on the discrete state of the regulator or vote of discrete states for more direct comparison to MEDUSA. We find that using a single “nearest regulator” for each target gives poor prediction results (similar or worse than the best guessing strategy) for both choices of the correlation metric. Taking the majority vote over a set of $k = 10$ discrete nearest regulators gives good test performance on the easier cross-validation set-up, but when replicate examples are grouped in folds, test error is again high (similar to the best guessing strategy). Finally, when we take a weighted vote of $k = 10$ real-valued nearest neighbors, test error is comparable to MEDUSA.

We again ask the question whether the real-valued k -nearest regulator method, whose prediction performance is similar to MEDUSA’s, in fact extracts meaningful regulators. In

the case of this smaller hypoxia dataset, we find that the biological relevance of the identified “nearest regulators” is lacking altogether. When we rank regulators based on the number of targets for which they are designated “nearest regulator”, we find that every regulator is picked at least once. Hap1, Rox1 and Mga2 are the key regulators mediating hypoxia and related responses. These regulators rank very low in the list (Hap1 ranks 452, Rox1 ranks 433 and Mga2 ranks). By contrast, in our main study (Chapter 6), MEDUSA identifies 54 statistically significant predictive regulators, many of which are known to be key regulators of the experimental conditions considered. Hap1, Rox1 and Mga2 rank among the top 15 regulators. We can conclude that, especially in the case of small datasets, single correlations between targets and regulators cannot be used to accurately predict expression of held-out examples and that the majority vote process leads to improved accuracy but does not identify biologically relevant regulators.

5.7 Biological validation

In this section, we show that MEDUSA is able to learn binding sites of several known transcription factors in the yeast, worm and human genomes. We also use the margin score (Section 5.4) to reveal context-specific regulators and motifs that regulate related groups of genes in different experiments. We present biological validation results on the yeast hypoxia dataset in Chapter 6.

5.7.1 MEDUSA discovers most known transcription factor binding sites in yeast

5.7.1.1 ESR dataset

For the ESR dataset, we compare motifs learned by MEDUSA to several known and putative binding sites, consensus sequences and PSSMs from five databases: TRANSFAC [128],




























TFNAME	DB-MOTIF	MOTIF	DBNAME	d(p,q)	TFNAME	DB-MOTIF	MOTIF	DBNAME	d(p,q)
CBF1	CACGTG		YPD	0.032635	RAP1	RMACCCA		SCPD	0.523059
CGG everted repeat	CGGN*CCG		YPD	0.032821	mPAC			AlignACE	0.552493
MSN2			TRANSFAC	0.085626	mRRPE			AlignACE	0.630740
HSF1	TTCNNNGAA		SCPD	0.102410	PHO4			TRANSFAC	0.672961
XBP1			TRANSFAC	0.140561	YAP1			TRANSFAC	0.777816
STE12			TRANSFAC	0.256750	MIG1	CCCCACAAA		YPD	0.799412
GCN4			SCPD	0.292221	MET31,32	AAACTGTGG		YPD	0.84893
TBP			TRANSFAC	0.376801	HAP2,3,4			TRANSFAC	1.070837
HAP1	CGGNNTWNCGG		YPD	0.423004					

Figure 5.6: MEDUSA motifs learned from the ESR dataset: By using the symmetrized KL divergence as a distance measure, we match PSSMs identified by MEDUSA's to motifs known in the literature. The table shows the logos of MEDUSA's PSSMs (column 3), the matching motif of the database (column 2), the corresponding transcription factor (column 1), the name of the database (column 4) and the distance (column 5).

TFD, SCPD, YPD and a set of PSSMs found by AlignACE [92]. After converting the sequences and consensus patterns to PSSMs, smoothed by background probabilities, we compare all PSSMs with the ones found by MEDUSA using the symmetrized Kullback-Leibler divergence which is the same distance metric we use for PSSM clustering in MEDUSA (Section 5.3.4). We define the best match for each of MEDUSA's PSSMs as the database PSSM that is closest to it in terms of this distance metric. Each splitter node in the alternating decision tree predicts on a particular subset of gene-experiment examples. Hence, the motifs in the splitter nodes define some subset of target genes. We can associate motifs with Gene Ontology (GO) annotations by looking for enriched GO annotations (Section 5.5.1.4) in these gene subsets, and we can estimate the putative functions of the targets of a transcription factor that might bind to the PSSM in each node.

We see matches to variants of the STRE element, the binding site for the Msn2/4 general stress response transcription factors. The genes passing through nodes containing these PSSMs are significantly enriched for the GO terms carbohydrate metabolism, response

to stress and energy pathways, consistent with the known functions of Msn2/4. Gcr1 and Rap1 are known to transcriptionally regulate ribosomal genes, consistent with enriched GO annotations associated with the nodes of the specific PSSMs. The heat shock factor Hsf1—which binds to the heat shock element (HSE)—plays a primary role in stress response to heat as well as several other stresses. The heat shock element exists as a palindromic sequence of the form nGAAnnTTCn. We find almost an exact HSE in the tree. In *S.cerevisiae*, several important responses to oxidative and redox stresses are regulated by Yap1, which binds to the YRE element. We find several strongly matching variants of the YRE. Comparison of PSSMs from AlignACE with our PSSMs reveals the PAC and RRPE motifs to be among the top three matches. These PSSMs also appear in the top 10 iterations in the tree, indicating they are also strongly predictive of the target gene expression. Both these putative regulatory motifs have been studied in great depth with respect to their roles in rRNA processing and transcription as well their combinatorial interactions. The enriched GO annotations of these nodes are the same as their putative functions. The tree contains 122 dimer motifs with variable gaps. These include the HSE motif (GAAnnTTC), Hap1 motif (CCGnnCCG), Gis1 motif (AGGGGCCCCT) as well as variants of the CCG everted repeat. A few examples of important biologically verified PSSMs learned by MEDUSA are given in Fig. 5.6.

5.7.1.2 DNA damage dataset

For the DNA damage dataset, we compare the probabilistic motifs discovered by MEDUSA to several known and putative binding sites of transcription factors. In addition to the TRANSFAC, SCPD, YPD and TFD databases we also use compare to motifs discovered by MacIsaac et al. [74] based on ChIP-chip data. Figure 5.7 shows a subset of the sequence motifs discovered by MEDUSA that have high scoring hits. We use the symmetrized Kullback-Leibler divergence as a distance metric for comparison. We calculate p -values based on the ALLR (averaged log likelihood ratio) score, where significance values were computed based on Karlin-Altschul statistics as reported by the MatAlign program (personal

communication, Dr. Gary Stormo). The p -value indicates the probability of observing an equal or higher ALLR score at random.

Workman *et al.* [129] recently identified a regulatory subnetwork for the DNA damage stress response in yeast using ChIP-chip experiments and gene expression data obtained from transcription factor deletion experiments. They target a set of 30 transcription factors believed to be important for DNA damage stress response. Of these, 25 have known or predicted binding sites. MEDUSA is able to find high scoring matches to 19 of these transcription factors, in addition to several others not used by Workman *et al.* [129]. These include several important cell cycle factors such as Ace2, Fkh1/2, Mcm1 and Swi4/5/6 as well as stress factors such as Yap1/2/3/4/5/6, Msn2/4 and Hsf1. Thus, we see that without using any prior knowledge of regulatory function, MEDUSA can automatically learn binding sites of key transcription factors involved in the DNA damage response.

5.7.2 MEDUSA regulatory programs uncover key regulators of the DNA damage signature

Gasch *et al.* [36] compare the gene expression response in the DNA damage dataset to that obtained from a diverse set of 13 environmental stress conditions [37]. Using hierarchical clustering, they identify only a small set of genes (Figure 5.8) that are specifically induced due to DNA damage, which they label the *DNA damage signature*. Among these signature genes, Rad51 and Rad54 are required for repair of DNA damage, while the *RNR* genes catalyze DNA synthesis and are some of the best studied targets of the Mec1 DNA damage response pathway [36]. We use the MEDUSA framework to study the regulatory phenomena behind this unique expression signature in 33 experiments involving MMS and radiation damage. We use the margin score to rank regulators and sequence motifs discovered by MEDUSA. Figure 5.8 shows a schematic representation of the regulatory program predicted by MEDUSA.

We find that MEDUSA is able to identify many transcription factors known to regulate

MEDUSA motifs	Database motifs	Transcription Factor	KL Distance	ALLR score	P-value
AGCTGGTat	GCTGGT	ACE2	0.120663	9.679	3.22E-08
cTCTCCG	TCTCC	ADR1	0.278808	8.505	1.69E-07
GACCTGA	RVACCTD	AFT2	0.587964	7.801	1.35E-06
TAGACAC	AGAcG	ARG80	0.900337	45.083	0
taatTTGATTgt	YTGAAT	ASH1	0.493051	6.417	2.29E-05
ACGTCAa	ACGTCA	ATF	0.097483	9.003	7.15E-08
aTGAAAAAt	AAGAAAA	AZF1	0.631665	6.299	3.52E-05
aataaGTAActa	TAANTAA	BAS2	0.901929	9.008	1.41E-07
ATTACAG	TTCACGTG	CBF1	0.941127	10.333	5.88E-09
tatGCGgTGaa	TGCGATGAG	CHA4	0.65007	46.23	0
CAAGCGG	CAGCGTG	CRZ1	0.887723	4.136	0.002215
GAAAAGA	GGAAAAAD	ECB	0.961028	13.666	4.80E-12
aTCGTAC	TCGTATA	ECM22	0.87614	4.308	0.001546
TACaaaaACAT	RYAACAWW	FKH1	1.29709	13.11	2.54E-11
aAGGGGGa	GGGG	GAL4	0.135774	7.575	1.08E-06
TAAGaaaaaTAAG	AGATAAG	GAT1	1.120242	53.073	0
AAGGGCC	GGGGCC	GC/FAR	0.966827	9.847	1.22E-08
TAAGTCA	AAGTCA	GCN4	0.315604	9.839	1.24E-08
GtCTTAC	GGCTTCaC	GCR1	1.116141	6.648	1.48E-05
CATCCTt	CATCC	GCR1	0.477299	7.107	3.60E-06
cCAGATAAc	GATAA	GLN3	0.509021	6.986	5.22E-06
TACCCAA	TACCAA	GTS1	1.066096	41.94	0
GATAAGAtaaa	GATAAG	GZF3	0.423012	38.822	0
CCAATGA	CCAATgAG	HAP2	0.535063	53.306	0
ttCCAAt	CCAAT	HAP3	0.721167	1.835	0.2006
TGAAaTTCA	NGAANNITCN	HSF1	0.71915	31.901	0
GAACTTC	GAANNITCC	HSTF	1.340932	20.007	0
ATGTG	CATGTG	INO2	0.370459	8.364	1.95E-07
aACAAAc	CAAA	KAR4	0.393091	6.129	2.23E-05
CCAATGG	TCCAATGGA	LYS14	0.55026	12.302	1.07E-10
aTGCTCAgt	TGCTCA	MAC1	0.253888	8.152	5.45E-07
CAATtttGTTA	YCNATTGTTW	mat1-Mc	1.005818	19.422	0
GTGTAG	TGATGAaAT	MATa1	1.380259	23.656	0
CGCGTAA	CGCGTAA	MBP1	0.018044	12.302	8.34E-11
cCCGCG	ACGCGT	MCB	0.589221	14.753	3.62E-13
CCAAaaaaTTGG	CCWWWWWWGG	MCM1	1.048785	44.754	0
ATGCAT	TATGCATT	MED8	0.809445	9.839	1.60E-08
tCTGTGT	CTGTGG	MET28	0.748681	3.666	0.00507
GTGTGTGC	gGTGTGGc	MET31	0.821885	17.842	9.99E-16
CCCGCGA	DCCCCGCGH	MIG1	0.724154	16.893	7.22E-15
AGGCATT	aAGGCA	MOT3	0.853234	40.701	0
aAAGGTAA	HAGGYA	MOT3	0.480557	10.123	7.83E-09
aaaaaGtCAAA	CRCAAAW	MSE	0.747361	6.318	3.94E-05
AAGGGCG	MAGGGGN	MSN2	0.81101	5.991	4.56E-05
GGGACCC	GGACCCCT	NRG1	0.631559	50.232	0
aCCCTT	CCCT	NRG2	0.030276	8.987	6.16E-08
CCGTGGAA	TCCGTGGA	PDR1	0.415259	10.711	3.04E-09
cGGCACct	AGGCAC	PHD1	0.545827	12.36	7.24E-11
acATTACcA	ATTAAg	PHO2	0.77705	14.976	3.40E-13
cGCACGTT	cGCACGTGgt	PHO4	0.403792	21.926	0
tGCCCGA	CCGA	RAF	0.72692	7.093	2.60E-06
AAACCTA	RMACCCA	RAP1	1.046281	1.475	0.4414
CGATCGA	TGACCGA	RC1	1.280144	5.414	0.0001528
AATGACC	GATGACC	RC2	0.646721	5.254	0.0002134
ttattGCACCTttt	TGCACCC	RCS1	0.543579	25.838	0
TTACCCaTt	TTACCCG	REB1	0.422826	16.379	2.11E-14
gACGGAT	CGGANNNA	RGT1	0.868243	5.697	8.44E-05
GCCAAGG	TGCCAAG	RIM101	0.677013	10.827	1.83E-09
GGTCACG	GGTCAC	RTG3	0.412244	10.827	1.57E-09
ATGTATGGg	ATGTACGGGT	SFP1	0.677032	26.623	0
GGCAGGG	GGCYGGC	SKN7	1.275842	5.908	5.43E-05
GCGAGAA	CGCGAAAA	STB1	1.122023	52.079	0
AtACCGCg	CCGCGG	STB5	0.675969	7.457	2.34E-06
TGAAAC	TGAAAC	STE12	0.0151	9.839	1.06E-08
aTACGGCG	cCGGGCGc	STP1	0.861497	8.296	4.78E-07
GCGGtttGCGG	RCGGCNNRCGGC	STP1	1.152129	30.053	0
GTGTGAC	WGTGACg	SUM1	0.966713	15.046	2.67E-13
gCGAAAGA	CNCGAAA	SWH	0.834734	14.591	7.92E-13
TGCTGGG	TGCTGG	SWI5	0.412277	10.827	1.57E-09
tTCGCGAcg	WCGCGW	SWI6	0.409984	13.865	3.49E-12
tatTATAATat	TATAAA	TBP	0.396691	5.313	0.0002774
CGGCCTc	CGGGCG	TEA1	0.622527	5.66	7.82E-05
aACATTTT	CATTCT	TEC1	0.726141	3.457	0.01008
TGTGAAT	ATGTGAaWW	UASINO	0.650638	14.651	7.85E-13
GCTCCCTG	CTTCCT	UASPHR	0.775921	4.696	0.0006724
aTCTGAAG	GtTCTGAgGtga	XBP1	0.569439	12.847	5.22E-11
tTACTAAt	TTACTAA	YAP1	0.417315	6.035	4.75E-05
AATAGCATttt	AAGCAT	YAP5	0.779864	40.719	0
TACCGGaa	ACCGGG	YDR026C	0.114926	11.455	4.82E-10
TTCCAAa	TTCGAA	YER051W	1.00147	41.679	0
tAATTGTG	TAATTG	YHP1	0.689182	5.458	0.0001365

Figure 5.7: Motifs identified by MEDUSA for DNA damage: MEDUSA motifs with most significant matches to database motifs are listed. Two filters are used to compute significance of the match, the symmetrized Kullback-Leibler divergence and the p -value for the ALLR score. The p -values are reported without correction for multiple testing.

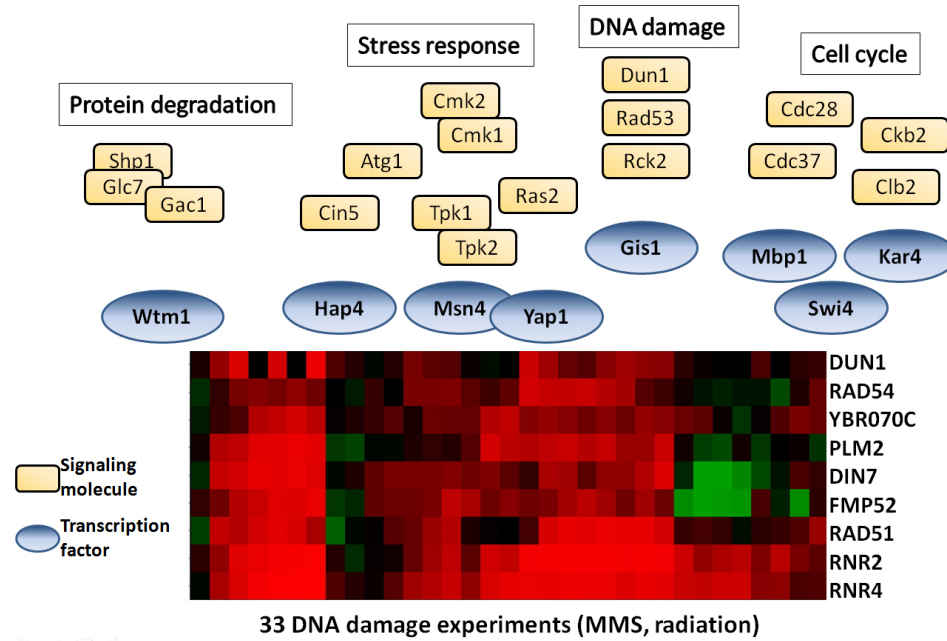


Figure 5.8: **Key regulators of DNA damage signature genes as identified by MEDUSA:** A network showing the predictive transcription factors and signal transducers identified by MEDUSA as regulators of DNA damage signature genes.

these target genes. Msn4 and Yap1 are stress response transcription factors that work in tandem [113]. The STRE element (AAGGGGt), which is the Msn4 binding site, is also among the top ranking predictive motifs. The mRNA profile of *HAP4*, which is involved in aerobic respiration, as well as its binding site (CCAAT) are found to be highly predictive. Gasch et al. [36] discuss the similarity between the DNA damage response and the hypoxia response, due to the involvement of the Rox1 transcription factor in both processes. It is therefore likely that Hap4 is also an important regulator of this DNA damage cluster. Wtm1, a protein that controls nuclear localization of Rnr2 and Rnr4 [70], is found to be a predictive regulator. One of the most important effects of DNA damage is cell-cycle arrest. Thus, it is interesting that we find the binding sites of Mbp1 (CGCGTAA), Swi4 (CNCGAAA) and Kar4 (CAAA) as high ranking motifs, as these are key cell cycle transcription factors [30].

We also find several components of the signaling pathways acting upstream of these transcription factors to be highly predictive in the MEDUSA regulatory program, including: Tpk1 and Tpk2, which are believed to regulate nuclear localization of Msn4; and Ras2,

which is the key component of the RAS pathway and affects targets through Yap1 and Msn4 [113]. Other MEDUSA-identified stress-related signaling components include Cmk1, Cmk2, Cin5 and Atg1. The role of the Mec1-Rad53 pathway in response to DNA damage has previously been recognized through its effects on *RNR* gene expression and cell-cycle arrest [52, 84, 99, 125, 142]. MEDUSA finds the Rad53 expression profile to be highly predictive. Also, Dun1, which is the only regulator that is part of the DNA damage signature, is predicted to have a strong regulatory role; it is a protein kinase required for the induction of the *RNR* genes and acts downstream of Mec1 [122]. Rck2, which is a radiation sensitive kinase, is also a highly-ranked regulator [24]. MEDUSA also finds several predictive cell-cycle related cyclins and kinases such as Cdc28, Cdc37, Ckb2 and Clb2 [30]. The highest ranking regulator found by MEDUSA is Shp1. It is a potential regulatory unit of Glc7 (also found to be predictive) and acts as an adaptor for protein degradation via the ubiquitin-proteasome pathway. Although there is no confirmed role for Shp1 in response to MMS or radiation induced DNA damage, it has been found that an *SHP1* null mutant shows increased sensitivity to bleomycin, which is a DNA damaging carcinogenic agent [3].

The DNA damage gene expression response is particularly confounding due to the strong general stress response component [37] and a strong post-transcriptional regulatory component. It is thus interesting to note that using our framework, we are able to filter out non-specific regulatory phenomena and highlight signaling components that are integral to the DNA damage pathways using mRNA expression and sequence data alone.

It is important to note that predictions made by MEDUSA based on the mRNA profiles of DNA-binding transcription factors do not necessarily represent direct binding; they could indicate indirect effects. However, evidence of a predictive binding site of a transcription factor discovered by MEDUSA suggests that the regulatory effect involves direct binding of the transcription factor.

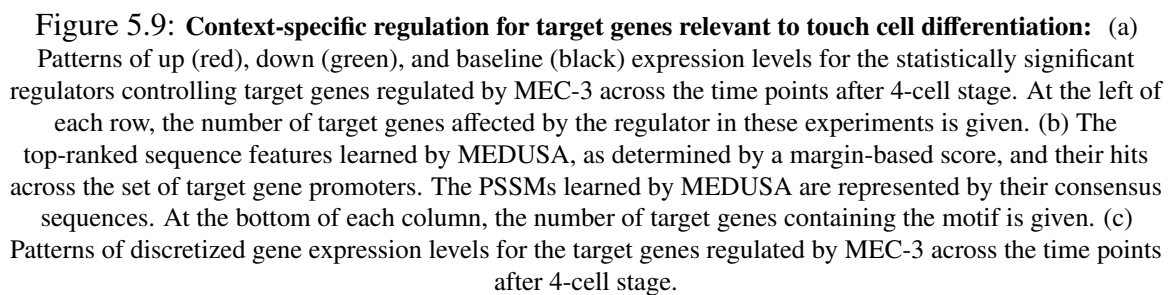
A striking difference we observe between the regulatory programs learned by MEDUSA for the DNA damage response and the general environmental stress response (ESR) is the

behavior of Msn4 as a predictive regulator. For the ESR dataset, Msn4 is predicted to be an important regulator primarily through its binding site (the STRE element) which ranks as the top scoring motif. The expression profile of MSN4, however, is only weakly predictive. In fact, a closer look at the expression profile of Msn4 in the ESR dataset shows that the fold changes are not very significant. This could indicate that Msn4's activity is primarily regulated by the PKA pathway through cellular localization. The Tpk1 kinase, one of the main components of the PKA pathway, is associated with the STRE motif in the regulatory program learned by MEDUSA. However, for the DNA damage response dataset, MEDUSA finds the binding site of Msn4 as well as its expression profile to be highly predictive. Several components of the PKA pathway are also found to be predictive. This could suggest that for the DNA damage response, Msn4 is specifically regulated both transcriptionally and post-transcriptionally.

5.7.3 Lineage-specific regulation in the early worm embryo

For the worm dataset, we compare the MEDUSA PSSMs learned in the first 500 boosting rounds against TRANSFAC and WormBook PSSMs. We find the binding site for Hlh-8, a helix-loop-helix transcription factor expressed in all body wall muscle cells from several cell lineages during embryogenesis, and for Mec-3, a transcription factor essential to touch cell differentiation in the neural lineage.

As a proof of principle that MEDUSA captures interesting context-specific regulation in this system, despite the limitation that the expression data came from whole embryo samples, we perform a case study relevant to touch receptor neurons. Six mechanosensory neurons (the touch cells) mediate the response of *C. elegans* to gentle touch. Experimental evidence suggests that the gene *MEC-3* encodes a transcription factor which specifies the differentiation of the touch cells [124, 131, 140], and a subset of 34 genes in our data set have been previously identified as Mec-3-dependent genes expressed in touch cells [140]. We first analyze this set of genes across all time points after the 4-cell stage during embryonic



development in order to find MEDUSA motifs which strongly affect this group of targets. We rank the motifs using the margin score (Section 5.4).

We find a Mec-3 binding site (ATCGAT) among the top motifs ranked by margin score. We also study two special time points, 53 and 83 minutes after the 4-cell stage, at which time Mec-3 is most up-regulated and potentially most active. In both cases, the same binding site scores the highest among all motifs. In this way, MEDUSA successfully discovers a Mec-3 motif despite the lineage-specific nature of touch cell differentiation. Figure 5.9 shows the most predictive motifs and regulators, as ranked by margin score, for the Mec-3-dependent genes.

5.7.4 Condition-specific regulators and motifs in human B cells

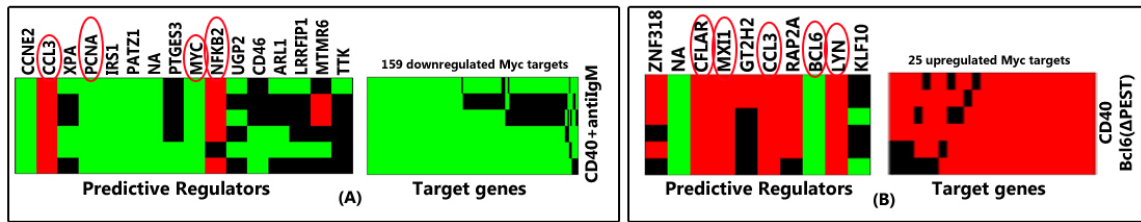


Figure 5.10: **Context-specific regulation in human B cells:** (A) Predictive regulators identified by MEDUSA for a set of 159 known MYC targets that are consistently downregulated in response to simultaneous CD40 and antiIgM stimulation. (B) Predictive regulators identified by MEDUSA for a set of 25 known MYC targets that are consistently upregulated in the BCL6(Δ PEST) mutant in response to the CD40 treatment. The regulators circled in red are known to have a key role in the conditions under study.

We restricted our analysis of the B cell data set (See Section 5.5.3) to a well-studied Burkitt lymphoma cell line (Ramos) treated in vitro to activate CD40 or B-cell receptor (antiIgM) signaling and cell lines engineered to stably express BCL6 and BCL6(*Delta*PEST) mutant. We use the margin-score based analysis (See Section 5.4) to study a set of 930 genes that are experimentally confirmed to be targets of the Myc transcription factor [138]. Below, we report results on two experimental groups.

The first group shown in Figure 5.10(A) is a set of 159 known Myc targets that are consistently downregulated in response to simultaneous CD40 and antiIgM stimulation.

We expect to see the strongest response from this group, since both the CD40 and BCR signaling pathways are active and regulate target genes through two transcription factors, Myc and NF- κ B. The nuclear factor- κ B (NF- κ B) family is essential for lymphocyte survival and activation and for mounting normal immune responses [71]. NF- κ B proteins are present in the cytoplasm in association with the inhibitory proteins I κ Bs. After activation of the CD40 and BCR pathways, the I κ B proteins become phosphorylated, ubiquitinated and, subsequently, degraded, which allows nuclear translocation of NF- κ B. NF- κ B can then transcriptionally activate its targets. Figure 5.10(A) shows that MEDUSA identifies both MYC and NF- κ B2 as predictive regulators. Also, we see that the expression of the targets is directly correlated with that of MYC, indicating its role as an activator. However, NF- κ B2's expression profile is inversely correlated with that of the targets. This could be because NF- κ B's activity is primarily controlled through cellular localization and hence its mRNA activity is not a direct indicator of its regulatory role. We also identify Ccl3 and PCNA as predictive regulators, both of which are downstream of the Bcl6 transcription factor. Bcl6 is believed to facilitate the proliferative expansion of germinal centers during normal immune responses through PCNA and its cell cycle arrest partner p21 [90].

The second group (Figure 5.10(B)) consists of a set of 25 known Myc targets that are consistently upregulated in the BCL6(Δ PEST) mutant in response to the CD40 treatment. BCL6 affects germinal center development and lymphomagenesis by transcriptional suppression of target genes controlling B cell activation and plasma cell differentiation. Bcl6 transcription is downregulated by signaling from the CD40 receptor [83]. Bcl6 protein stability is regulated by signaling from the B cell receptor that induces MAP kinase-mediated phosphorylation of Bcl6, thus targeting Bcl6 for rapid degradation by the ubiquitin-proteasome pathway. In the Bcl6(Δ PEST) mutant however, the Bcl6 protein is resistant to degradation due to the absence of the PEST domain. Thus, the mRNA activity of Bcl6 is largely representative of its regulatory activity. Since Bcl6 is a repressor, we would expect the targets genes' expression to be inversely correlated with Bcl6 expression. Figure 5.10(B) shows that

MEDUSA identifies Bcl6 as a predictive repressor of the target genes under study. Other predictive regulators identified by MEDUSA include CLFAR, Mxi1 and Lyn. CLFAR is involved in positive regulation of I- κ B kinase/NF- κ B cascades [97]. Mxi1 along with the Max gene antagonize Myc dependent activation [97]. The Lyn tyrosine kinase is part of the BCR signaling pathway [97]. Thus, we see that many of the regulators identified by MEDUSA have key roles in the conditions under study.

MEDUSA also learns the cACGTGAc and ggcTTTCctg motifs which are near exact matches to the known binding sites of the Bcl6 (CACGTG) and NF- κ B (gggACTTTCC) consensus binding sites respectively [58].

5.8 Conclusion

In this chapter, we propose a new algorithm called MEDUSA for learning binding site motifs along with a predictive model for gene regulation. MEDUSA jointly learns from promoter sequence data and multiple gene expression experiments, together with a candidate list of putative regulators (transcription factors and signaling molecules), and builds motif models whose presence in the promoter region of a target gene, together with the activity of regulators in an experiment, is predictive of up/down regulation of the gene. We can readily evaluate the predictive accuracy of the learned motifs and regulation model on test data, and we present cross-validation results for datasets of various sizes probing many different experimental conditions in organisms such as yeast, worm and humans. We see that MEDUSA is able to accurately predict expression data in all cases. We also show that MEDUSA's binding site motifs are better able to predict regulatory response on held-out experiments than binding site sequences taken from ChIP-chip transcription factor occupancy data, TRANSFAC motifs or previously published computationally-derived PSSMs.

Popular cluster-first motif discovery strategies often require heuristic or even manual

preprocessing to determine suitable putative clusters of coregulated genes. In practice, in addition to using gene expression profiles in the clustering algorithm, one might need to incorporate annotation data or even use hand curation to properly refine the putative clusters [53]. One must then carefully apply a standard motif discovery algorithm to find overrepresented motifs in the promoter sequences of genes in each cluster, which may involve optimizing parameters in the algorithm and thresholds for each of the extracted motif models. By contrast, MEDUSA avoids clustering and manual preprocessing altogether, and automatically determines PSSMs together with thresholds used for determining PSSM hits by optimizing boosting loss. In our experiments, MEDUSA learns many of the known binding site motifs in yeast.

The MEDUSA algorithm builds binding site motifs while producing a single regulation model for all target genes without introducing conceptual subunits like “clusters” or “transcriptional modules”. This single regulation model is arguably more biologically realistic and can capture combinatorial regulatory effects on overlapping sets of targets. The regulation model can also be interpreted as a gene regulatory network, since the activity of regulators predicts differential expression of targets via binding sites, although necessarily this network is large and contains many nodes. Nonetheless, we can use this model to address specific biological questions. We introduce the margin-score, which we use to analyze regulation of interesting gene sets in specific experimental conditions thus revealing context-specific regulators and motifs and analyze. Most of our observations are validated in the literature.

One difficulty of using complex parametrized models is that they require careful training methodologies to avoid poor local optima and severe overfitting. MEDUSA uses very few tunable parameters and can be run “out-of-the-box”, making it easy to reproduce results and allowing non-specialists to apply the algorithm to new datasets. Moreover, it is difficult to statistically validate the full structure or the components of complex network models; in the literature, most work using these models for gene regulation has focused on biological

validation of particular features in the graph rather than generalization measures like test loss. MEDUSA's predictive methodology—using large-margin learning strategies to focus on improving generalization—produces binding site motifs that achieve good accuracy for prediction of regulatory response on held-out experiments. The fact that we can easily evaluate the predictive performance of our learned motifs and regulation model gives us a simple statistical test of confidence in our results.

The superior performance of MEDUSA in discovering predictive motifs is very encouraging for applying such large-margin techniques to analysis of expression data for as-yet unannotated genomes and for elucidating the transcriptional regulatory mechanisms of more complex organisms.

Case Study: Regulation of hypoxia responses in yeast

In this chapter, we present a specific case study to showcase the post-processing and visualization aspects of our framework. We use GeneClass and MEDUSA to study the oxygen regulatory network in *S. cerevisiae* using a small data set of perturbation experiments. Mechanisms of oxygen sensing and regulation underlie many physiological and pathological processes, and only a handful of oxygen regulators have been identified in previous studies. We uncover detailed information about the global oxygen regulatory network. We also use biochemical experiments to validate several hypotheses generated by our analysis. This chapter is based on work presented in [61].

6.1 Introduction

Oxygen is critical for the survival and development of virtually all living organisms. As such, living organisms ranging from yeast to humans have developed sophisticated mechanisms to respond to changes of oxygen level [18]. Several microarray gene expression studies have been performed to understand oxygen sensing and regulation at a genome-wide level [63, 65, 93, 115, 118] in the yeast *Saccharomyces cerevisiae*. However, most of these

studies mainly identify genes responding to low levels of oxygen [63, 65, 93, 115, 118] or determine the DNA-binding sites for several known oxygen regulators, such as Rox1 [115]. Recently, there has also been a cluster analysis of expression profiles under hypoxia and reoxygenation in glucose versus galactose media [65, 66], where the authors looked for enrichment of functional annotations and known transcription factor binding sites within gene clusters and also applied existing motif discovery algorithms to the clusters. These previous microarray studies have provided further evidence of the role of known regulators such as Hap1, Rox1, and Upc2, but they have had limited success in identifying novel components of the oxygen and heme regulatory network.

In this chapter, we apply an integrative computational approach to analyze genome-wide changes in expression in response to perturbations of the oxygen regulatory network by varying levels of oxygen, heme, Hap1, and Cobalt (Co^{2+}). We use MEDUSA and GeneClass to learn predictive cis regulatory motifs and regulatory programs by integrating promoter sequence, promoter occupancy data from ChIP-chip experiments, and the expression levels of potential regulators. We use a novel margin-based score (See Section 5.4) to extract the condition-specific regulators and putative DNA binding site motifs that are most significant for predicting the expression of particular sets of target genes. We summarize this information with a global map of the oxygen regulatory network, which includes both known and novel regulators. Since MEDUSA associates regulators to target genes via motifs in the promoter sequence, we directly test the predicted regulators for the *OLE1* gene by experimental analysis of its promoter activity under deletion of each of these regulators. In each case, the change in *OLE1*'s promoter activity under hypoxia is found to be consistent with MEDUSA's predictions. These results confirm that several novel regulators are indeed involved in oxygen regulation. Finally, we perform a comprehensive comparison of the motif discovery results of MEDUSA with a conventional cluster-first motif discovery algorithm, and we find that MEDUSA identifies many DNA binding site motifs that are relevant to hypoxia and missed by the traditional approach.

The gene expression dataset that we analyze is a small dataset consisting of 24 microarray assays spanning 8 related experimental conditions. Most computational methods that learn regulatory networks or discover cis-regulatory motifs de-novo, require substantial amounts of data to learn reliable models. However, the reality on the ground is that most biology laboratories tend to produce small gene expression datasets similar to the one we analyze. It is thus important to be able to design algorithms that can learn reliable models using limited amounts of data to answer specific biological questions. GeneClass and MEDUSA use boosting (See Section 3.3) to avoid over-fitting as they search through the a massive space of possible regulators and sequence motifs. As a result, we achieve high prediction accuracy in cross-validation results. More important using our margin score and post-processing framework we are able to extract reliable signal from noisy data and reveal novel regulatory mechanisms. We show that even in this difficult setting, we are able to not only expose the hypoxia regulome but also capture subtle gene regulation at the level of a single gene.

6.2 Microarray experiments

In this section, we briefly describe the hypoxia dataset generated in the laboratory of our collaborator Dr. Li Zhang. Details of data generation and microarray data processing are presented in Appendix D. RNA samples are prepared from 8 different experimental conditions:

Normoxic (HAP1): Yeast cells bearing the Hap1 expression plasmid maintained under aerobic conditions.

Normoxic (Δ hap1): Yeast cells bearing the empty expression plasmid maintained under aerobic conditions i.e. *HAP1* is deleted.

Anaerobic, early (HAP1): Yeast cells bearing the Hap1 expression plasmid maintained under anaerobic conditions for 1.5 hours.

Anaerobic, late (HAP1): Yeast cells bearing the Hap1 expression plasmid maintained under anaerobic conditions for 6 hours.

Anaerobic, late (Δ hap1): Yeast cells bearing the empty expression plasmid maintained under anaerobic conditions for 6 hours.

Normoxic, +Co²⁺ (HAP1): Yeast cells bearing the Hap1 expression plasmid in the presence of 400 μ M cobalt chloride for 6 hours.

Heme sufficient (Heme): Yeast cells grown in medium containing 250 μ g/ml (heme-sufficient) 5-aminolevulinic acid.

Heme deficient (Δ Heme): Yeast cells grown in medium containing 2.5 μ g/ml (heme-deficient) 5-aminolevulinic acid.

For each condition, three replicates are generated by preparing RNA samples from three batches of independently grown cells. The gene expression data is obtained from single channel Affymetrix microarrays. Each of the knockout, stress or perturbation microarray experiments is compared to a corresponding reference microarray. The expression fold-changes are converted to p -values using an intensity-specific noise model obtained from replicate data (See Section 5.5.1.1). The fold-changes are then discretized into +1, 0 or -1 labels using a p -value threshold of 0.05. A label of +1 (-1) indicates up-regulation (down-regulation) beyond the threshold level of noise.

6.3 Perturbations of the oxygen regulatory network reveal diverse expression signatures

Prior to performing more integrative computational analysis, we examine the broad patterns of gene expression in our data set. These results also allow us to compare MEDUSA to

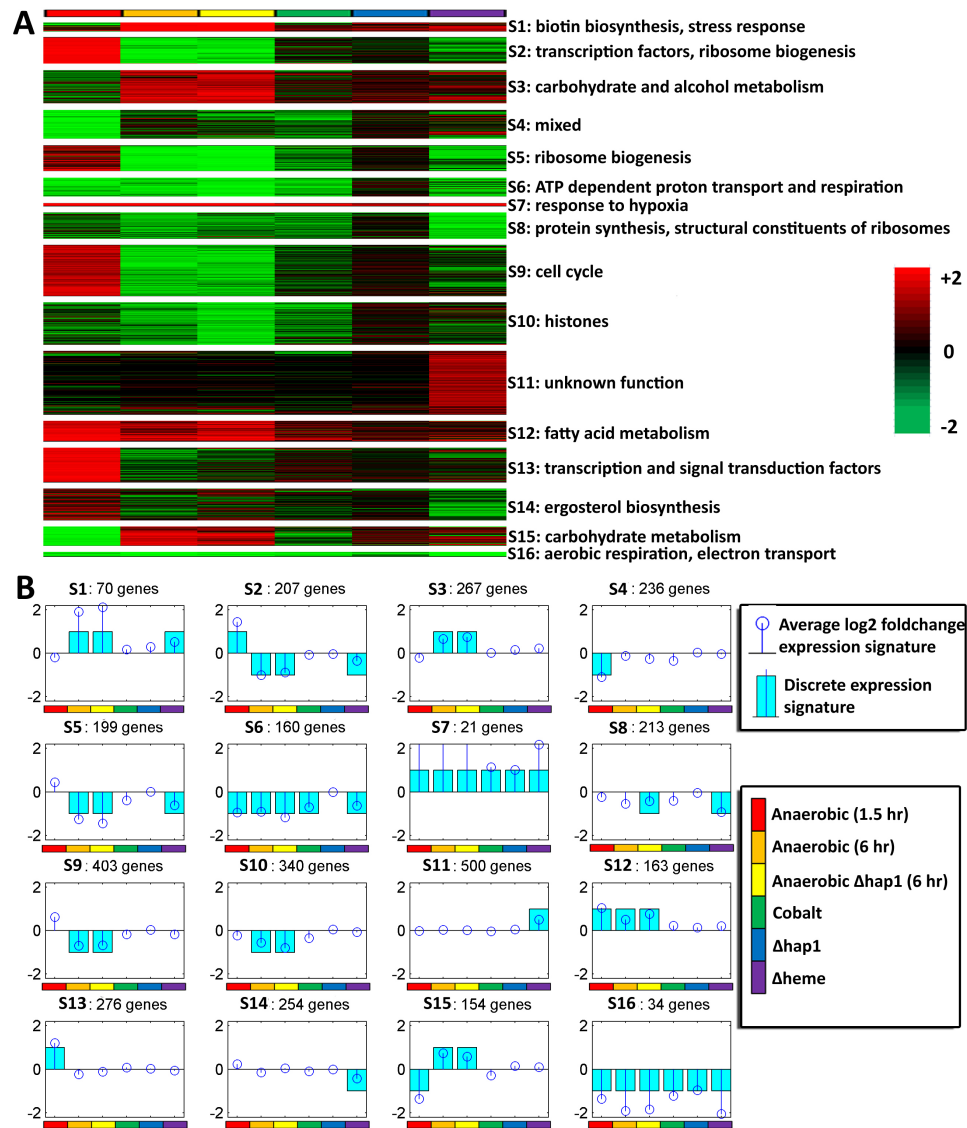


Figure 6.1: Expression signatures identified by perturbation of the oxygen regulatory network. (A) Heat maps showing real-valued expression profiles of genes that are members of the 16 signatures identified. The expression values are in \log_2 . The rows represent genes and the columns represent the 6 experimental conditions. Bright red indicates strong upregulation, bright green indicates strong downregulation, and black indicates no change in expression. Each signature is labeled with statistically significant functional annotations (B) Each block displays the average real-valued expression (stem plot in dark blue) and discrete expression profile (bar plot in light blue) for each signature over the 6 experimental conditions. The real-valued expression values are in \log_2 .

other “cluster-first” motif discovery methods. We use a two-phase procedure to partition the genes into expression signatures.

The first phase is used to identify the number of unique signatures. We first average the expression data for each gene in each experimental condition over all replicates. We then discretize this expression data into 2 levels (± 1) based on the sign of the foldchange in expression. We group genes into sets of unique patterns across all experiments. We then average the real-valued expression data for all genes belonging to each pattern to obtain a “mean expression signature”. Patterns with small support (< 10 genes) are hierarchically merged with their nearest pattern until there are no patterns with < 10 genes. The nearest neighbor is determined using a square-euclidean distance metric over the mean expression signatures for each pattern. This procedure groups the 3358 significantly expressed genes into 16 sets.

The second phase is used to refine the clustering. We recluster the genes into 16 expression signatures using the k -means clustering algorithm. We use the square-euclidean distance metric over the real-valued expression data. In an attempt to avoid local minima, we repeat the clustering procedure 10 times with different starting points. The initial candidate genes for cluster centroids are obtained by randomly sampling the 16 gene sets identified in the first phase. The final expression signatures are calculated by taking a majority vote over the 3-level ($+1/0/-1$) discretized expression for all genes in each gene set.

As shown in Figure 6.1, we identify 16 distinct discretized co-expression signatures to which we assign the differentially expressed genes. We perform Gene Ontology functional analysis (See Section 5.5.1.4) on the 16 expression signatures. In most cases, the expression signatures can be assigned significant functional terms, though in general only a smaller subset of the genes in a signature belong to the enriched category. Detailed functional analysis of each of the expression signatures is presented in Appendix E.

6.4 Learning procedure

We use a two phase procedure to learn a global hypoxia regulatory program using gene expression data, gene promoter sequences, ChIP chip data and a global set of potential regulators. The first phase is a de-novo motif discovery phase. We use the MEDUSA algorithm introduced in Chapter 5 for 450 iterations thus obtaining a set of 450 PSSMs, their targets and thresholds. We decide on the stopping criterion based on the number of iterations required for the mean test-loss on cross-validated folds to plateau. In the second phase of regulatory program learning, we use GeneClass for 450 iterations using all the data for training and using the 450 MEDUSA PSSMs as well as the ChIP-chip occupancy data as input.

We use a candidate set of 507 regulators consisting of 240 known and putative transcription factors and 267 known and putative signaling molecules such as kinases, phosphatases and receptors.

For the motif discovery phase in MEDUSA, we use 1000 bp. promoter sequences upstream of the transcription start site of all *S. cerevisiae* genes. We scan these sequences for all occurring k -mers ($k = 3 \dots 7$) as well as 3-3 and 4-4 dimer motifs allowing a middle gap of up to 8 bp. We restrict the set of all dimers to those whose two components have specific relationships, consistent with most known dimer motifs: equal, reversed, complements, or reverse complements.

We use ChIP-chip data from Harbison *et al.* [42] for 203 transcription factors in living yeast cells under 13 diverse environmental conditions. We discretize the data using a p -value threshold of 0.1.

The resulting alternating decision tree (ADT) consists of 450 splitter-prediction node pairs. Each splitter node consists of a stabilized set of regulators in combination with a stabilized set of PSSMs and/or ChIP chip occupancy profiles. The PSSMs learned by MEDUSA dominate the tree. The earliest ChIP chip feature is observed at iteration 121. It is the ChIP chip occupancy profile of the Hap1 transcription factor which is a key hypoxia

regulator. The ADT is at most 3 levels deep. There are 361 single node paths, 87 paths consisting of 2 nodes and only 2 paths consisting of 3 nodes. This final model is used for all biological post-processing analysis.

Prediction accuracy using different cross-validation experiments are provided in Section 5.6.1. The high prediction accuracy (92%) on held-out data gives us confidence that the algorithm is not overfitting the small dataset and that our model is reliable.

We introduced the margin score in Section 5.4. The score assesses how significantly an individual feature contributes to the confidence of predictions over a specific set of target genes and experiments. We use this score to extract and rank regulators and motifs from the regulation programs specific to different gene sets and experimental groups.

6.5 Cis regulatory motifs: Comparison to “cluster-first” motif discovery algorithms

In this section, we show that MEDUSA is able to automatically learn the DNA binding sites of almost all known hypoxia-related transcription factors and several new cis regulatory motifs. We also do a global comparison of transcription factor binding motifs attributed by MEDUSA to those found by AlignACE [53] which is a “cluster-first” motif discovery algorithm. Since AlignACE [53] is only applicable to clusters of genes we use MEDUSA and AlignACE to reveal motifs relevant to the 16 expression signatures (Section 6.3) in our data set.

6.5.1 Global comparison of motifs

Figure 6.2 shows a comprehensive comparison of MEDUSA to AlignACE motif discovery results across all 16 signatures. We use AlignACE with default settings on 1000 base pair promoter sequences of genes belonging to each signature and use AlignACE’s *maximum a posteriori (MAP) score* [53] to rank the overrepresented motifs by their statistical signifi-

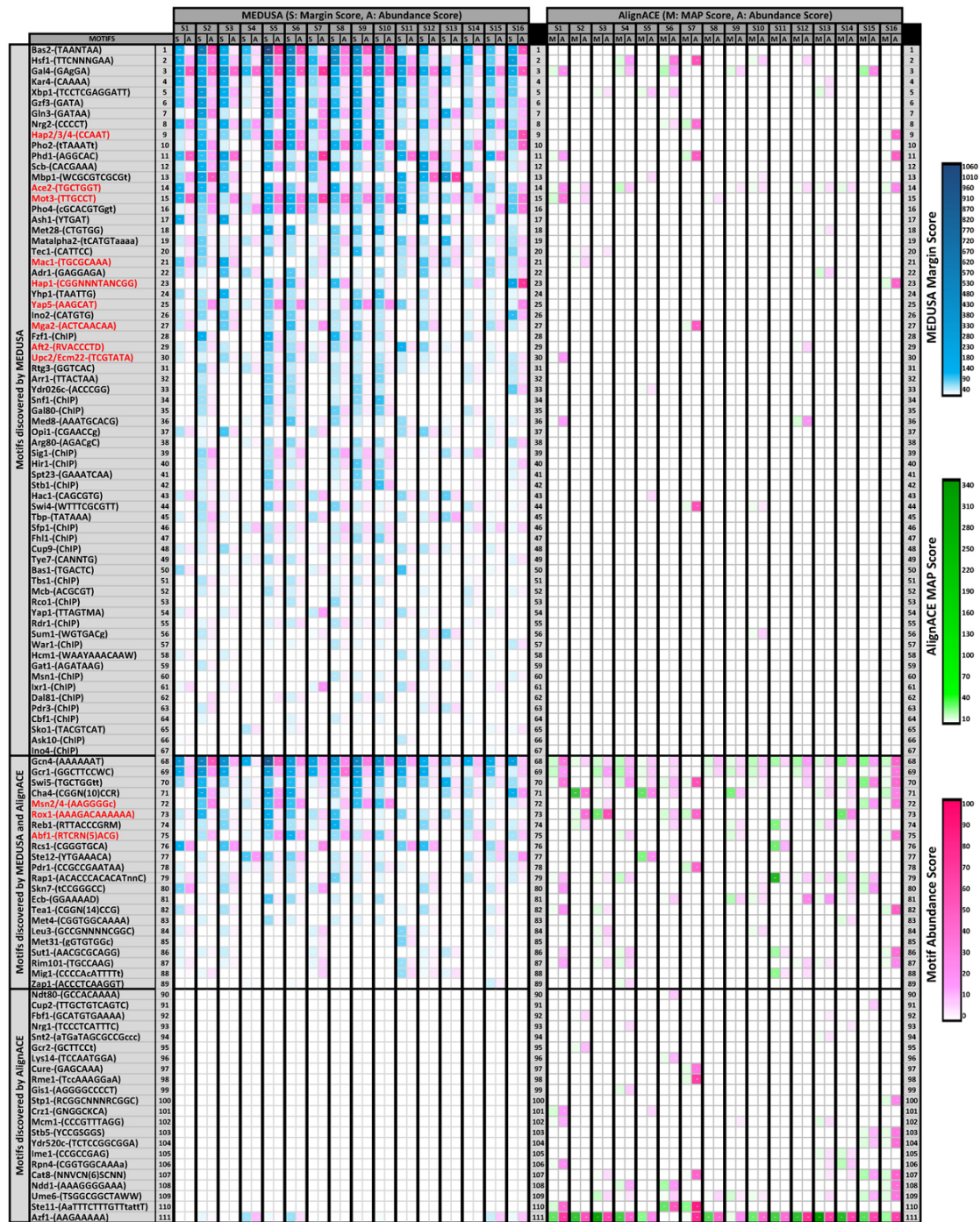


Figure 6.2: Comparison of motifs learned by MEDUSA and AlignACE for the 16 expression signatures identified in the hypoxia dataset

cance. We also define the *abundance score* for each motif as the fraction of promoters that were found to have the motif.

Similarly, we use margin scoring to identify significant MEDUSA motifs for each of the signatures, reporting only those motifs with a positive margin score. For MEDUSA, we define the *abundance score* for a motif as the fraction of promoters in each signature set that are found to have the motif based on the tree structure of the learned model. In order to compare the two methods, we report in Figure 6.2 only those motifs that match known transcription factor binding sites' PSSMs in TRANSFAC, SCPD or YPD (using Kullback-Leibler divergence to compare motifs) or match consensus sequences found by Macisaac *et al.* [74]. If multiple motifs are found to be strong matches to the same known binding site, we report the one with the highest statistical score. In total, we match 111 motifs found by either one or both methods to known binding sites, and we sort these motifs into 3 categories based on the difference between the cumulative margin score and cumulative MAP score across all the signatures.

As seen in Figure 6.2, the first set consists of 67 motifs identified by MEDUSA but not by AlignACE; the second set consists of 22 motifs that are identified by both MEDUSA and AlignACE; and the third set consists of 22 motifs that are identified by AlignACE but not MEDUSA. In Figure 6.2, the motifs highlighted in red are binding sites of transcription factors known to play a key role in hypoxia-related conditions. MEDUSA is able to identify a number important hypoxia-related transcription factor binding sites, including Hap1 (CGGnnTAnCGG), Hap2/3/4 (CCAAT), Mga2 (ACTCAACAA), Upc2/Ecm22 (TCGTATA), Ace2 (TGCTGGT), Mot3 (TTGCCT), Mac1 (TGCGCAA), Aft2 (RVACCCTD), Msn2/4 (AAGGGGc), Rox1 (AAAGACAAAAA) and Abf1 (RTCRnnnnnACG). Among these, AlignACE is able to identify Rox1, Msn4 and Abf1, and it finds the Hap1 and Hap2/3/4 binding sites only for a single signature (signature 16). Moreover, none of the motifs exclusively identified by AlignACE are known to have any role in the hypoxia-related conditions. In particular, the top scoring AlignACE motif is a low complexity motif

(AAAAAAAA) that matches the Azf1 binding site. These results show that MEDUSA outperforms AlignACE in finding relevant sequence motifs for our data set.

6.5.2 Motifs specific to a functional regulon

We further analyze whether MEDUSA can correctly extract regulatory information about a known functional regulon, and we compare the results of MEDUSA analysis with AlignACE [53]. A functional regulon is a group of genes experimentally confirmed to be coregulated by a common set of transcription factors. The functional regulon we examine is signature 16 (See Figure 6.1), one of the smaller expression patterns consisting of a set of 34 *HAP1*-dependent genes that are strongly suppressed in all conditions including the Δ hap1 experiment. These genes (such as the *COX* and *QCR* genes) are involved in aerobic respiratory processes, electron transport and heme-dependent oxidoreductase activity. Starting with the global regulatory program, we extract all sequence motifs and regulators with positive margin score for this set of genes and further rank them based on the number of target genes they are predicted to regulate.

We first consider transcription factors that are identified as high ranking regulators and motifs i.e. they are high scoring regulators based on their mRNA expression profiles and their binding site motifs are discovered by MEDUSA and ranked as significant and frequent for the 34 genes in the signature. This set of transcriptional regulators consists of Hap1, Mot3, Ace2, Mac1, Msn2, Ste12, Gcn4, Pho4 and Hap2/3/4. MEDUSA also ranks the Hap1 and Hap4 ChIP chip occupancy profiles as highly significant features for these genes. By examining the literature, we find support for many of these transcription factors regulating at least several of the target genes in this set. For example, *CYC1*, *CYC7* and *CYT1* are known to be directly regulated by Hap1 [114]. Hap2/3/4 are known to directly affect expression of *COX4*, *COX5*, *COX6*, *CYC1* and *CYT1* [114]. Mot3 is known to directly regulate *CYC1* expression [40]. *FRE1* and *CTR3* are known to be regulated by Mac1 [64, 114]. Upc2 and Mga2 are also important hypoxia regulators, and MEDUSA identifies their binding sites

as high scoring motifs for smaller subsets of genes within the signature. MEDUSA also identifies Abf1 as a significant regulator through its mRNA expression, but the sequence motif corresponding to the Abf1 binding site and the Abf1 ChIP chip data have low margin scores. Some of the *COX* genes are known to be regulated by Abf1 in other conditions [114], and the Abf1 binding site is present in several of the genes. In this case, our MEDUSA analysis suggests mixed evidence for Abf1 as a transcriptional regulator of the regulon, and it is possible that under hypoxia other regulators dominate.

As a comparison, we also use AlignACE to find overrepresented sequence motifs in the promoter regions of this gene set. Since the signature is small and represents a true functional regulon, it provides an ideal case for traditional motif discovery algorithms. Using the motif discovery program in the most permissive way (that is, without enforcing any significance threshold on the motifs), AlignACE is only able to find significant hits for the binding sites of Gcn4, Cha4, Hap1 and Ace2. For Hap2/3/4, the MAP score (3.4) has very low statistical significance, even though the motif is very abundant in this gene set (46.1%) and is known to regulate most of these genes [114]. Also, AlignACE is not able to identify more subtle context-specific regulators such as Mot3, Mac1, and Mga2, which are known to regulate these genes.

6.6 Post-processing and visualization framework

In this section, we present our post-processing framework and show several interesting visualizations of the data and features extracted from our learned models. These methods help answer specific types of biological questions.

We show how our predictive framework can focus on different scales of gene regulation. We are able to reveal regulators of a single gene *OLE1* which we validate using wet-lab experiments. Further, we are able to segment the context-specific activity of regulators and motifs in each of the experimental conditions by analyzing massive groups of up and down-

regulated genes. At the genome-wide scale, we are able to expose the hypoxia regulome consisting of several known and novel master regulators that mediate the hypoxia response.

6.6.1 Regulation of the *OLE1* gene: Biochemical experiments validate hypotheses

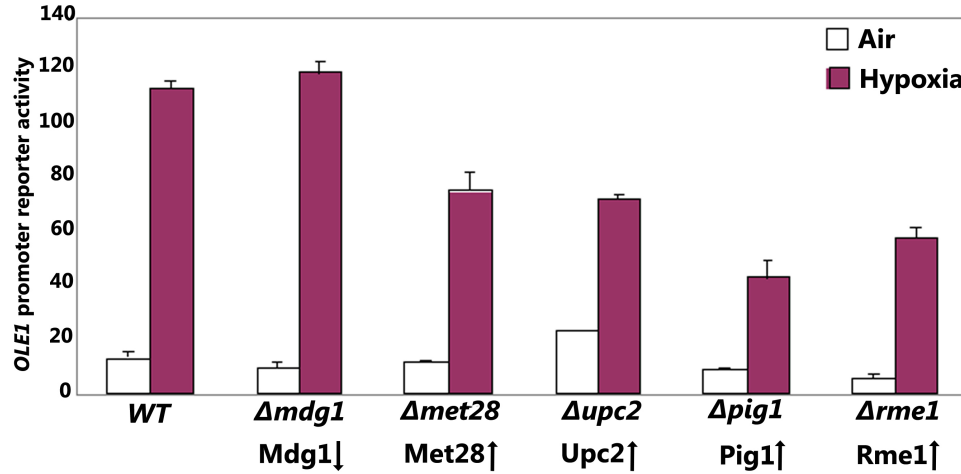


Figure 6.3: **Experimental confirmation of the oxygen regulators identified by MEDUSA:** MEDUSA identifies Mdg1, Met28, Upc2, Pig1 and Rme1 as specific regulators of the hypoxia-inducible *OLE1* gene. To detect the effects of these regulators on the *OLE1* gene, the full-length *OLE1* promoter-lacZ reporter was transformed into the wild type or mutant cells with one of the indicated genes deleted. β -galactosidase activities were measured in cells grown in air or in hypoxic chamber. Data plotted here are averages from at least three independent transformants. The arrows indicate the effects of hypoxia on the expression levels of Mdg1, Met28, Upc2, Pig1 and Rme1. That is, Mdg1 was downregulated whereas the rest were upregulated in hypoxic cells.

To experimentally test specific regulators identified by using MEDUSA, we examine the promoter of the *OLE1* gene, because this promoter has been well characterized previously [21,55,121]. The full-length *OLE1* promoter-lacZ reporter activity is strongly induced by hypoxia [121]. By applying margin-based scoring to a set of previously identified *OLE1*-like genes [21,55,121], we extract significant *OLE1*-specific motifs and regulators under hypoxia. Among the significant motifs, we find LORE (low oxygen response element), which has been experimentally determined [121] and is known to be the Mga2 binding site [55], as well as the binding sites for Hap1 and Aft1/2, which are also known to bind

OLE1 [74].

Using margin-based scoring for regulators, we identify Mdg1, Met28, Upc2, Pig1 and Rme1 as potential regulators for the *OLE1* promoter. Only Upc2 is previously known to be involved in oxygen regulation. The expression of these regulators but Mdg1 is upregulated by hypoxia. Note that the MEDUSA model does not assert that these regulators directly bind the *OLE1* promoter but does predict that they regulate *OLE1* expression, perhaps through indirect interactions. To determine the effects of these regulators on the *OLE1* promoter-*lacZ* reporter activity, we measure β -galactosidase activities in wild type and mutant cells with one of the regulator genes deleted (Figure 6.3). Except for Δ mdg1 cells, the reporter activity in hypoxic mutant cells are all reduced, compared to that in wild type cells (Figure 6.3). Because hypoxia suppresses *MDG1* expression, indicating its negative role in *OLE1* induction, it is conceivable that its deletion will not affect the reporter activity in hypoxic cells. In contrast, because hypoxia induces the expression of other regulators, indicating their positive role in *OLE1* induction, their deletion causes the reporter activity to decrease in hypoxic cells. Deletion of *MET28*, *UPC2*, and *PIG1* also significantly reduce the fold induction of the *OLE1* reporter activity by hypoxia (Figure 6.3). These experimental results strongly support the power of our methods to predict regulators of individual genes.

6.6.2 Context-specific regulators in different experimental conditions

In order to identify the most significant regulators and motifs controlling specific sets of differentially expressed target genes under specific experimental conditions, we rank regulators and motifs using the margin score.

We first use margin scoring to identify statistically significant regulators that may mediate the regulation of various target gene sets. Figure 6.4 provides an example of the condition-specific regulators and motifs identified by MEDUSA. To clarify the roles of Hap1 and heme in oxygen regulation, we identify and compare the potential regulators (Figure 6.4A) and motifs (Figure 6.4B) that may mediate the regulation of hypoxically

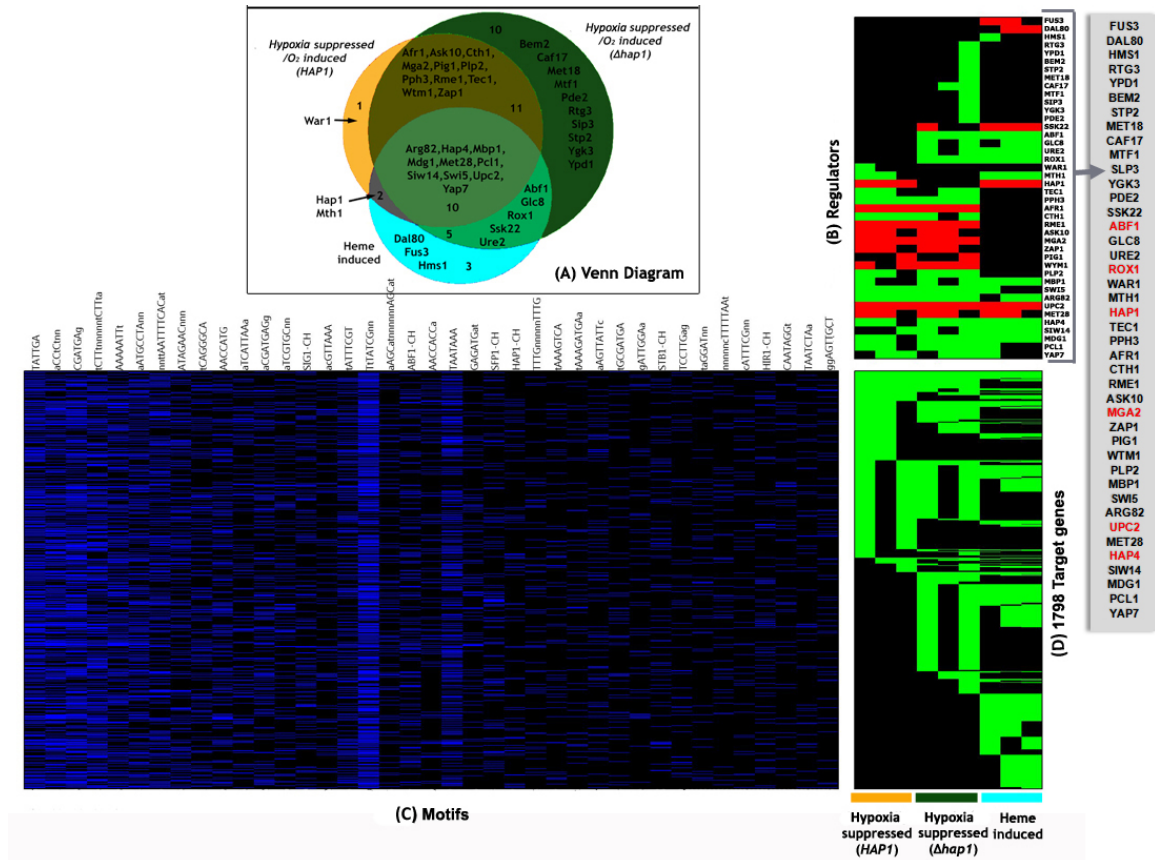


Figure 6.4: Heat maps showing predictive regulators, predictive motifs, and targets induced by oxygen and heme: (A) A Venn diagram illustrating the regulators involved in controlling hypoxically suppressed (oxygen-induced) genes in HAP1 and $hap1$ cells, and heme-induced genes. For each experiment, the statistically significant regulators associated with the set of downregulated target genes are determined by use of a margin-based score. (B) Patterns of up (red), down (green), and baseline (black) expression levels for the statistically significant regulators controlling downregulated target genes across the three experimental conditions. (C) The top-ranked sequence features learned by MEDUSA, as determined by a margin-based score, and their hits across the set of target gene promoters. The PSSMs learned by MEDUSA are represented by their consensus patterns. ChIP-chip occupancy features also occur in the list of most significant features. For example, SIG1-CH refers to ChIP-chip occupancy by the transcription factor SIG1 and appears as a highly-ranked promoter sequence feature. (D) Discretized gene expression levels for the full set of target genes represented in the Venn diagram (total of 1798 genes), given by combining the down-regulated target gene list from each of the three experimental conditions. Note that the expression patterns include only down and baseline expression levels across all three conditions.

suppressed (oxygen-induced) genes in wild type HAP1 or hap1 cells, and heme-induced (heme deficiency-suppressed) genes. Note that intracellular heme levels are low under hypoxic growth conditions [50], so hypoxically suppressed (oxygen-induced) genes correlate with heme deficiency-suppressed (heme-induced) genes. The expression levels of identified regulators and target genes are indicated in Figure 6.4C and 6.4D, respectively.

MEDUSA analysis identifies a number of known regulators whose predicted condition-specific role is consistent with previous knowledge of oxygen and heme regulation. For example, consistent with existing knowledge [139, 143, 144], Hap1 is important for the regulation of oxygen-induced genes in cells bearing the Hap1 expression plasmid and for heme induction of target genes (see Figure 6.4A and 6.4C). Likewise, Hap4 is important for heme induction and for the regulation of oxygen-induced genes in both cells bearing the Hap1 expression plasmid (HAP1) and the empty vector (Δ hap1, Figure 6.4A and 6.4C). Rox1 appears to be important for the regulation of oxygen-induced genes only in Δ hap1 cells. This is not surprising because Rox1 expression is known to be under the control of Hap1 [143, 144]. In cells bearing the Hap1 expression plasmid (HAP1), Hap1 would be the dominant regulator. Another notable case is Mga2, which has been shown to be important for oxygen regulation of certain genes, such as *OLE1* [51, 55]. Here we find that it is indeed important for oxygen induction of genes in both cells bearing the Hap1 expression plasmid (HAP1) and the empty vector, but it is not important for heme regulation, as expected.

We also identify and compare statistically significant regulators that may mediate the regulation of hypoxically induced genes in cells bearing the Hap1 expression plasmid (HAP1) and the empty vector (Δ hap1) and those that mediate heme deficiency-induced (heme-suppressed) genes (Figure 6.5A). Likewise, we identify and compare regulators that may mediate the regulation of Co^{2+} -inducible genes with those mediating the regulation of hypoxically induced genes (Figure 6.5B). The comparison of these regulators mediating oxygen regulation, heme regulation, and Co^{2+} -inducible regulation provides several important insights into the regulatory network mediating oxygen sensing and regulation.

First, more than half of the MEDUSA-identified regulators mediating heme regulation may also be involved in mediating oxygen regulation both in HAP1 cells (12 out of 20 regulators) and in Δ hap1 cells (15 out of 20 regulators) (Figure 6.4A). Many regulators predicted to be involved in heme suppression of target genes may also be involved in oxygen induction in wild type HAP1 cells (13 out of 18) and in Δ hap1 cells (11 out of 18) (Figure 6.5A). These results are consistent with the previous idea that heme serves as a secondary messenger of oxygen and plays a major role in mediating oxygen regulation of target genes.

Second, Hap1 plays a major role in oxygen regulation. In the absence of Hap1, the number of regulators mediating oxygen regulation may be significantly increased both for oxygen-induced genes (Figure 6.4A) and hypoxically induced genes (Figure 6.5A).

Third, relatively few regulators may be involved in mediating the regulation of hypoxically induced and Co^{2+} -inducible genes (Figure 6.5B). These results suggest that the Co^{2+} -inducible oxygen regulatory pathway plays only a minor role in mediating oxygen sensing and regulation. MEDUSA identifies the regulators that may mediate the regulation of oxygen-regulated genes that are affected at the early stage (1.5 hours) of anaerobic growth (Figure 6.5C and D) in cells bearing the Hap1 expression plasmid (HAP1), finding some regulators common to both time points and some specific to early or late stages. The results from analysis of both target genes and regulators (Figure 6.5C and 6.5D) suggest that there is a significant switch in the regulatory and expression programs in the cells as anaerobic conditions prolong.

6.6.3 The hypoxia regulome

To reveal the statistical importance of various regulators in the global oxygen and heme regulatory network, we summarize our results by using a global regulatory map (Figure 6.6). Figure 6.6 illustrates the significance of the regulators for predicting the up or down regulation of target genes under the tested six different experimental conditions, ranked by margin

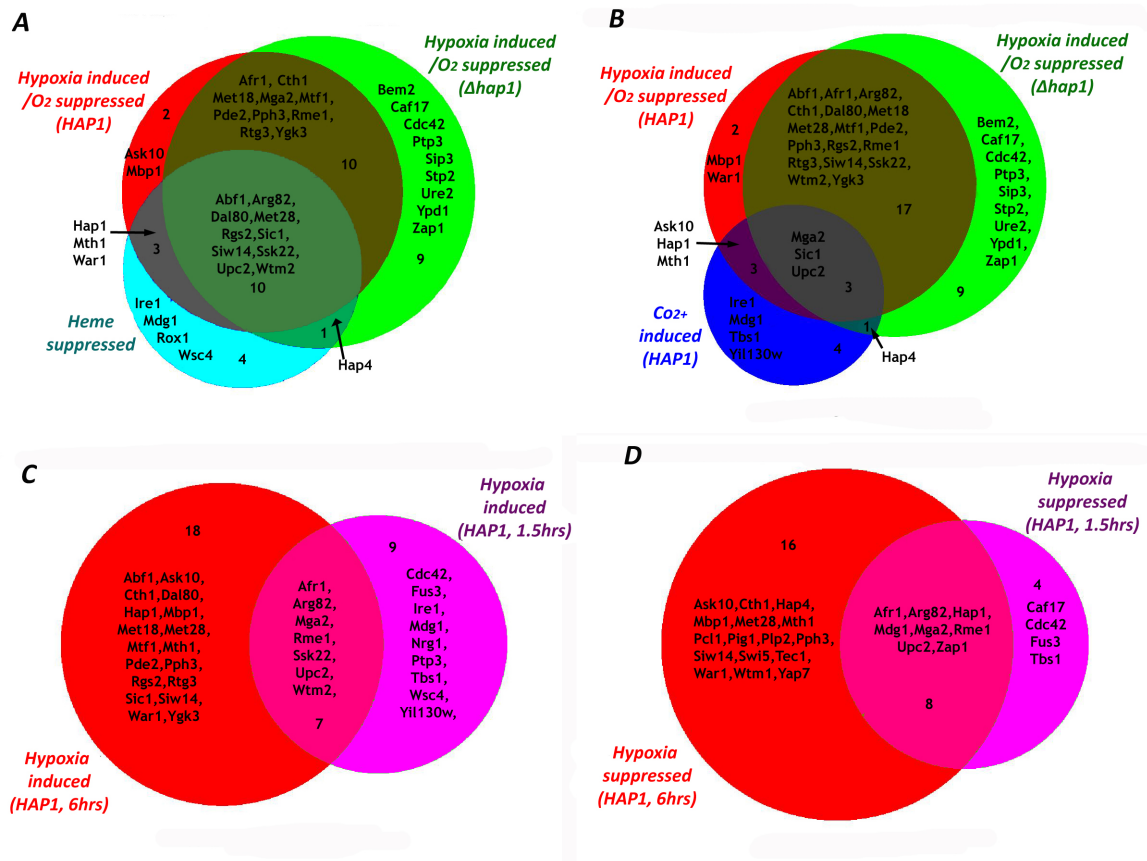


Figure 6.5: Venn diagrams showing the statistically significant, high ranking regulators mediating the regulation of oxygen-regulated, heme-regulated, and Co²⁺-inducible genes in HAP1 and Δ hap1 cells: . (A) A Venn diagram illustrating the regulators involved in controlling hypoxically induced (oxygen-suppressed) genes in HAP1 and Δ hap1 cells, and heme-suppressed genes. (B) A Venn diagram illustrating the regulators involved in controlling hypoxically induced (oxygen-suppressed) genes in HAP1 and Δ hap1 cells, and Co²⁺-inducible genes. (C) A Venn diagram illustrating the regulators involved in controlling hypoxically induced (oxygen-suppressed) genes in HAP1 cells at 1.5 or 6 hours after shifting to anaerobic growth conditions. (D) A Venn diagram illustrating the regulators involved in controlling hypoxically suppressed (oxygen-induced) genes in HAP1 cells at 1.5 or 6 hours after shifting to anaerobic growth conditions.

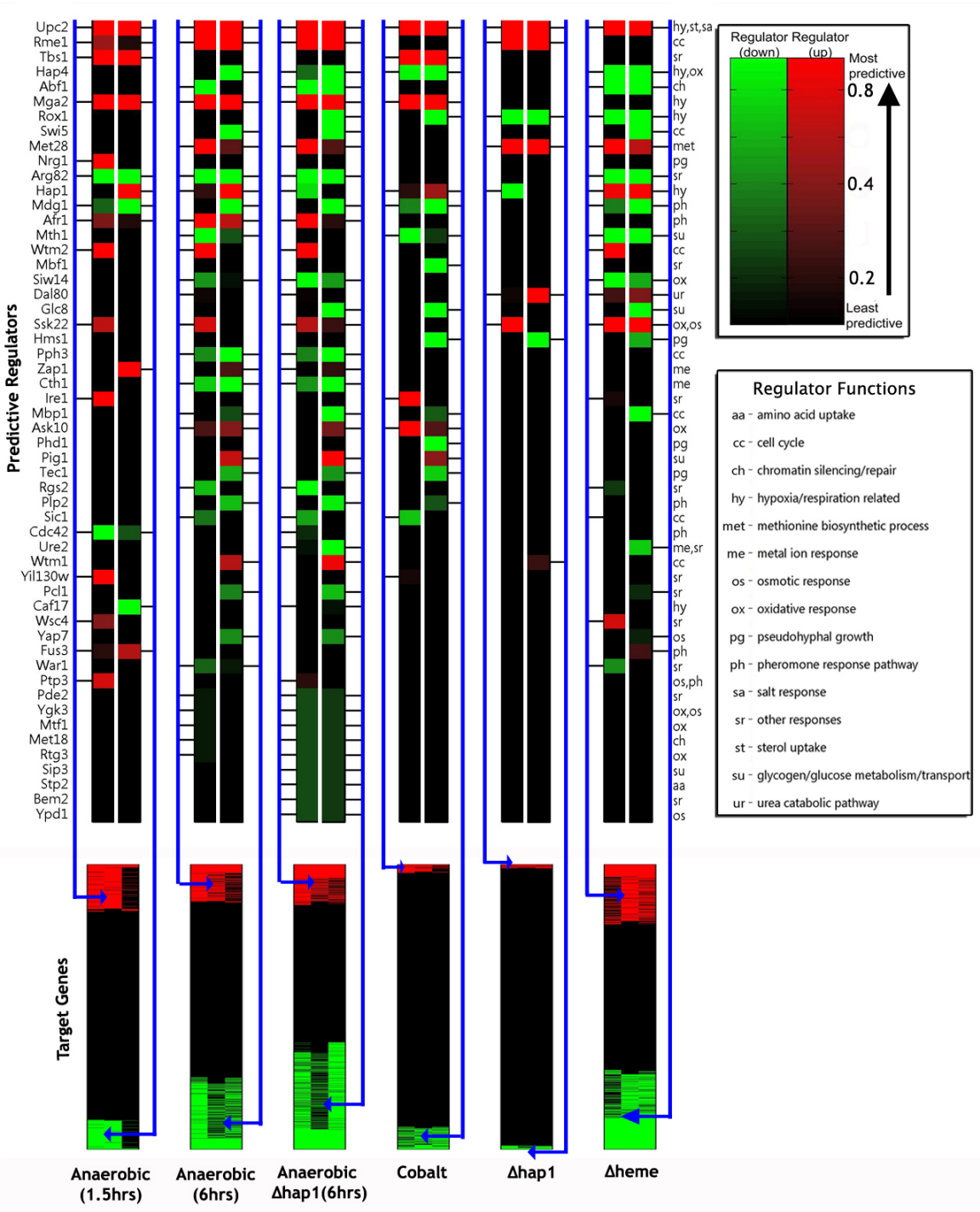


Figure 6.6: The hypoxia regulome: Our methods reveal a hypoxia regulome of 54 predictive regulators. For each experimental condition, we show the state of the regulator in red (upregulated) or green (downregulated), where the brightness of the color indicates the significance of its contribution to up or down predictions for the targets, based on normalized margin score. Regulators are ranked from top to bottom in order of overall predictive significance across experiments. The functional category for each regulator is indicated by an annotation given at the right of the figure and explained in the legend.

score. We identify 54 predictive regulators in the hypoxia regulome. In Figure 6.6, for each condition, we show the state of the regulator in red (upregulated) or green (downregulated), where the brightness of the color indicates the significance of its contribution to up or down predictions for the targets, based on normalized margin score. Significance of the regulators to the up-regulated targets is shown in the left half of the column, while contribution to the down-regulated targets is shown in the right half. Some regulators contribute significantly to the prediction of both up- and down-regulated targets within a condition due to indirect regulation (e.g. a transcriptional activator that controls a repressor), combinatorial effects, and promoter sequence information. Regulators are ranked from top to bottom in order of overall predictive significance across experiments, computed by taking the larger of the normalized margin scores for up and down targets in each experiment and then averaging across experiments. The functional category for each regulator is indicated by an annotation given at the right of the figure and explained in the legend.

Several previously characterized regulators that are known to be important for oxygen and/or heme regulation, including Upc2, Rox1, Mga2, Hap4, and Hap1 [51,55,56,63,73,109,143,144], rank highly in this global regulatory map. Among the most significant regulators, six are previously known to be important for hypoxia response or oxygen regulation. Seven regulators known to be involved in cell cycle are identified by MEDUSA in this network. Intriguingly, six regulators known to be involved in pheromone response are identified (Figure 6.6). Likewise, several regulators known to regulate osmotic, salt and pseudohyphal growth are also identified. These results suggest that oxygen and heme regulation may share many regulators with pheromone and other signaling pathways.

We also predict many new regulators of oxygen and heme regulation, such as Pph3, Bem2, and Pcl1. In support of the regulation program, several identified regulators are known to interact with each other. For example, Mbp1 and Ure2 are known to coexist in one complex, and the MAP kinase kinase kinase Ssk22 acts upstream of Mbp1. Pph3 and Bem2 are known to coexist in one complex, and both likely mediate the regulation of both

hypoxically induced genes and oxygen-induced genes in $\Delta hap1$ cells (Figure 6.4A, 6.5, and 6.6). Wtm1 and Afr1, which are known to coexist in one complex, act in concert to promote oxygen regulation in wild type HAP1 cells (Figures 6.4A and 6.6). Likewise, Ire1, which acts upstream of Rgs2, may act with Rgs2 to mediate the regulation of heme-suppressed genes (Figures 6.5A and 6.6).

Our analysis does not identify the regulators that mediate general stress responses, such as Msn2, Msn4, Tpk1, Usv1, Yap1, and Hsf1 [28, 81, 111], although motifs for some of these regulators are identified in the promoters of target genes. In some aspects, anaerobic and heme-deficient conditions exhibit certain characteristics of stress responses. As such, certain genes induced by stress, such as genes involved in ribosome synthesis, are induced by anaerobic and heme-deficient conditions. However, the regulatory network mediating oxygen and heme regulation is clearly different from the general stress response network. The most significant regulators in the oxygen and heme regulatory network are not those involved in general stress responses. Interestingly, however, this oxygen and heme regulatory network shares many regulators with other signaling pathways, such as pheromone signaling and osmotic responses.

Finally, it is important to note that the number of significant regulators identified by MEDUSA is much smaller than the number of regulators whose expression is changed in a specific experiment. For example, in normoxic *HAP1* cells, we identify 18 significant regulators that may mediate the regulation of the oxygen-induced genes (Figure 6.6, first column), out of 98 regulators whose mRNA levels are significantly altered in the experiment. This dramatic filtering is achieved by two aspects of our computational approach: First, we require that regulators control their putative targets through shared motifs in the promoter sequences. Second, we train on examples from multiple experimental conditions. If a regulator cannot be associated with a binding site motif through which it contributes to target gene regulation in a consistent way across multiple conditions, it is not identified as a significant regulator by the algorithm.

6.7 Conclusion

MEDUSA’s “cluster-free” approach has the advantage that it can still be effective for small expression data sets, where clustering only generates large and functionally heterogeneous gene sets. One of the reasons for the success of methods that explicitly or implicitly rely on clustering, is their ability to segment the data and provide a visually appealing small-scale view of the regulatory network. However, as discussed in Section 5.2 forcing a modular structure on genes is not always a reasonable biological assumption for learning gene regulation. In this chapter, we show that it is not necessary to model the genome as static modules of genes to learn context-specific regulation of gene sets. Our methodology involves learning a single global model of regulation across all genes and experiments in a dataset. We then use a powerful post-processing framework to extract and visualize context-specific regulation. We are able to focus on different scales of regulation, ranging from genome-wide regulomes to regulators of a single gene.

In this study, we apply GeneClass and MEDUSA to learn regulatory programs underlying oxygen regulation and heme regulation. We identify many DNA sequence motifs important for oxygen and heme regulation (Figure 6.2). Further, experimental data from measuring *OLE1* promoter activity confirms the specific predictions made by MEDUSA (Figure 6.3). A comprehensive comparison with a traditional “cluster-first” motif discovery approach demonstrates that MEDUSA is more successful at identifying binding site motifs relevant to oxygen regulation.

Our analyses suggest a remarkable flexibility of the oxygen and heme regulatory network. Another feature of the oxygen and heme regulation network is its complexity. Although several previously known oxygen and heme regulators are confirmed to be important in oxygen and heme regulation by our analysis, many other regulators appear to play important roles in global oxygen and heme regulation. Through biochemical validation of the predicted regulators for the *OLE1* promoter, we have taken the first step in confirming the novel components of the oxygen regulatory network. While much experimental work remains

to be done, we are encouraged by our success in generating condition- and target-specific hypotheses that we validate experimentally.

Conclusion

In this chapter, we provide a summary of the main contributions in the thesis. We discuss some of the limitations of our framework and future challenges and extensions.

7.1 Summary

In this thesis, we present a predictive modeling framework, based on *boosting* algorithms for learning details of transcriptional regulation from heterogeneous sources of high-throughput genomic data. We integrate regulatory sequence data, DNA-binding data and microarray expression data into a unified model that is based on biologically meaningful assumptions.

We introduce a novel algorithm called *GeneClass* to learn gene regulation programs from gene expression data and regulatory sequence data. Specifically, we model the problem as a classification task in which we predict the up-regulation and down-regulation of genes in different sets of experimental conditions. We avoid gene clustering assumptions and instead learn a single global model of gene regulation for all genes and experiments. We develop methods to extract context-specific predictive regulators and motifs from our models. We specifically apply GeneClass to learn regulation programs relevant to environmental stress responses and DNA damage stress responses in yeast. Many of the inferred relationships are supported in the literature.

We extend GeneClass to learn *cis* regulatory motifs *ab initio* from promoter sequence data and gene expression data. In *MEDUSA*, we simultaneously learn regulation programs and regulatory sequence motifs. We search through all possible subsequences (*k*-mers) and gapped elements (dimers) in the promoter sequences of all genes to identify probabilistic motifs that allow us to predict up/down expression of target genes. We apply *MEDUSA* to various datasets of different sizes in yeast, worm and human B-cells. We learn yeast motifs whose ability to predict differential expression of target genes outperforms motifs from a compendium of known binding sites and from a previously published candidate set of learned motifs. We also show that *MEDUSA* retrieves many experimentally confirmed transcription factor binding sites.

Typically, we need to work with very high dimensional feature spaces and limited amounts of noisy training data. This makes the learning task computationally and statistically challenging. We develop efficient algorithms that use domain specific knowledge to learn prediction functions that are both accurate and qualitatively interpretable. Our models are quantitatively and qualitatively predictive. We use rigorous statistical validation procedures such as cross-validation and randomization experiments to objectively evaluate the predictive performance of our algorithms. We also show that our methods outperform other standard approaches such as nearest neighbor algorithms.

We introduce a post-processing framework to extract and display interesting biological information and generate testable hypotheses. In order to test the usefulness of our algorithms in the field, we use GeneClass and *MEDUSA* to rigorously analyze a small gene expression dataset that probes the response of yeast to the hypoxia stress response. Using our framework, we are able to learn the hypoxia regulome. We not only identify several known regulators and sequence motifs but also discover several new ones. We validate some of our hypotheses using wet-lab experiments. Thus, we show that our methods are able to decipher novel regulatory relationships even in the presence of limited amounts of noisy data.

7.2 Limitations and Future directions

In this thesis, we have shown that our algorithms are capable of learning models of gene regulation that can accurately predict gene expression. However, we make several simplifying assumptions and do not model some important aspects of gene regulation. Below, we present some of the limitations of our models and how we plan to address them in future work.

7.2.1 Representing other modes of regulation

There are several other modes of gene regulation that our current models fail to capture.

The steady state mRNA expression level of a gene, measured by microarrays, is determined by the transcription rate and the mRNA decay rate. Decay rates can be modulated by various RNA binding proteins that bind to sequence motifs in the 3'-untranslated regions of genes. GeneClass and MEDUSA do not model the regulation of decay rates.

MicroRNAs (miRNA) are single-stranded RNA molecules of about 21-23 nucleotides in length, which regulate gene expression. In animals, miRNAs bind specific complementary sites in the 3-untranslated regions of mRNAs leading to an inhibition of protein translation or facilitation of mRNA cleavage. While microRNAs are not present in yeast, they have been found to be critical for repression of gene expression in higher eukaryotes. Hence, it is critical to extend our methods to model microRNA regulation.

In higher eukaryotes, cis regulatory sequences also exist in distal regulatory regions such as enhancers and silencers. In our current models, we only account for proximal regulatory sequence (promoters). Moreover, a single gene can encode for multiple gene products (RNAs and proteins). Alternative splicing is a mechanism whereby different sections of a gene's primary transcript are separated. Some sections are spliced out and others are reconnected to produce alternative transcripts. The process of alternative splicing is carefully regulated. Our current feature space is unable to account for regulation of alternative gene

transcripts.

Another limitation of our current models is the reliance on mRNA expression levels of regulators, which might not always represent regulator activity due to post-translational modifications. This issue is at least partially addressed in two ways. First, we include signal transducers, whose mRNA expression levels (due to various feedback mechanisms) are often found by GeneClass and MEDUSA to be predictive of the targets of the transcription factors that they phosphorylate. Second, while the mRNA expression levels of transcription factors are sometimes not predictive – their mRNA expression level can be constant over a set of experiments – we still identify the binding sites for these transcription factors. Thus, a regulator's activity can be represented in two ways, through its mRNA expression level or through its binding site or the binding site of a transcription factor related to it. This dual representation helps to address another problem, which is that a transcription factor might sometimes regulate a target without binding DNA, for example by forming a complex with other DNA-binding factors. Our model can represent this situation; our only assumption is that some factor is binding the DNA, so that some binding site might identify the appropriate target genes. Ultimately, we envision protein expression data becoming available in a wide range of conditions and collected in parallel to microarray expression experiments. At that point, one could use protein expression levels of regulators to predict differential mRNA expression of targets in our models.

7.2.2 Discretization of expression data

Another current limitation is the discretization of gene expression data, which results in the loss of more subtle differences in expression. Our algorithms uses binary prediction of significantly up versus significantly down examples as a practical way of dealing with noisy expression measurements and also to leverage state-of-the-art binary prediction algorithms. We also discretize the gene expression levels of potential regulators. In several biological contexts such as developmental, subtle changes in the gene expression levels of a regulator

can cause small but significant changes in the expression of its targets. Our binary classification framework will fail to model such systems. Other learning problem formulations such as multiclass classification, ranking and regression can lead to more refined predictions.

We can extend our current framework to include these formulations. For example, we can consider a ranking problem, where we want to correctly rank two measurements if the difference in their expression levels is larger than the systemic noise, i.e. if example a is significantly more highly expressed than b , we want our prediction function f to correctly rank $f(a) > f(b)$. Fortunately, there has been considerable recent work on learning rankings in the machine learning literature (see e.g. [98] for an approach related to boosting). Similarly, handling real-valued regulator expression levels and using boosting-like algorithms for regression involve standard extensions to our learning algorithm [59, 101]. The primary question to investigate is whether the high level of noise in mRNA expression data allows us to effectively learn these more detailed output functions.

7.2.3 Integration of other high-throughput data sources

An important extension to our methods is to tie the predictive modeling approach more tightly to the underlying biological mechanisms of activated signaling pathways and binding interactions between regulators (which may act as a complex of proteins) and regulatory DNA. While this mechanistic information is represented indirectly in the features (and therefore in the regulatory programs) learned by MEDUSA, we often saw pieces of this underlying biology — for example, sometimes all components of a regulatory protein complex were associated by MEDUSA with the same transcription factor occupancy feature, or all components of a known signaling pathway were retrieved as regulators for a particular set of target genes [62]. We can better represent these interactions by incorporating new high-throughput data sources.

In the last year, the high throughput determination of kinase-substrate interactions [75] has provided a new and highly relevant source of interaction data. So far, this data has

established a partial “kinome” network for yeast, mapping out all potential phosphorylation reactions for a large proportion of kinases. We anticipate that kinome data will soon be available for other model organisms as well. By incorporating kinome network data into the regulatory program learning algorithm, we can hope to discover signaling pathways and their context-specific activation. In addition, we can represent protein-protein interaction data from high throughput assays like yeast two-hybrid (e.g. [38] as well as interaction databases (e.g. MIPS, KEGG) in order to infer protein complexes that act as regulators in a regulatory program.

In order to represent the interaction network structure in our learning algorithm, we can replace the expression levels of single regulators with that of subgraphs from the interaction network. To do this, one could use efficient graph mining algorithms [16, 132] to explore the exponential search space of subgraphs to find the ones that are predictive. At each boosting round, we can look for the most predictive kinome or interactome subnetwork and the transcription factor occupancy or sequence motif through which it regulates targets relative to the current weighting of the training data.

High throughput protein-protein interaction datasets tend to be very noisy. Our boosting approach will be robust to false positives in the interaction data, since including these edges in subgraphs will not improve prediction of target gene regulation. In order to handle missing edges we plan to use two strategies. First, we plan an interolog strategy [135], where known interactions from one organism are transferred via sequence homology to infer a putative interaction between the orthologous proteins in another organism. Second, we plan a random sampling strategy, where we randomly add edges between regulators with correlated expression signatures to the base network, and we see if any of these edges contribute optimally predictive subgraphs.

7.2.4 Motif discovery in higher eukaryotes: cis regulatory modules

A key issue in developing more realistic gene regulation models and scaling up these models to higher organisms such as mammals is the representation of greater complexity in the regulatory DNA. In the MEDUSA algorithm, we discover putative binding site patterns that can individually help to predict differential regulation, when paired with the activity level of a particular regulator. However, in various contexts in higher eukaryotes, binding of trans acting factors appears to be less specific, and regions containing spatial clusters of binding sites, called cis regulatory modules (CRMs), are considered to be the irreducible functional elements [13, 95]. MEDUSA does not explicitly model motif combinatorics or CRMs.

We could extend our predictive modeling approach to model and learn CRMs directly from regulatory sequences, without requiring that individual binding site patterns be known ahead of time.

We consider two cases: homotypic and heterotypic CRMs. Homotypic CRMs consist of spatial regions in the regulatory DNA with multiple hits for the transcription factor; heterotypic CRMs are clusters of binding sites for different transcription factors. We anticipate that homotypic CRMs can be learned using a similar boosting and sequence agglomeration technique that MEDUSA uses: first we find all k -mers with multiple (approximate) hits within some window in a promoter sequence, and then we agglomerate predictive k -mers to learn PSSMs that define homotypic CRMs.

For heterotypic CRMs, we can leverage state-of-the-art pattern mining techniques from the data mining community within our boosting framework. We can start by finding seed k -mers or PSSMs using the same strategy as in the homotypic case. Then we can mine for patterns of motif combinations that are predictive of target expression, when associated with the corresponding set of regulators. We have many options for the type of pattern mining, reflecting different possible statistical definitions of a CRM: itemset mining [123, 136], where we represent a regulatory sequence region by the sets of motifs it contains and look for predictive motif subsets; sequential pattern mining [133], where we view regulatory DNA

as sequences of motif hits and find predictive subsequence patterns; graph mining [16, 132], where we define a graph on the motif set for a sequence, joining two motif instances with an edge if they are proximal (a directed edge if we also want a partial order), and mine predictive subgraphs, i.e. proximal (partially ordered) motif hits.

In most cases, computational approaches to CRM discovery use sequence only and rely on a database of known motifs [91, 108]; in cases where motifs as well as CRMs are returned, usually only a small set of closely related genes is modeled [95, 141]. In our planned approach, we will not need to start with known motifs (though these can be added if available), and we learn CRMs across all differentially expressed genes by integrating efficient pattern mining algorithms with boosting to learn regulatory programs. We believe that this algorithmic approach has not been tried before and would constitute an innovative contribution to the field; if successful in our goals, we would also advance the current state of knowledge of metazoan gene regulation.

7.2.5 Comparative genomic approaches for learning conserved regulation

A common approach to searching for *cis* acting sequences that control transcriptional regulation — both at the level of individual transcription factor binding sites and CRMs — is to compare genome sequences of related species and look for conserved regions in the non-coding sequence flanked by regions of lower conservation (reviewed in e.g. [12]). While these conservation-based approaches are often quite simple, they have been successful in identifying *cis* regulatory elements in yeast [60] and regulatory sequences in the fly *Drosophila melanogaster* [11] and in mammalian genomes [43, 88, 130]. Other efforts have been made to combine information about conserved non-coding sequence with conserved expression patterns of target genes to locate functional sequence [12].

This prior work suggests that a comparative strategy in the our framework should improve motif and CRM discovery for higher eukaryotes. In the current implementation of

MEDUSA, k -mer occurrences in the promoter sequences of target genes serve as seeds for an agglomerative hierarchical clustering algorithm that is used, at each round of boosting, to produce candidate motifs in the form of PSSMs. If aligned promoter sequences from one or more related species are available, we can restrict these seed k -mers to those found in conserved regions in the original genome; to avoid using an alignment, we can simply use k -mer occurrence from all the orthologous sequences as seeds and require that the learned motifs have hits across multiple genomes. These simple strategies extend naturally to the case of learning CRMs.

7.2.6 Use of epigenomic data to improve identification of cis regulatory elements

A major problem in discovering cis regulatory elements in higher eukaryotes is the length of regulatory sequences such as promoters. In yeast, the average promoter stretches approximately 1000 base pairs upstream of a gene's transcription start site (TSS). In sharp contrast, human promoter sequences are estimated to have regulatory signals up to 10000 base pairs away from the TSS. Just this 10 fold increase in search space alone leads to serious statistical and computational problems for *ab initio* motif discovery. The problem is further exacerbated by the importance of potentially position-independent unidentified enhancer and silencer elements that can exist several 10000s of base pairs away from the genes they regulate. We plan to use epigenomic information such as information about nucleosome positioning and histone modifications to filter and reweight various sequence chunks in order to amplify regulatory signals in sequence data.

The dynamics of chromatin structure are tightly regulated through multiple mechanisms including histone modification, chromatin remodeling, histone variant incorporation, and histone eviction. The position of nucleosomes is critical in regulating access of transcriptional activators, repressors and RNA polymerase to naked DNA. Recent high throughput experiments [107] and subsequent computational analysis have identified weak sequence

specific signals in genomic sequences that are correlated with nucleosome positioning. We plan to use this data to reweight various sequence k -mers based on their positional overlap with observed and predicted nucleosome occupancy patterns.

Regulatory proteins tend to bind specific regions in the genome such as promoters and enhancers and recent ChIP chip studies of histone modifications in the ENCODE regions [45] of the human genome show the presence of strong discriminative histone-modification patterns that characterize promoters, enhancers and other regulatory sequence chunks. We plan to use this data to narrow down the search space for regulatory motifs.

Other recent studies [14, 26] have revealed large scale tissue-specific DNA methylation patterns distinguishing active euchromatin from silenced heterochromatin. This data can also be used to distinguish between context-specific active and inactive regulatory sequences.

7.3 End note

Building computational models of biological systems is both an art and a science. The art of modeling involves identifying the problem, using domain knowledge to make key modeling assumptions, choosing an appropriate modeling approach, and developing innovative methods to extract relevant information from the models. The science of modeling involves converting domain knowledge both quantitative and qualitative into efficient and flexible mathematical abstractions.

As Samuel Karlin [57] puts it: “The purpose of models is not to fit the data but to sharpen the questions.” Computational biology is a strange marriage of two fields that employ very different philosophies of exploration. The computational field tends to be more theoretical while biology is a more empirical science. It is thus important to have constant communication across the border in order to build useful biological models. The ultimate utility of a computational approach, no matter how elegant and theoretically sound it might be, is in its ability to contribute something new to the field of biology. It is a challenge

to build models that are both statistically reliable and empirically useful. Overly complex methods run the risk of overfitting to data. Other algorithms might learn from data. But, the models produced might be irrelevant and provide meaningless hypotheses. Similarly, a model is relatively useless if it is statistically sound but unable to provide any interesting insight into the underlying mechanism. Our hope is that in this thesis, we have shown that it is possible to effectively integrate heterogeneous sources of high-throughput data to build accurate, predictive models that can provide interesting insights into regulatory biology. We hope to extend our framework and develop new algorithms capable of handling the massive amounts and different types of high-throughput data that will be available in the near future.

Bibliography

- [1] The gene ontology (go) project in 2006. *Nucleic Acids Res*, 34(Database issue):D322–6, 2006. 1362–4962 (Electronic) Journal Article.
- [2] Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature*, 447:799–816, June 2007.
- [3] Mustapha Aouida, Nicolas Page, Anick Leduc, Matthias Peter, and Dindial Ramotar. A Genome-Wide Screen in *Saccharomyces cerevisiae* Reveals Altered Transport As a Mechanism of Resistance to the Anticancer Drug Bleomycin. *Cancer Res*, 64(3):1102–1109, 2004.
- [4] F. M. Ausubel, R. Brent, R. E. Kingston, D. D. Moore, J. G. Seidman, J. A. Smith, and K. Struhl. *Current protocols in molecular biology*. John Wiley & Sons, Inc., 2000.
- [5] T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol.*, 2:28–36., 1994.
- [6] Katia Basso, Adam A Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human B cells. *Nat Genet*, 37(4):382–390, Apr 2005. Comparative Study.
- [7] A.J. Battle, E. Segal, and D. Koller. Probabilistic discovery of overlapping cell processes and their regulation. In *Eight Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, San Diego, CA, April 2004.
- [8] L. Ryan Baugh, Andrew A. Hill, Donna K. Slonim, Eugene L. Brown, and Craig P. Hunter. Composition and dynamics of the *Caenorhabditis elegans* early embryonic transcriptome. *Development*, 130(5):889–900, 2003.
- [9] L. Ryan Baugh, Andrew A. Hill, Donna K. Slonim, Eugene L. Brown, and Craig P. Hunter. Composition and dynamics of the *Caenorhabditis elegans* early embryonic transcriptome. *Development*, 130(5):889–900, 2003.
- [10] M. A. Beer and S. Tavazoie. Predicting gene expression from sequence. *Cell*, 117(2):185–98., Apr 16 2004.
- [11] CM Bergman, BD Pfeiffer, DE Rincon-Limas, RA Hoskins, A Gnirke, C Mungall, A Wang, B Krommiller, J Pacleb, and S Park. Assessing the impact of comparative genomic sequence data on the functional annotation of the drosophila genome. *Genome Biol*, 3:research0086.1–0086.20, 2002.
- [12] Benjamin Berman, Barret Pfeiffer, Todd Laverty, Steven Salzberg, Gerald Rubin, Michael Eisen, and Susan Celniker. Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in *drosophila melanogaster* and *drosophila pseudoobscura*. *Genome Biology*, 5(9):R61, 2004.

- [13] B.P. Berman, Y. Nibu, B.D. Pfeiffer, P. Tomancak, S.E. Celniker, M. Levine, G.M. Rubin, and M.B. Eisen. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the drosophila genome. *Proceedings of the National Academy of Sciences*, 99:757–762, 2002.
- [14] Bradley E. Bernstein, Alexander Meissner, and Eric S. Lander. The mammalian epigenome. *Cell*, 128(4):669–681, February 2007.
- [15] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–93, 2003. 1367-4803 (Print) Evaluation Studies Journal Article Validation Studies.
- [16] Christian Borgelt. On canonical forms for frequent graph mining. In *3rd Int. Workshop on Mining Graphs, Trees and Sequences (MGTS'05, Porto, Portugal)*, pages 1–12, Porto, Portugal, 2005. ECML/PKDD 2005 Organization Committee.
- [17] Martha L. Bulyk, Philip L. F. Johnson, and George M. Church. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucl. Acids Res.*, 30(5):1255–1261, 2002.
- [18] H. F. Bunn and R. O. Poyton. Oxygen sensing and molecular adaptation to hypoxia. *Physiol Rev*, 76(3):839–85, 1996.
- [19] P. V. Burke, D. C. Raitt, L. A. Allen, E. A. Kellogg, and R. O. Poyton. Effects of oxygen concentration on the expression of cytochrome c and cytochrome c oxidase genes in yeast. *J Biol Chem*, 272(23):14705–12, 1997. 0021-9258 Journal Article.
- [20] H. J. Bussemaker, H. Li, and E. D. Siggia. Regulatory element detection using correlation with expression. *Nature Genetics*, 27:167–171, 2001.
- [21] J. Y. Choi, J. Stukey, S. Y. Hwang, and C. E. Martin. Regulatory elements that control transcription activation and unsaturated fatty acid-mediated repression of the *saccharomyces cerevisiae ole1* gene. *J Biol Chem*, 271(7):3581–9, 1996. 0021-9258 Journal Article.
- [22] M. C. Costanzo, J. D. Hogan, M. E. Cusick, B. P. Davis, A. M. Fancher, P. E. Hodges, P. Kondu, C. Lengieza, J. E. Lew-Smith, C. Lingner, K. J. Roberg-Perez, M. Tillberg, J. E. Brooks, and J. I. Garrels. The yeast proteome database (YPD) and *Caenorhabditis elegans* proteome database (WormPD): comprehensive resources for the organization and comparison of model organism protein information. *Nucleic Acids Res*, 28(1):73–76, 2000.
- [23] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1990.
- [24] Anna Dahlkvist and Per Sunnerhagen. Two novel deduced serine/threonine protein kinases from *saccharomyces cerevisiae*. *Gene*, 139:27–33, February 1994.
- [25] P. D’Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing*, 1999.
- [26] Florian Eckhardt, Joern Lewin, Rene Cortese, Vardhman K Rakyan, John Attwood, Matthias Burger, John Burton, Tony V Cox, Rob Davies, Thomas A Down, Carolina Haefliger, Roger Horton, Kevin Howe, David K Jackson, Jan Kunde, Christoph Koenig, Jennifer Liddle, David Niblett, Thomas Otto, Roger Pettett, Stefanie Seemann, Christian Thompson, Tony West, Jane Rogers, Alex Olek, Kurt Berlin, and Stephan Beck. Dna methylation profiling of human chromosomes 6, 20 and 22. *Nat Genet*, 38(12):1378–1385, December 2006.
- [27] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Nat. Acad. Sci. USA*, 95:14863–8, 1998.
- [28] A. M. Erkin, S. F. Magrogan, E. A. Sekinger, and D. S. Gross. Cooperative binding of heat shock factor to the yeast hsp82 promoter in vivo and in vitro. *Mol Cell Biol*, 19(3):1627–39, 1999. 0270-7306 (Print) Journal Article Research Support, Non-U.S. Gov’t Research Support, U.S. Gov’t, P.H.S.

- [29] Jason Ernst, Oded Vainas, Christopher T Harbison, Itamar Simon, and Ziv Bar-Joseph. Reconstructing dynamic regulatory maps. *Mol Syst Biol*, 3:74, 2007.
- [30] T. S. Spellman et al. Comprehensive identification of cell cycle-related genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. of the Cell*, 9:3273–3297, 1998.
- [31] Y. Freund and L. Mason. The alternating decision tree learning algorithm. *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 124–133, 1999.
- [32] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.
- [33] N Friedman, M Linial, I Nachman, and D Pe’er. Using Bayesian networks to analyze expression data. *J Comput Biol*, 7(3-4):601–620, 2000.
- [34] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805, 2004.
- [35] Itay Furman and Yitzhak Pilpel. Promoting human promoters. *Mol Syst Biol*, 2:2006.0030, 2006. Comment.
- [36] A. P. Gasch, M. Huang, S. Metzner, D. Botstein, S. J. Elledge, and P. O. Brown. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol Biol Cell*, 12(10):2987–3003., Oct 2001.
- [37] A. P. Gasch, P. T. Spellman, C. M. Kao, Orna Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, 11:4241–4257, 2000.
- [38] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. M. Michon, C. M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature.*, 415(6868):141–7., Jan 10 2002.
- [39] David Gifford. Blazing pathways through genetic mountains. *Science*, 293:2049–2051, 2001.
- [40] A.V. Grishin, M. Rothenberg, M.A. Downs, and K.J. Blumer. Mot3, a zn finger transcription factor that modulates gene expression and attenuates mating pheromone signaling in *saccharomyces cerevisiae*. *Genetics*, 149:879–92, 1998.
- [41] L. Guarente, B. Lalonde, P. Gifford, and E. Alani. Distinctly regulated tandem upstream activation sites mediate catabolite repression of the *cyc1* gene of *s. cerevisiae*. *Cell*, 36(2):503–11, 1984.
- [42] C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J. B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature.*, 431(7004):99–104., Sep 2 2004.
- [43] RC Hardison. Comparative genomics. *PLoS Biol*, 1:E58, 2003.
- [44] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing*, 2001.
- [45] Nathaniel D Heintzman, Rhona K Stuart, Gary Hon, Yutao Fu, Christina W Ching, R David Hawkins, Leah O Barrera, Sara Van Calcar, Chunxu Qu, Keith A Ching, Wei Wang, Zhiping Weng, Roland D Green, Gregory E Crawford, and Bing Ren. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat Genet*, 39(3):311–318, March 2007.
- [46] G. Z. Hertz and G. D. Stormo. Identifying and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–77., Jul-Aug 1999.

- [47] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
- [48] Jorg D. Hoheisel. Microarray technology: beyond transcript profiling and genotype analysis. *Nat Rev Genet*, 7(3):200–210, March 2006.
- [49] Carina I. Holmberg, Ville Hietakangas, Andrey Mikhailov, Jouni O. Rantanen, Marko Kallio, Annika Meinander, Jukka Hellman, Nick Morrice, Carol MacKintosh, Richard I. Morimoto, John E. Eriksson, and Lea Sistonen. Phosphorylation of serine 230 promotes inducible transcriptional activity of heat shock factor 1. *EMBO J.*, 20(14):3800–3810, 2001.
- [50] T. Hon, A. Dodd, R. Dirmeier, N. Gorman, P. R. Sinclair, L.* Zhang, and R.O. Poyton. A mechanism of oxygen sensing in yeast: Multiple oxygen-responsive steps in the heme biosynthetic pathway affect hap1 activity. *J. Biol. Chem.*, 278:50771–80, 2003.
- [51] T. Hoppe, K. Matuschewski, M. Rape, S. Schlenker, H. D. Ulrich, and S. Jentsch. Activation of a membrane-bound transcription factor by regulated ubiquitin/proteasome-dependent processing. *Cell.*, 102(5):577–86., Sep 1 2000.
- [52] M. Huang and S. J. Elledge. Identification of RNR4, encoding a second essential small subunit of ribonucleotide reductase in *Saccharomyces cerevisiae*. *Molecular Cell Biology*, 17:6105–6113, 1997.
- [53] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *Journal of Molecular Biology*, 296(5):1205–14, 2000.
- [54] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of affymetrix genechip probe level data. *Nucleic Acids Res*, 31(4):e15, 2003. 1362-4962 (Electronic) Journal Article.
- [55] Y. Jiang, M. J. Vasconcelles, S. Wretzel, A. Light, L. Gilooly, K. McDaid, C. S. Oh, C. E. Martin, and M. A. Goldberg. Mga2p processing by hypoxia and unsaturated fatty acids in *saccharomyces cerevisiae*: impact on ore-dependent gene expression. *Eukaryot Cell*, 1(3):481–90, 2002. 1535-9778 Journal Article.
- [56] Y. Jiang, M. J. Vasconcelles, S. Wretzel, A. Light, C. E. Martin, and M. A. Goldberg. Mga2 is involved in the low-oxygen response element-dependent hypoxic induction of genes in *saccharomyces cerevisiae*. *Mol Cell Biol*, 21:6161–9, 2001.
- [57] S Karlin. 11th r a fisher memorial lecture, royal society 20, April 1983.
- [58] N. Kawamata, T. Miki, K. Ohashi, K. Suzuki, T. Fukuda, S. Hirose, and N. Aoki. Recognition DNA Sequence of a Novel Putative Transcription Factor, BCL6. *Biochemical and Biophysical Research Communications*, 204(1):366–374, 15 October, 1994.
- [59] Balázs Kégl. Robust regression by boosting the median. In *COLT*, pages 258–272, 2003.
- [60] M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E. S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–54., May 15 2003.
- [61] A. Kundaje, C. Lan, M. Zhou, C. Leslie, and L. Zhang. A predictive model of the oxygen sensing and regulatory network in yeast. Submitted for publication.
- [62] A. Kundaje, M. Middendorf, M. Shah, C. H. Wiggins, Y. Freund, and C. Leslie. classification-based framework for predicting and analyzing gene regulatory response. *BMC Bioinformatics.*, 7 Suppl 1:S1–5., Mar 20 2006.
- [63] K. E. Kwast, L. C. Lai, N. Menda, D. T. 3rd James, S. Aref, and P. V. Burke. Genomic analyses of anaerobically induced genes in *saccharomyces cerevisiae*: functional roles of rox1 and other factors in mediating the anoxic response. *J Bacteriol*, 184:250–65, 2002.
- [64] S. Labb, Z. Zhu, and D. J. Thiele. Copper-specific transcriptional repression of yeast genes encoding critical components in the copper transport pathway. *The Journal of biological chemistry*, 272(25):15951–8, 1997. 00219258.

- [65] Liang-Chuan Lai, Alexander L. Kosorukoff, Patricia V. Burke, and Kurt E. Kwast. Dynamical remodeling of the transcriptome during short-term anaerobiosis in *saccharomyces cerevisiae*: Differential response and role of *msn2* and/or *msn4* and other factors in galactose and glucose media. *Mol. Cell. Biol.*, 25(10):4075–4091, 2005. 10.1128/MCB.25.10.4075-4091.2005.
- [66] Liang-Chuan Lai, Alexander L. Kosorukoff, Patricia V. Burke, and Kurt E. Kwast. Metabolic-state-dependent remodeling of the transcriptome in response to anoxia and subsequent reoxygenation in *saccharomyces cerevisiae*. *Eukaryotic Cell*, 5(9), 2006. 10.1128/EC.00107-06.
- [67] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.
- [68] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. R. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298:799–804, 2002.
- [69] Tong Ihn Lee, Nicola J. Rinaldi, Francois Robert, Duncan T. Odom, Ziv Bar-Joseph, Georg K. Gerber, Nancy M. Hannett, Christopher T. Harbison, Craig M. Thompson, Itamar Simon, Julia Zeitlinger, Ezra G. Jennings, Heather L. Murray, D. Benjamin Gordon, Bing Ren, John J. Wyrick, Jean-Bosco Tagne, Thomas L. Volkert, Ernest Fraenkel, David K. Gifford, and Richard A. Young. Transcriptional Regulatory Networks in *Saccharomyces cerevisiae*. *Science*, 298(5594):799–804, 2002.
- [70] Yang David Lee and Stephen J. Elledge. Control of ribonucleotide reductase localization through an anchoring mechanism involving Wtm1. *Genes Dev.*, 20(3):334–344, 2006.
- [71] Qiutang Li and Inder M. Verma. Nf-[kappa]b regulation in the immune system. *Nature Reviews Immunology*, 2(10):725–734, 2002.
- [72] Janine T. Lin and John T. Lis. Glycogen Synthase Phosphatase Interacts with Heat Shock Factor To Activate CUP1 Gene Transcription in *Saccharomyces cerevisiae*. *Mol. Cell. Biol.*, 19(5):3237–3245, 1999.
- [73] C. V. Lowry and R. S. Zitomer. Rox1 encodes a heme-induced repression factor regulating *anb1* and *cyc7* of *saccharomyces cerevisiae*. *Mol Cell Biol*, 8(11):4651–8, 1988.
- [74] K. D. MacIsaac, T. Wang, D. B. Gordon, D. K. Gifford, G. D. Stormo, and E. Fraenkel. An improved map of conserved regulatory sites for *saccharomyces cerevisiae*. *BMC Bioinformatics*, 7:113, 2006.
- [75] A. S. Mah, A. E. Elia, G. Devgan, J. Ptacek, M. Schutkowski, M. Snyder, M. B. Yaffe, and R. J. Deshaies. Substrate specificity analysis of protein kinase complex dbf2-mob1 by peptide library and proteome array screening. *BMC Biochem.*, 6:22., Oct 21 2005.
- [76] Tsz-Kwong Man and Gary D. Stormo. Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucl. Acids Res.*, 29(12):2471–2478, 2001.
- [77] V. Matys, O. V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. E. Kel, and E. Wingender. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res*, 34(Database issue):D108–110, 2006.
- [78] M. Middendorf, A. Kundaje, M. Shah, Y. Freund, C. Wiggins, and C. Leslie. Motif discovery through predictive modeling of gene regulation. *Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2005)*, 2005.
- [79] M. Middendorf, A. Kundaje, C. Wiggins, Y. Freund, and C. Leslie. Predicting genetic regulatory response using classification. *Proceedings of the Twelfth International Conference on Intelligent Systems for Molecular Biology (ISMB 2004)*, 2004.

- [80] M. Middendorff, A. Kundaje, C. Wiggins, Y. Freund, and C. Leslie. Predicting genetic regulatory response using classification: Yeast stress response. *Proceedings of the First Annual RECOMB Regulation Workshop*, 2005.
- [81] K. A. Morano, N. Santoro, K. A. Koch, and D. J. Thiele. A trans-activation domain in yeast heat shock transcription factor is essential for cell cycle progression during stress. *Mol Cell Biol*, 19(1):402–11, 1999. 0270-7306 (Print) Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. Research Support, U.S. Gov't, P.H.S.
- [82] IM Ota and A Varshavsky. A Gene Encoding a Putative Tyrosine Phosphatase Suppresses Lethality of an N-End Rule-Dependent Mutant. *PNAS*, 89(6):2355–2359, 1992.
- [83] L. Pasqualucci, O. Bereschenko, H. Niu, U. Klein, K. Basso, R. Guglielmino, G. Cattoretti, and R. Dalla-Favera. Molecular pathogenesis of non-Hodgkin's lymphoma: the role of Bcl-6. *Leuk Lymphoma*, 44(Suppl 3):S5–12, 2003.
- [84] A. G. Paulovich and L.H. Hartwell. A checkpoint regulates the rate of progression through S phase in *S. cerevisiae* in response to DNA damage. *Cell*, 82:841–847, 1995.
- [85] Ivo Pedruzzi, Niels Burckert, Pascal Egger, and Claudio De Virgilio. *Saccharomyces cerevisiae* Ras/cAMP pathway controls post-diauxic shift element-dependent transcription through the zinc finger protein Gis1. *EMBO J.*, 19(11):2569–2579, 2000.
- [86] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Proceedings of the Ninth International Conference on Intelligent Systems for Molecular Biology*, 2001.
- [87] D. Pe'er, V. Regev, and A. Tanay. A fast and robust method to infer and characterize an active regulator set for molecular pathways. *Proceedings of the Tenth International Conference on Intelligent Systems for Molecular Biology*, 2002.
- [88] LA Pennacchio and EM Rubin. Genomic strategies to identify mammalian regulatory sequences. *Nat Rev Genet*, 2:100–109, 2001.
- [89] K. Pfeifer, K. S. Kim, S. Kogan, and L. Guarente. Functional dissection and sequence of yeast hap1 activator. *Cell*, 56(2):291–301, 1989.
- [90] Ryan T Phan, Masumichi Saito, Katia Basso, Huifeng Niu, and Riccardo Dalla-Favera. Bcl6 interacts with the transcription factor miz-1 to suppress the cyclin-dependent kinase inhibitor p21 and cell cycle arrest in germinal center b cells. *Nat Immunol*, 6(10):1054–1060, October 2005.
- [91] A. A. Philippakis, F. S. He, and M. L. Bulyk. Modulefinder: a tool for computational discovery of cis regulatory modules. *Pac Symp Biocomput.*, pages 519–30., 2005.
- [92] Y. Pilpel, P. Sudarsanam, and G. M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, 2:153–159, 2001.
- [93] M. D. Piper, P. Daran-Lapujade, C. Bro, B. Regenberg, S. Knudsen, J. Nielsen, and J. T. Pronk. Reproducibility of oligonucleotide microarray transcriptome analyses. an interlaboratory comparison using chemostat cultures of *saccharomyces cerevisiae*. *J Biol Chem*, 277(40):37001–8, 2002. 0021-9258 (Print) Comparative Study Journal Article Research Support, Non-U.S. Gov't.
- [94] D. C. Raitt, A. L. Johnson, A. M. Erkin, K. Makino, B. Morgan, D. S. Gross, and L. H. Johnston. 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.*, 11(7):2335–2347, 2000.
- [95] N. Rajewsky, M. Vergassola, U. Gaul, and E. D. Siggia. Computational detection of genomic cis-regulatory modules applied to body patterning in the early drosophila embryo. *BMC Bioinformatics.*, 3:30. Epub 2002 Oct 24., Oct 24 2002.
- [96] E. Ramil, C. Agrimonti, E. Shechter, M. Gervais, and B. Guiard. Regulation of the *cyb2* gene expression: transcriptional co-ordination by the hap1p, hap2/3/4/5p and adr1p transcription factors. *Mol Microbiol*, 37(5):1116–32., Sep 2000.

- [97] M. Rebhan, V. Chalifa-Caspi, J. Prilusky, and D. Lancet. GeneCards: encyclopedia for genes, proteins and diseases. Weizmann Institute of Science, Bioinformatics Unit and Genome Center (Rehovot, Israel), 1997.
- [98] Cynthia Rudin, Corinna Cortes, Mehryar Mohri, and Robert E. Schapire. Margin-based ranking meets boosting in the middle. In *COLT*, pages 63–78, 2005.
- [99] Y. Sanchez, B. A. Desany, W. J. Jones, Q. Liu, B. Wang, and S.J. Elledge. Regulation of RAD53 by the ATM-like kinases MEC1 and TEL1 in yeast cell cycle checkpoint pathways. *Science*, 271:357–360, 1996.
- [100] R. E. Schapire. Theoretical views of boosting and applications. In *Tenth International Conference on Algorithmic Learning Theory*, 1999.
- [101] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
- [102] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.
- [103] T D Schneider and R M Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18, October 1990. PMC332411.
- [104] E. Segal, M. Shapira, A. Regev, D. Pe’er, D. Botstein, D. Koller, and N. Friedman. Module networks: Identifying regulatory modules and their condition specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, 2003.
- [105] E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, 19, 2003.
- [106] E Segal, R Yelensky, and D Koller. Genome-wide discovery of transcriptional modules from dna sequence and gene expression. *Bioinformatics (Oxford, England)*, 19 Suppl 1:i273–82, 2003. PMID: 12855470.
- [107] Eran Segal, Yvonne Fondufe-Mittendorf, Lingyi Chen, AnnChristine Thastrom, Yair Field, Irene K. Moore, Ji-Ping Z. Wang, and Jonathan Widom. A genomic code for nucleosome positioning. *Nature*, 442(7104):772–778, 2006.
- [108] Roded Sharan, Ivan Ovcharenko, Asa Ben-Hur, and Richard M. Karp. CREME: a framework for identifying cis-regulatory modules in human-mouse conserved segments. *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology*, 2003.
- [109] K. V. Shianna, W. D. Dotson, S. Tove, and L. W. Parks. Identification of a upc2 homolog in *saccharomyces cerevisiae* and its involvement in aerobic sterol uptake. *J Bacteriol*, 183:830–4, 2001.
- [110] Noam Slonim, Nir Friedman, and Naftali Tishby. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 129–136. ACM Press, 2002.
- [111] A. Smith, M. P. Ward, and S. Garrett. Yeast pka represses msn2p/msn4p-dependent gene expression to regulate growth, stress response and glycogen accumulation. *Embo J*, 17(13):3556–64, 1998. 0261-4189 (Print) Journal Article Research Support, U.S. Gov’t, P.H.S.
- [112] V. A. Smith, E. D. Jarvis, and A. J. Hartemink. Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics.*, 18 Suppl 1:S216–24., 2002.
- [113] Ariel Stanhill, Naomi Schick, and David Engelberg. The Yeast Ras/Cyclic AMP Pathway Induces Invasive Growth by Suppressing the Cellular Stress Response. *Mol. Cell. Biol.*, 19(11):7529–7538, 1999.
- [114] V. V. Svetlov and T. G. Cooper. Review: compilation and characteristics of dedicated transcription factors in *saccharomyces cerevisiae*. *Yeast (Chichester, England)*, 11(15):1439–84, 1995. 0749503.

- [115] S. L. Tai, V. M. Boer, P. Daran-Lapujade, M. C. Walsh, J. H. de Winde, J. M. Daran, and J. T. Pronk. Two-dimensional transcriptome analysis in chemostat cultures. combinatorial effects of oxygen availability and macronutrient limitation in *saccharomyces cerevisiae*. *J Biol Chem*, 280(1):437–47, 2005. 0021-9258 (Print) Journal Article Research Support, Non-U.S. Gov't.
- [116] Amos Tanay, Roded Sharan, and Ron Shamir. *Handbook of Computational Molecular Biology*, chapter Biclustering Algorithms: A Survey, pages 26–1–26–17. Chapman and Hall/CRC Press, 2006.
- [117] S Tavazoie, J D Hughes, M J Campbell, R J Cho, and G M Church. Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–285, Jul 1999.
- [118] J. J. ter Linde, H. Liang, R. W. Davis, H. Y. Steensma, J. P. van Dijken, and J. T. Pronk. Genome-wide transcriptional analysis of aerobic and anaerobic chemostat cultures of *saccharomyces cerevisiae*. *J Bacteriol*, 181(24):7409–13, 1999. 0021-9193 Journal Article.
- [119] Y. Tu, G. Stolovitzky, and U. Klein. Quantitative noise analysis for gene expression microarray experiments. *Proceedings of the National Academy of Sciences*, 99(22):14031–14036, 2002.
- [120] V Vapnik. *The Nature of Statistical Learning Theory*. 1995.
- [121] Michael J. Vasconcelles, Yide Jiang, Kevin McDaid, Laura Gilooly, Sharon Wretzel, David L. Porter, Charles E. Martin, and Mark A. Goldberg. Identification and characterization of a low oxygen response element involved in the hypoxic induction of a family of *saccharomyces cerevisiae* genes. implications for the conservation of oxygen sensing in eukaryotes. *J. Biol. Chem.*, 276(17):14374–14384, 2001.
- [122] Lindsey Walsh, Jacqueline Schmuckli-Maurer, Nicholas Billinton, Gordon Barker, Wolf-Dietrich Heyer, and Walmsley Richard. DNA-damage induction of RAD54 can be regulated independently of the RAD9- and DDC1-dependent checkpoints that regulate RNR2. *Current Genetics*, 41(4):232–240, 2002.
- [123] Jianyong Wang, Jiawei Han, and Jian Pei. Closed constrained gradient mining in retail databases. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):764–769, 2006.
- [124] J. C. Way and M. Chalfie. *mec-3*, a homeobox-containing gene that specifies differentiation of the touch receptor neurons in *C. elegans*. *Cell*, 54(1):5–16, 1988.
- [125] T. A. Weinert, G. L. Kiser, and L. H. Hartwell. Mitotic checkpoint genes in budding yeast and the dependence of mitosis on dna replication and repair. *Genes and Development*, 8(6):652–665, 1994.
- [126] Wikipedia. Chromatin immunoprecipitation — wikipedia, the free encyclopedia. 2007. [Online; accessed 17-April-2007].
- [127] Wikipedia. Dna microarray — wikipedia, the free encyclopedia. 2007. [Online; accessed 17-April-2007].
- [128] E. Wingender, X. Chen, R. Hehl, H. Karas, I. Liebich, V. Matys, T. . Meinhardt, M. Prüss, I. Reuter, and F. Schacherer. TRANSFAC: an integrated system for gene expression regulation. *Nucleic Acids Research*, 28:316–319, 2000.
- [129] C. T. Workman, H. C. Mak, S. McCuine, J.-B. Tagne, M. Agarwal, O. Ozier, T. J. Begley, L. D. Samson, and Ideker T. A systems approach to mapping dna damage response pathways. *Science*, 312(5776):1054–1059, 2006.
- [130] X. Xie, J. Lu, E. J. Kulbokas, T. R. Golub, V. Mootha, K. Lindblad-Toh, E. S. Lander, and M. Kellis. Systematic discovery of regulatory motifs in human promoters and 3' utrs by comparison of several mammals. *Nature*, 434(7031):338–45. Epub 2005 Feb 27., Mar 17 2005.
- [131] D. Xue, M. Finney, G. Ruvkun, and M. Chalfie. Regulation of the *mec-3* gene by the *C.elegans* homeoproteins UNC-86 and MEC-3. *EMBO J*, 11(13):4969–4969, 1992.
- [132] X. Yan and J. Han. gspan: Graph-based substructure pattern mining, 2002.
- [133] Xifeng Yan, Jiawei Han, and Ramin Afshar. Closan: Mining closed sequential patterns in large databases. In *SDM*, 2003.

- [134] M. K. Yeung, J. Tegner, and J. J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Science*, 99:6163–8, 2002.
- [135] H. Yu, N. M. Luscombe, H. X. Lu, X. Zhu, Y. Xia, J. D. Han, N. Bertin, S. Chung, M. Vidal, and M. Gerstein. Annotation transfer between genomes: protein-protein interologs and protein regulogs. *Genome Res.*, 14(6):1107–18., Jun 2004.
- [136] Mohammed Javeed Zaki and Ching-Jiu Hsiao. Charm: An efficient algorithm for closed itemset mining. In *SDM*, 2002.
- [137] P. Zarzov, C. Mazzoni, and C. Mann. The slt2(mpk1) MAP kinase is activated during periods of polarized cell growth in yeast. *EMBO J.*, 15:83–91, 1996.
- [138] Karen I Zeller, Anil G Jegga, Bruce J Aronow, Kathryn A O'Donnell, and Chi V Dang. An integrated database of genes responsive to the Myc oncogenic transcription factor: identification of direct genomic targets. *Genome Biol.*, 4(10):R69, 2003.
- [139] L. Zhang, A. Hach, and C. Wang. Molecular mechanism governing heme signaling in yeast: a higher-order complex mediates heme regulation of the transcriptional activator hap1. *Mol Cell Biol.*, 18(7):3819–28, 1998.
- [140] Y. Zhang, C. Ma, T. Delohery, B. Nasipak, B. C. Foat, A. Bounoutas, H. J. Bussemaker, S. K. Kim, and M. Chalfie. Identification of genes expressed in *C. elegans* touch receptor neurons. *Nature.*, 418(6895):331–5., Jul 18 2002.
- [141] Q. Zhou and W. H. Wong. Cismodule: de novo discovery of cis-regulatory modules by hierarchical mixture modeling. *Proc Natl Acad Sci U S A.*, 101(33):12114–9. Epub 2004 Aug 5., Aug 17 2004.
- [142] Z. Zhou and S. J. Elledge. DUN1 encodes a protein kinase that controls the dna damage response in yeast. *Cell*, 75(6):1119–1127, 1993.
- [143] R. S. Zitomer, P. Carrico, and J. Deckert. Regulation of hypoxic gene expression in yeast. *Kidney Int.*, 51(2):507–13., Feb 1997.
- [144] R. S. Zitomer and C. V. Lowry. Regulation of gene expression by oxygen in *Saccharomyces cerevisiae*. *Microbiol Rev.*, 56(1):1–11., Mar 1992.

Stabilization criteria for GeneClass

Let us consider two different weak rules h_1, h_2 . Let $A_{h_i} = \{\mathbf{x} | h_i(\mathbf{x}) = 1\}$ be the set of examples \mathbf{x} on which learner h_i predicts one. We define the symmetric difference of two sets of examples as the set of examples for which one but not both rules predict one i.e. the set of all examples \mathbf{x}_i for which $h_1(\mathbf{x}_i) + h_2(\mathbf{x}_i) = 1$ holds. The two weak rules h_1 and h_2 then have a highly correlated prediction if the total weight of the symmetric difference $A_{h_1} \ominus A_{h_2}$ is small. We denote this weight by $W(A_{h_1} \ominus A_{h_2})$.

In order to test the statistical significance of $W(A_{h_1} \ominus A_{h_2}) > \epsilon$, we need to consider the distribution of the weights w_i of the examples in the two sets A_{h_1} and A_{h_2} . If the distribution is very skewed, small changes in the symmetric difference set can cause large changes in the corresponding weight. If the distribution is more uniform, then fluctuations in the size of the symmetrical difference set will cause appropriately scaled fluctuations in the weight of the symmetric difference set.

The theorem below is based on Chernoff bounds.

Theorem: Let $X_i \in \{0, 1\}$ ($i = 1, \dots, m$) be m indicator random variables drawn i.i.d. from the distribution $p(X_i = 1) = p$ and $p(X_i = 0) = (1 - p)$. To every X_i is associated a real-valued weight w_i , such that $\sum_i w_i = 1$. Let W be the random variable $W = \sum_i X_i w_i$. Then, for $\epsilon > 0$

$$Pr[W > E(W)(1 + \epsilon)] \leq e^{-\frac{1}{2}\epsilon^2 \frac{p}{1-p} \frac{1}{\sum_i w_i^2}}$$

Proof: For any $t > 0, \kappa > 1$, using Markov's inequality we get

$$Pr[W > \kappa E(W)] = Pr[e^{tW} > e^{t\kappa E(W)}] \leq \frac{E(e^{tW})}{e^{t\kappa E(W)}} = e^{-(t\kappa E(W) - \ln E(e^{tW}))} \quad (\text{A.1})$$

Let $h(t) \equiv t\kappa E(W) - \ln E(e^{tW})$ and using $E(W) = \sum_i p w_i = p$ we have

$$h(t) = t\kappa p - \ln E(e^{t \sum_i w_i X_i}) \quad (\text{A.2})$$

$$= t\kappa p - \ln E(\prod_i e^{t w_i X_i}) \quad (\text{A.3})$$

$$= t\kappa p - \ln \prod_i E(e^{t w_i X_i}), \text{ (independence)} \quad (\text{A.4})$$

$$= t\kappa p - \sum_i \ln (p e^{t w_i} + (1 - p)) \quad (\text{A.5})$$

We now optimize the inequality by maximizing $h(t)$ for the case where $w_i \ll 1$ (usually the case if m large). Taylor expansion up to second order in w_i gives

$$h(t) \approx t\kappa p - \sum_i p t w_i + \frac{1}{2}(1 - p)p t^2 w_i^2 \quad (\text{A.6})$$

$$= t\kappa p - p t - \frac{1}{2}(1 - p)p t^2 \sum_i w_i^2 \quad (\text{A.7})$$

$$\frac{dh(t)}{dt} = \kappa p - p - (1 - p)p t \sum_i w_i^2 \quad (\text{A.8})$$

then

$$\left. \frac{dh(t)}{dt} \right|_{t^*} = 0 \quad (\text{A.9})$$

$$\Leftrightarrow t^* = \frac{\kappa p - p}{(1 - p)p \sum_i w_i^2} \quad (\text{A.10})$$

so

$$h(t^*) = \frac{(\kappa p - p)^2}{(1-p)p \sum_i w_i^2} - \frac{1}{2} \frac{(\kappa p - p)^2}{(1-p)p \sum_i w_i^2} \quad (\text{A.11})$$

$$h(t^*) = \frac{1}{2} \frac{(\kappa p - p)^2}{(1-p)p \sum_i w_i^2} \quad (\text{A.12})$$

substituting into equation A.1 gives

$$\Pr[W > E(W)\kappa] = \exp\left(-\frac{1}{2} \frac{(\kappa p - p)^2}{(1-p)p \sum_i w_i^2}\right) \quad (\text{A.13})$$

and setting $\epsilon = \kappa - 1$

$$\Pr[W > E(W)(1 + \epsilon)] = \exp\left(-\frac{1}{2} \epsilon^2 \frac{p}{(1-p)} \frac{1}{\sum_i w_i^2}\right) \quad (\text{A.14})$$

□

This bound suggests that to use a parameter η to test the statistical significance of $W > E(W)(1 + \epsilon)$ where

$$\exp\left(-\frac{1}{2} \epsilon^2 \frac{p}{(1-p)} \frac{1}{\sum_i w_i^2}\right) = \eta \quad (\text{A.15})$$

$$\epsilon = \sqrt{\ln\left(\frac{1}{\eta^2}\right) \frac{1-p}{p} \sum_i w_i^2} \quad (\text{A.16})$$

Hence, values of W are statistically significant for

$$\frac{W}{E(W)} - 1 > \sqrt{\ln\left(\frac{1}{\eta^2}\right) \frac{1-p}{p} \sum_i w_i^2} \quad (\text{A.17})$$

$$= \eta' \sqrt{\sum_i w_i^2} \quad (\text{A.18})$$

where η' absorbs parameters p and η .

While the theorem does not directly map to the GeneClass algorithm it suggests the term $\sqrt{\sum_i w_i^2}$ to test the statistical significance for a sum of weighted random variables. Therefore, for a pair of weak rules h_1 and h_2 , we average them if

$$W(A_{h_1} \ominus A_{h_2}) \leq \eta_1 \sqrt{\sum_i w_i^2} \quad (\text{A.19})$$

where η_1 is an empirically determined parameter and the summation of weights is over all examples that reach the prediction node to which we want to add the stabilized weak rule. For unnormalized weights we use

$$W(A_{h_1} \ominus A_{h_2}) \leq \eta_1 \sqrt{\frac{\sum_i w_i^2}{(\sum_i w_i)^2}} \quad (\text{A.20})$$

GeneClass pseudocode

We start with a candidate set of M motifs $\{\mu\}$ representing known or putative transcription factor binding sites and a candidate set of R regulators $\{\pi\}$. Let $M_{\mu g} \in \{1, 0\}$ represent the presence or absence of a motif μ in the regulatory sequence of a gene g . Each gene g can then be represented by a vector $\{M_{\mu g}\}$ of motif occurrences. Let $P_{\pi e} \in \{-1, 0, 1\}$ represent the state of regulator π in an experiment e . The experimental context can then be represented by a vector $\{P_{\pi e}\}$ of the expression states of all the candidate regulators in that experiment. The feature vector for each training example x_{ge} is given by $\{\{M_{\mu g}\}, \{P_{\pi e}\}\}$. The hypothesis space (set of weak rules) on which the prediction function is defined can be written as $\chi = \{-1, 0, 1\}^R \times \{0, 1\}^M$. Each weak rule is a pairing of a motif μ with a regulator π in state s represented by $[\mu, \pi, s]$. GeneClass uses the Adaboost algorithm to learn an alternating decision tree (ADT). An ADT consists of alternate rows of splitter nodes and prediction nodes. At each iteration $t = 1 \dots T$, a splitter node and its corresponding prediction node are added to the tree. The splitter node contains the weak rule h_t output by the weak learner and the prediction node contains the coefficient for the weak rule α_t . The weak learner selects the weak rule that minimizes boosting loss L at each iteration. This means picking a motif-regulator pair $[\mu, \pi, s]$ and its position in the tree i.e the prediction node to which the new splitter node is to be added. A prediction node can be followed by multiple splitter nodes.

Definitions: For training set $\{(x_{ge}, y_{ge}) : (g, e) \text{ is differentially expressed}\}$ of size N :

x_{ge} = feature representation for (g, e) = motif/ChIP chip profile for g , regulator expression levels in e

y_{ge} = label of $(g, e) \in \{\pm 1\}$

c_1 = precondition corresponding to existing path in ADT

$c_2(\mu, \pi, s)$ = new condition corresponding to motif μ , regulator π in state s ,

w_{ge} = weight of example (g, e) ,

$$W(c) = \sum_{(g,e):c=1} w_{ge}$$

$$W_+(c) = \sum_{(g,e):c=1, y_{ge}=+1} w_{ge}$$

$$W_-(c) = \sum_{(g,e):c=1, y_{ge}=-1} w_{ge}$$

$h[c]$ = weak rule for condition c , i.e $h[c](x_{ge}) = 1(0)$ exactly when $c = 1(0)$

α_t = coefficient of weak rule at iteration t

$F(x_{ge})$ = prediction value for (g, e)

Parameters:

T = number of boosting iterations,

η_1 = parameter deciding how “similar” the stabilized features should be ($\eta_1 > 0$),

η_2 = parameter deciding whether to stabilize at a given iteration ($\eta_2 > 0$),

Initialization: Set root of tree to be single prediction node corresponding to a constant weak rule:

$$h_0 = 1$$

$$\alpha_0 = \frac{1}{2} \log \frac{W_+(1)}{W_-(1)}$$

Initialize weights: $w_{ge} \leftarrow \frac{1}{N} \exp(-y_{ge}\alpha_0)$

Main Loop:

for $t = 1 \dots T$

$$L(c_1, c_2) = W(\neg c_1 \vee \neg c_2) + 2 \sqrt{W_+(c_1 \wedge c_2) W_-(c_1 \wedge c_2)}$$

$$(c_1^*, c_2^*(\mu^*, \pi^*, s^*)) = \operatorname{argmin}_{c_1, c_2} (L_1(c_1, c_2(\mu, \pi, s))),$$

Calculate stabilization criterion γ

$$\gamma(c_1^* \wedge c_2^*) = \frac{1}{2} |W_+(c_1^* \wedge c_2^*) - W_-(c_1^* \wedge c_2^*)|$$

$$\text{if } \gamma(c_1^* \wedge c_2^*) \geq \eta_2 \sqrt{\frac{\sum_{(g,e), c_1^*=1} w_{(g,e)}^2}{(\sum_{(g,e), c_1^*=1} w_{(g,e)})^2}}$$

for $\mu = 1 \dots M$, for $\pi = 1 \dots P$, for $s = 1 \dots S$,

$$\text{do } \Delta(\mu, \pi, s) = \sum_{c_1^*=1, (c_2^*=1 \wedge c_2(\mu, \pi, s)=0) \vee (c_2^*=0 \wedge c_2(\mu, \pi, s)=1)} w_i$$

end for

Obtain set C of weak rules to be averaged

$$C := \{c_2(\mu, \pi, s) \mid \Delta(\mu, \pi, s) \leq \eta_1 \sqrt{\frac{\sum_{(g,e), c_1^*=1} w_{(g,e)}^2}{(\sum_{(g,e), c_1^*=1} w_{(g,e)})^2}}\}$$

$$\text{for all } c_2(\mu, \pi, s) \in C \text{ do } \alpha(\mu, \pi, s) = \frac{1}{2} \log \frac{W_+(c_1^* \wedge c_2(\mu, \pi, s))}{W_-(c_1^* \wedge c_2(\mu, \pi, s))}$$

$$c(\theta) = \left(\sum_C |\alpha_1(\mu, \pi, s) h_{c_1^* \wedge c(\mu, \pi, s)}| > \theta \right) \in \{0, 1\}$$

$$\theta^* = \frac{1}{2} (\sum_C |\alpha(\mu, \pi, s)| - \min_C |\alpha(\mu, \pi, s)|)$$

$$\alpha_t = \frac{1}{2} \log \frac{W_+(c_1^* \wedge c(\theta^*))}{W_-(c_1^* \wedge c(\theta^*))}$$

$$h_t = h[c_1^* \wedge c(\theta^*)]$$

else

$$\alpha_t = \frac{1}{2} \log \frac{W_+(c_1^* \wedge c_2^*)}{W_-(c_1^* \wedge c_2^*)}$$

$$h_t = h[c_1^* \wedge c_2^*]$$

end if

Update weights of all examples:

$$w_{(g,e)} \leftarrow w_{(g,e)} \exp(-y_{(g,e)} \alpha_t h_t(x_{(g,e)}))$$

end for

Output final prediction function: $F(x_{(g,e)}) = \sum_{t=0}^T \alpha_t h_t(x_{(g,e)})$

MEDUSA pseudocode

The inputs to the algorithm are (i) the promoter sequences of target genes and (ii) the discretized expression levels of a set of candidate regulator genes. The sequence data is represented only via occurrence or non-occurrence of motifs represented by all possible length- k words known as k -mers and gapped homodimers. We restrict the set of all dimers to those whose two components (monomers) have specific relationships, consistent with most known dimer motifs: equal (e.g. ACG_ACG), reversed (e.g. ACG_GCA), complements (e.g. ACG_TGC), or reverse complements (e.g. ACG_CGT). Let $\{\mu_d\}$ represent the set of deterministic motifs i.e. all k -mers and dimers. Let $M_{\mu_d g}$ indicate the presence ($M_{\mu_d g} = 1$) or absence ($M_{\mu_d g} = 0$) of a motif μ_d in the promoter sequence of gene g , and let $P_{\pi e}^s$ indicate the up-regulation ($s = +1$) or down-regulation ($s = -1$) of a regulator π in experiment e ($P_{\pi e}^s = 1$, if regulator π is in state s in experiment e , and $P_{\pi e}^s = 0$, otherwise). Hence, the feature vector for a gene g in an experiment e is given by $\{\{M_{\mu_d g}\}, \{P_{\pi e}^s\}\}$. MEDUSA uses the Adaboost algorithm to learn an alternating decision tree (ADT). An ADT consists of alternate rows of splitter nodes and prediction nodes. At each iteration $t = 1 \dots T$, a splitter node and its corresponding prediction node are added to the tree. The splitter node contains the weak rule h_t output by the weak learner and the prediction node contains the coefficient for the weak rule α_t . Each weak rule is a pairing of a motif μ with a regulator π in state s represented by $[\mu, \pi, s]$. The motif μ can be a deterministic sequence motif μ_d or a PSSM μ_p .

which is obtained using hierarchical sequence agglomeration at each iteration. The weak learner selects the weak rule that minimizes boosting loss L at each iteration. This means picking a motif-regulator pair $[\mu, \pi, s]$ and its position in the tree i.e the prediction node to which the new splitter node is to be added. A prediction node can be followed by multiple splitter nodes.

Definitions: For training set $\{(x_{ge}, y_{ge}) : (g, e) \text{ is differentially expressed}\}$ of size N :

x_{ge} = feature representation for (g, e) = promoter sequence for g , regulator expression levels in e

y_{ge} = label of $(g, e) \in \{\pm 1\}$

c_1 = precondition corresponding to existing path in ADT

$c_2(\mu, \pi, s)$ = new condition corresponding to motif μ , regulator π in state s ,

μ_d = deterministic motif (k -mer or gapped homodimer)

μ_p = probabilistic motif (PSSM)

w_{ge} = weight of example (g, e) ,

$$W(c) = \sum_{(g,e):c=1} w_{ge}$$

$$W_+(c) = \sum_{(g,e):c=1, y_{ge}=+1} w_{ge}$$

$$W_-(c) = \sum_{(g,e):c=1, y_{ge}=-1} w_{ge}$$

$h[c]$ = weak rule for condition c , i.e $h[c](x_{ge}) = 1(0)$ exactly when $c = 1(0)$

$$L(c_1, c_2) = \text{loss for } h[c_1 \wedge c_2] = W(\neg c_1 \vee \neg c_2) + 2 \sqrt{W_+(c_1 \wedge c_2) W_-(c_1 \wedge c_2)}$$

α_t = coefficient of weak rule at iteration t

Initialization: Set root of tree to be single prediction node corresponding to a constant weak rule:

$$h_0 = 1$$

$$\alpha_0 = \frac{1}{2} \log \frac{W_+(1)}{W_-(1)}$$

Initialize weights: $w_{ge} \leftarrow \frac{1}{N} \exp(-y_{ge} \alpha_0)$

Main Loop:

for $t = 1 \dots T$

Minimize boosting loss over preconditions c_1 already in ADT and choices of deterministic motif-regulator conditions c_2 :

$$(c_1^*, c_2(\mu^*, \pi^*, s^*)) = \operatorname{argmin}_{c_1, c_2} L(c_1, c_2(\mu_d, \pi, s))$$

Learn best PSSM from best deterministic motif:

Take top K motifs $\mu_d^1 = \mu^*, \mu_d^2, \dots, \mu_d^K$ with lowest boosting loss $L(c_1^*, c_2(\mu_d, \pi^*, s^*))$

Apply hierarchical agglomerative clustering to $\{\mu_d^j\}_{j=1\dots K}$

Obtain optimal probabilistic motif μ_p^* and threshold Θ^*

If loss is smaller than best deterministic motif, set $\mu^* \leftarrow (\mu_p^*, \Theta^*)$ Add new splitter node and prediction node to the ADT corresponding to new weak rule:

$$h_t = h[c_1^* \wedge c_2^*(\mu^*, \pi^*, s^*)]$$

$$\alpha_t = \frac{1}{2} \log \frac{W_+(c_1^* \wedge c_2^*)}{W_-(c_1^* \wedge c_2^*)}$$

Update weights of all examples:

$$w_{ge} \leftarrow w_{ge} \exp(-y_{ge} \alpha_t h_t(x_{ge}))$$

end for

Output final prediction function: $F(x_{ge}) = \sum_{t=0}^T \alpha_t h_t(x_{ge})$

Hierarchical Agglomeration:

Given deterministic motifs $\mu_d^1 = \mu^*, \mu_d^2, \dots, \mu_d^K$ associated to $(c_1^*, c_2(\mu^*, \pi^*, s^*))$:

Convert each μ_d^j into a PSSM μ_p^j using small pseudocounts

Perform hierarchical clustering agglomeration:

For a pair of PSSMs μ_p^1 and μ_p^2 , we obtain a new PSSM μ_p^{12}

that minimizes the symmetrized KL-divergence

$$d(\mu_p^1, \mu_p^2) \equiv \min_{\text{offsets}} [b_1 D_{KL}(p \| b_1 \mu_p^1 + b_2 \mu_p^2) + b_2 D_{KL}(q \| b_1 \mu_p^1 + b_2 \mu_p^2)]$$

where $b_{1,2} = G_{1,2}/(G_1 + G_2)$, G_1, G_2 are the numbers of target genes

for μ_p^1 and μ_p^2

At each step, agglomerate pair with minimal $d(\mu_p^1, \mu_p^2)$ to obtain μ_p^{12}

Set PSSM threshold $\theta_{\mu_p^{12}} \leftarrow \text{argmin}_{\theta} (L_1(c_1^*, c_2(\mu_p^{12}, \pi^*, s^*)))$

Obtain $K - 1$ candidate PSSMs from agglomeration

Return $\mu_p^* = \text{argmin}_{\{\text{candidates } \mu_p\}} L(c_1^*, c_2(\mu_p, \pi^*, s^*))$, corresponding Θ^*

Yeast samples and microarray data generation for the hypoxia dataset

In this section, we describe details of data generation and microarray data processing for the hypoxia dataset. The data is generated in the laboratory of our collaborator Dr. Li Zhang.

D.1 Yeast cell growth and treatment

Yeast strains used are L51 (MATa, *ura3-52*, *leu2-3*, *112*, *his4-519*, *ade1-100*, *trp1::HisG*, *hap1::LEU2*) and MHY100 (MATa, *ura3-52*, *leu2-3*, *112*, *his4-519*, *ade1-100*, *hem1-Δ100*) [139]. L51 is used for studies of oxygen regulation, and MHY100 is used for studies of heme regulation. To avoid variations from the differences accumulated after many generations of growth of strains, we transform the L51 strain with the *HAP1* gene deleted for studies of Hap1 function. Hap1 protein is expressed in L51 cells by transforming an ARS-CEN plasmid bearing the complete *HAP1* genomic sequence [89]. For comparison with cells without Hap1 expressed, an empty vector is transformed into L51 cells. The use of Hap1 expression plasmid generates much more reproducible results than the use of different strains. Yeast cells with or without Hap1 expressed grow at similar rates under both anaerobic and aerobic conditions.

We choose to use a low oxygen level (10 ppb) in this study, in order to identify oxygen-regulated genes. Previous studies have shown that most, if not all, oxygen-regulated genes are affected at low concentrations, but some genes are not affected at higher oxygen levels (for example, > 1 ppm) [19, 50]. Anaerobic (~10 ppb O₂, measured by using an oxygen monitor and confirmed by CHEMetrics oxygen kits) growth condition is created by using an anaerobic chamber (Coy Laboratory, Inc.) and by filling the chamber with a mixture of 5% H₂ and 95% N₂ in the presence of palladium catalyst [50]. The oxygen level in the chamber is monitored by using the Model 10 gas analyzer (COY laboratory Inc). H₂ is filled to keep the measured oxygen level at zero. The precise level of oxygen is further measured by using rhodazine kit (K-7511) with MDL at 1 ppb, and a range of 0-20 ppb (http://www.envco.info/prod.php?product_id=469). L51 cells bearing the Hap1 expression or empty vector are grown under normoxic or anaerobic conditions for 1.5 or 6 hours. The UAS1/CYC1-lacZ reporter plasmid [41] is transformed into yeast cells to confirm the expression of Hap1 and the oxygen levels. Cells are grown in yeast synthetic complete media. Co²⁺-induced cells are grown in the presence of 400 μ M cobalt chloride for 6 hours, as described previously [55, 56]. MHY100 cells are grown in medium containing 2.5 μ g/ml (heme-deficient) or 250 μ g/ml (heme-sufficient) 5-aminolevulinic acid [139]. For RNA preparations, yeast cells are inoculated so that the optical density of yeast cells is in the range of 0.8-1.0 immediately before the collection of cells.

D.2 RNA preparation and microarray gene expression profiling

RNA is extracted from yeast cells exactly as previously described [4]. RNA samples are prepared from 8 different experimental conditions:

Normoxic (HAP1): L51 yeast cells bearing the Hap1 expression plasmid maintained under aerobic conditions.

Normoxic (Δ hap1): L51 yeast cells bearing the empty expression plasmid maintained under aerobic conditions.

Anaerobic, early (HAP1): L51 yeast cells bearing the Hap1 expression plasmid maintained under anaerobic conditions for 1.5 hours.

Anaerobic, late (HAP1): L51 yeast cells bearing the Hap1 expression plasmid maintained under anaerobic conditions for 6 hours.

Anaerobic, late (Δ hap1): L51 yeast cells bearing the empty expression plasmid maintained under anaerobic conditions for 6 hours.

Normoxic, +Co²⁺ (HAP1): L51 yeast cells bearing the Hap1 expression plasmid in the presence of 400 μ M cobalt chloride for 6 hours.

Heme sufficient (Heme): MHY100 cells grown in medium containing 250 μ g/ml (heme-sufficient) 5-aminolevulinic acid.

Heme deficient (Δ Heme): MHY100 cells grown in medium containing 2.5 μ g/ml (heme-deficient) 5-aminolevulinic acid.

For each condition, three replicates are generated by preparing RNA samples from three batches of independently grown cells. Microarray expression analyses are performed by using three batches of replicate RNA samples. The quality of RNA is high as assessed by measuring absorbance at 260 and 280 nm, by gel electrophoresis, and by the quality of microarray data (see below).

The synthesis of cDNA and biotin-labeled cRNA are carried out exactly as described in the Affymetrix GeneChip Expression Analysis Technical Manual (2000). The yeast *Saccharomyces cerevisiae* genome 2.0 arrays were purchased from Affymetrix, Inc. Probe hybridization and data collection are carried out by the Columbia University Affymetrix GeneChip processing center. Specifically, the Affymetrix GeneChip Hybridization Oven

640 and the next generation GeneChip Fluidics Station 450 are used for hybridization and chip processing. Chip scanning is performed by using the GeneChip scanner 3000. Initial data acquisition, analysis is performed by using the Affymetrix Microarray suite. By using GCOS1.2 with the PLIER (probe logarithmic intensity error) algorithm, we calculate and examine the parameters reflecting the image quality of the arrays. Arrays with a high background level in any region are discarded and replaced. The average noise or background level is limited to less than 5%. The average intensity for those genes judged to be present is at least 10-fold higher than those judged to be absent. Also, arrays that deviate considerably in the percentage of present and absent genes from the majority of the arrays are replaced. Arrays with a β -actin 3'/5' ratio greater than 2 are replaced.

D.3 Normalization and discretization of microarray data

For each microarray, we convert the .DAT image files into .CEL files using the Affymetrix GCOS software. These raw .CEL files are further processed into expression values using the RMA express software by Bolstad *et al.* [15]. This software uses the robust multiarray average method by Irizarry *et al.* [54], which involves a background correction and a quantile-based normalization scheme.

Each of the knockout, stress or perturbation microarray experiments is compared to a corresponding reference microarray. The expression fold-changes are converted to p -values using an intensity-specific noise model obtained from replicate data (See Section 5.5.1.1). The fold-changes are then discretized into +1, 0 or -1 labels using a p -value threshold of 0.05. A label of +1 (-1) indicates up-regulation (down-regulation) beyond the threshold level of noise.

Functional annotations for the hypoxia expression signatures

As shown in Figure 6.1, we identify 16 distinct discretized co-expression signatures in the hypoxia dataset. We perform Gene Ontology functional analysis (See Section 5.5.1.4) on the 16 expression signatures. In most cases, the expression signatures can be assigned significant functional terms, though in general only a smaller subset of the genes in a signature belong to the enriched category.

Signature 1 (sig1) consists of 70 genes that are strongly upregulated 6 hours into anaerobiosis independent of Hap1 deletion. They are also upregulated in response to heme deletion, as opposed to the 267 genes in signature 3 (sig3) which are exclusively upregulated in the late hypoxia experiments. Sig1 genes are mainly involved in cell wall biogenesis ($5.9\text{e-}07$) and stress response ($2.6\text{e-}06$). The biotin biosynthesis gene cassette (BIO3/4/5) is also part of this group.

Sig3 on the other hand is enriched for genes involved in carbohydrate and alcohol metabolism ($2.4\text{e-}07$).

Signature 15 (sig15) shows strong induction in late hypoxia similar to Sig3. However, Sig15 genes are also significantly downregulated in early hypoxia. This set is also weakly enriched for genes involved in carbohydrate metabolism ($1.4\text{e-}05$).

Signature 2 (sig2) is enriched for several essential transcription factors ($1e-14$) and rRNA/ribosome processing genes ($1e-14$). These 207 genes are induced in early hypoxia but suppressed at the 6 hour time point independent of Hap1 deletion. These genes are also significantly downregulated in the heme deletion experiment.

Signature 5 (Sig5) is diametrically opposite to Sig1 but functionally similar to sig2. It consists of typical stress suppressed genes involved in ribosome biogenesis ($2.5e-10$).

Signature 6 (sig6) consists of 160 genes several of which are involved in ATP synthesis dependent proton transport and respiration ($3.1e-10$). These genes are strongly suppressed in all but the Δ hap1 condition, indicating that they might not be regulated by Hap1.

However, signature 16 (sig16) consists of 34 Hap1-dependent genes that are strongly suppressed in all conditions including the Δ hap1 experiment. These genes (such as the *COX* and *QCR* genes) are involved in aerobic respiratory processes ($1.0e-14$), electron transport ($4.4e-14$) and heme-dependent oxidoreductase activity ($5.8e-14$).

Signature 8 (sig8) shows strong downregulation in Δ heme and late hypoxia (Δ hap1) experiments and weak suppression in the other conditions. It is mainly made up of structural constituents of ribosomes ($1.4e-06$) and other genes involved in protein synthesis and metabolism ($5.8e-06$).

Signature 9 (sig9) and 10 (sig10) consist of genes significantly downregulated in the late hypoxia conditions. However, sig9 also shows weak induction in early hypoxia. Sig9 is made up of several cell cycle genes ($5.6e-12$). The histone genes ($3.0e-07$) are part of sig10.

Genes in signature 11 (sig11) and signature 14 (sig14) appear to be strongly regulated by heme. Sig11 genes are exclusively induced by heme deletion whereas sig14 genes are suppressed in the same experiment. The ergosterol biosynthesis genes which are part of sig14 are known to be heme regulated. Sig11 is an intriguing set of genes made up of 500 genes of which 354 are functionally uncharacterized ($3.2e-13$). These could be an important class of heme-regulated genes.

Signature 13 (sig13) consists of 276 genes many of which are transcription factors

(3.15×10^{-14}) and signal transduction factors (7.23×10^{-8}). These genes are exclusively induced in early hypoxia and seem to represent an early regulatory response.

Downloadable source code and data

GeneClass MATLAB source code for the GeneClass algorithm described in Chapter 4 -

<http://www.ccls.columbia.edu/compbio/robust-geneclasse>

MEDUSA MATLAB source code for the MEDUSA algorithm described in Chapter 5 -

<http://www.ccls.columbia.edu/compbio/medusa>

Hypoxia dataset Datasets and source code for the analysis of the response of yeast to hy-

poxia as described in Chapter 6 - <http://cbio.mskcc.org/leslielab/projects/>

medusa