Sackler Faculty of Exact Sciences, School of Computer Science

Computational Analysis of Transcriptional Programs: Function and Evolution

THESIS SUBMITTED FOR THE DEGREE OF
“DOCTOR OF PHILOSOPHY”

by

Amos Tanay

The work on this thesis has been carried out
under the supervision of **Prof. Ron Shamir**

Submitted to the Senate of Tel-Aviv University
August 2005

Acknowledgments

It is a real pleasure to write this short note, acknowledging the many people that helped me during the research for this thesis.

I'm first and foremost indebted to my adviser Prof. Ron Shamir for agreeing to take me from the high-tech battlefield and for showing me the way in his rare combination of scientific integrity and open and broad mindedness. I still do not understand how can Ron do so much, so well. I learn from him continuously at all levels.

There are so many others I want to thank. I'll just mention some of the names; my deep appreciation goes to all of you: To my collaborators Irit Gat-Viks (first not only alphabetically), Dana Pe'er, Daniela Raijman, Aviv Regev, Roded Sharan and Israel Steinfeld. To Prof. Martin Kupiec for many insightful discussions and ideas. To Nir Friedman. To Benny Chor. To Jacque Tanay for making me read that DOS manual when I was 12, and R. Godemant's "Algebra" when I was 21, among other things. To my collaborators at the Church Lab, Aimee Dudley and Daniel Jense. To all of my current and former lab mates, in particular to Rani Elkon, Shay Halperin, Tzvika Hartman, Gadi Kimmel, Chaim Linhart, Adi Maron-Katz, Itsik Pe'er, Rotem Sorek and Igor Ulitsky. To the Horovitz Complexity center, for the support.

I also thank Rotem, Yotam and Noa (I'm aligning to the standard here), but I'm keeping the option to hug them as well.

Finally (and you, the reader, may join me in this), I have to thank my good luck for being able to be here during this exciting revolution of 21st century biology. Since I first read "the Cell", I was surprised almost daily by what nature is still hiding from us. Just reading the journals every week, not to mention probing into the data, is deeply satisfying (and fun!), wouldn't you agree?

Abstract

Transcriptional programs are operating in all living organisms, allowing the activity of a genome to be controlled by regulated modifications of the transcription of its genes. Transcriptional programs are therefore critical to the proper operation of many biological mechanisms, and their characterization is one of the most important challenges in modern biology. In this thesis we describe our studies of biological transcriptional programs. We use computational techniques from graph theory, probabilistic models and combinatorial optimization and integrate them with biological principles to develop models for transcriptional programs and algorithms that learn them from data. We use our methods to study extensive biological datasets on yeast and derive applicable hypotheses on the regulation of specific biological processes. We study the evolving transcriptional program at the single locus and at the co-regulated gene module level. We infer the function of extant systems from their evolutionary behavior and suggest new evolutionary mechanisms by merging evolutionary theories with functional genomic information.

Contents

Chapter 1

Introduction and main results

Transcriptional programs are operating in all living organisms, allowing the activity of a genome to be controlled by regulated modifications of the transcription of its genes. Transcriptional programs are therefore critical to the proper operation of many biological mechanisms, and their characterization is one of the most important steps in the journey of modern biology towards the understanding of complex biological processes. In this thesis we describe our studies of biological transcriptional programs: their mathematical modeling and the algorithmic aspects of inferring them from data. We specifically focus on the interplay between the evolutionary process that gave rise to extant transcriptional programs and the functional properties of the regulatory switches in them. We build understanding of function and evolution from a unified point of view, using ideas and results from functional analysis to understand evolution and vise versa.

Our field of research (termed ”computational biology” or ”bioinformatics”) is combining themes from biology and computer science (or mathematics in general) as well as ideas from other disciplines (physics, engineering). As the new discipline develops, deeper and deeper integration of concepts is constantly in demand. It was suggested that computational biology is not merely a new kind of applied mathematics, and that new problems from biology have the potential to project back into mathematics and play a role similar to that played by physics for centuries [24]. On the other hand, the role of mathematical thinking in biology may be even more profound. The traditional approach of biologists, which is based on careful (usually qualitative) study of examples, is challenged by the need to analyze quantitatively very large networks of complex regulatory interactions. The technological revolution of high throughput biology increased dramatically the amount of experimental information that is routinely collected, making experimental results intractable by the traditional manual expert analysis. The research we report on in this thesis is therefore relying on intimate integration of ideas and methods from mathematics and biology. While we approach the problems from the computer science perspective, we do discuss many biological examples and the biological implications of our computational studies. A short survey (Chapter 2) provides some background on the most important biological concepts we are using, but cannot substitute the standard textbook introduction to molecular biology [2] and molecular evolution [89] or the up-to-date reviews on specific biological model systems we are using.

Below we outline the main results presented in this thesis. We move in two parallel paths studying functional and evolutionary aspects of transcriptional switches and their role in biological regulation. On the functional path we take a top-down approach, starting with the development of algorithms for the dissection of large biological networks into *modules* and then extending the formulation to construct detailed models for transcriptional regulation inside a module. On the evolutionary path we take a complementary approach. We start from the single locus level, analyzing first the selective forces that act on transcription factor binding sites. We then move up the hierarchy, and develop tools to study the evolution of transcriptional modules and their regulation.

1.1 Function: from modules to models of transcriptional programs

We define a *functional (gene) module* as a group of genes that share common biological properties in a statistically significant way. The notion of biological properties is a mathematical abstraction that can be used to represent almost any form of genome-wide experiment, including profiles of gene expression, transcription factor location, protein interactions, growth sensitivity and more. In Chapter 3 we introduce *biclustering* as a method to detect functional modules given a large compendium of heterogeneous biological datasets. We develop a graph theoretic formulation that represents the data as a weighted bipartite graph and the biclustering problem as the problem of finding the heaviest subgraphs in the graph. We discuss the combinatorial aspects of the problem, characterize its complexity, and provide a polynomial algorithm in case nodes on one side of the bipartite graph have bounded degrees. We also discuss the problem of finding an optimal set of biclusters in the graph and provide two formulations for it, one combinatorial and the other probabilistic. Our methods, along with several heuristics that enable practical computations with very large datasets, were implemented in the SAMBA program for biclustering biological data.

Results from this chapter were published in the *Proceeding of ISMB 2002* [134] and in the *Handbook of Bioinformatics* [135]. The work described in the chapter was done in collaboration with Roded Sharan.

In Chapter 4 we discuss applications of our biclustering algorithm to study function in the budding yeast (*S. cerevisiae*) genome. We have constructed a large experimental compendium by combining data from 60 different studies using 7 different technologies. We applied SAMBA to generate a comprehensive set of functional modules. We present examples for modules that successfully integrate data from several sources and show how to construct models for regulation given the highly specific biclusters that SAMBA derives from the large compendium. We systematically assign putative function to 800 uncharacterized yeast genes and validate our specificity both experimentally and computationally. The collection of functional modules also allows for the characterization of global principles in the yeast regulatory network. We show that the network is modular and often organized hierarchically. We discuss in detail the analysis of two specific experiments using the compendium and the SAMBA-derived functional modules. For experiments perturbing the galactose metabolic pathway we reveal regulatory discrepancy for specific mutants and suggest an explanation for the slow growth of these mutants. For a set of experiments treating yeasts with hyper-osmotic shock, we characterize an additive transcriptional switch that combines signals from two signaling pathways.

Results in this chapter were published in *PNAS* (2004) [133] and in *Molecular Systmes Biology* (2005) [136]. This study was done in collaboration with Martin Kupiec's lab. Another application of SAMBA that was performed in collaboration with George Church's lab (Harvard) is not described in this thesis. That study was published in *Molecular Systems Biology* (2005) [34]

We develop an extended model for transcriptional programs in Chapter 5. The model explicitly expresses three important components of transcriptional switches: a set of transcription factors (TFs), their binding characteristics to a gene's promoter, and the logical relations among their binding sites. The three components together determine the regulation of the gene's transcrtipion. We introduce a combinatorial model that represents these biological entities, and develop polynomial algorithms for optimizing key model features. We show how to use experimental data in order to learn other features of the model heuristically and how to infer values for hidden variables. We then revisit the response to perturbation in the galactose pathway and apply our algorithms to reveal the regulatory mechanisms that control galactose enzymes and regulators.

Results from this chapter were published in *Proceedings of RECOMB 2003* [131]. A journal version was published in the Journal of Computational Biology (2004) [132].

1.2 Evolution: from binding sites to the evolving transcriptional module

Transcriptional networks are defined by the interactions of transcription factors and genes. Physically, a protein-DNA interaction associates a sequence-specific transcription factor and a short DNA sequence, called here *transcription factor binding site* (TFBS). Since evolution, to a first approximation, operates on the DNA, the smallest component of the transcriptional network with direct evolutionary relevance is the single TFBS. In Chapter 6 we study the evolution of TFBSs, aiming at the functional characterization of these short sequences using analysis of the selective forces acting on them. By a systematic phylogenetic analysis and statistical estimation of the rate of substitution between any pair of DNA octamers in promoters we represent the entire TFBS vocabulary as nodes in a graph and the evolutionary relations between pairs of TFBSs as edges in that graph. We call this graph the *TFBS selection network* and use it to infer TFBSs' function. First, we identify dense clusters of TFBSs (groups of similar short sequences with high substitution rates between them). We prove that over 90% of them consist of sequences of either known binding sites of characterized TFs, or can be validated to be functionally related using other sources of information (gene expression or TF location). Clusters in the selection network therefore correspond to targets of specific transcription factors. Second, we observe several cases where the set of targets that are identifiable by a single TF are subdivided into two clusters, suggesting that evolution avoids substituting a TFBS from one subcluster by a TFBS from another subcluster. We analyze two such cases in detail, showing that TFBSs in each subcluster have different binding affinities, supporting a possible functional role for the observed subdivision.

Results from this Chapter were published in *Genome Research* (2004) [128]. This study was done in collaboration with Irit Gat-Viks.

The organization of transcriptional networks into modules is a principle that carry great functional importance in the coordination of complex cellular responses. In Chapter 7 we study the evolutionary consequences of this functional modularity. Specifically, we explore the effects that co-regulation of dozens of genes may have on the evolution of the TFBSs in their promoters. We develop methods for comparative gene expression analysis and identify transcriptional modules that are conserved between the distant yeasts *S. cerevisiae* and *S. pombe*. We then combine sequence analysis with phylogenetic reconstruction to explore the evolution of the TFBSs that drive the co-regulation of genes in these transcriptional modules. Using our methods, we discover cases where conserved transcriptional modules have conserved TFBSs, and, more suprisingly, conserved modules showing complete divergence at the TFBS-level. The most notable case of cis-regulatory divergence is the module of ribosomal proteins, containing over 100 tightly co-regulated genes. Phylogenetic analysis of the evolutionary history of the regulatory regions controling ribosomal proteins suggests that divergence of TFBSs happened in three phases: First, an ancient TFBS controlled the module. Second, a novel TFBS emerged in dozens of promoters, generating a redundant transcription program that persisted for some time in the history of the module. Third, the ancient TFBS was eliminated (in some lineages) from virtually all of the promoters, generating extant regulatory schemes that use only the new element. In other cases, our analysis suggests alternative scenarios for the evolution of modules' regulation, including, for example, gradual drift of the TFBSs sequences. The paradigm we introduce for the understanding of the evolution of transcriptional modules combines several of the methods described in this thesis, and may be the basis for future attempts to understand the evolution of more complex entities in transcriptional networks.

The results from this Chapter were published in *PNAS* (2005) [129]. This study was done in collaboration with Aviv Regev (Harvard).

Chapter 2

Background

In this chapter we give a brief introduction to some of the biological concepts used in this thesis. We cite only the key facts and provide references for more extensive expositions of the various subjects matters. For background on computational concepts used in the thesis, see e.g., [40, 102, 36].

2.1 Genes and genomes

According to the *central dogma* of molecular biology, most of the information that defines living organisms is encoded in their DNA, which is present in cells as long linear chains of nucleotides. Segments of the DNA that are called *genes* are *transcribed* into RNA, processed into RNA messages and *translated* into proteins. Most of the diverse biochemical processes inside cells are performed by complex protein machines. The activity of these machines is determined by the availability of their protein components, which is affected by the activity level (transcription and translation) of genes [2].

2.1.1 Genomes are regulated to produce functional diversity and to respond to the environment

The genome of unicellular or multicellular organisms contains millions (or billions) of nucleotides and thousands of genes. The static view of the genome as simply containing the instruction for generating proteins cannot explain the remarkable flexibility and diversity of cells. For example, yeast cells can survive on dramatically different nutritional media and can respond to a rapidly changing environment efficiently, although their genome is almost completely unchanged. An even more striking example is the developmental programs of differentiation in multi-cellular organisms. Such programs allow a single germ-line cell to develop into a complete organism, including billions of cells from numerous types, although the genome of all these cells is unchanged. The key to understanding the nature of this diversity is a battery of mechanisms and processes collectively called *genomic regulation*. Regulatory mechanisms allow cells to choose the appropriate level of activity for each of the genes in the genome. For example, neuron cells can express genes that code for neuro-transmitters and avoid expressing genes that code for lipid turnover proteins. Genomic regulation is a dynamic process even within the same cell: for example, the yeast genome can respond to a sudden heat shock by rapidly increasing the expression of proteins that identify damaged peptides and destroy them [45].

2.1.2 Transcription factors catalyze transcription initiation

Genome regulation is a complex process involving interaction between proteins and DNA and among proteins. 40 years of ingenious experiments (starting with the fundamental work by Jacob and Monod [73]) have characterized the basic building blocks of transcriptional regulatory switches. The key players in such switches are proteins of a special class, called *transcription factors* (TFs). Transcription factors are proteins that are dedicated to genomic regulation. Their biochemical function can involve binding of DNA, remodeling of chromatin in preparation to transcription, assembly of components of the RNA polymerase machinery, and much more. The DNA sequence and chromatin structure around each gene contain signals that are identifiable by some of the TFs and therefore, each genomic region may recruit a distinct combination of transcription factors, resulting in different rate of polymerase assembly and transcription. On the other hand, cells may contain many copies (dozens to thousands [53]) of a certain TF, and so copies of a single TF may interact with many genomic regions. The flexibility of transcriptional switches relies on the ability of cells to modify the level of activity of TFs. This can be done by altered transcriptional regulation of genes coding for the TFs (TFs, being proteins, are themselves encoded by genes) or by post-translational modifications that can affect the potency of the TF to bind DNA or to interact with other factors. The overall picture is believed today to be even more complex than what was thought ten years ago. Basically, there are hundreds of transcription factors that interact with each other and with other proteins and regulate thousands of genes, including genes coding for TFs. In addition, other factors (e.g., small RNA molecules [8]) and additional kinds of interactions (e.g., chromosomal localization and interaction) play an important, yet uncharacterized role in regulation. For our purpose here, we will focus on transcription factors and their regulated genes, as the characterization of the regulatory process, even when limited to these factors, is a major open challenge.

2.1.3 Gene promoters encode transcriptional switches

The ability of genomes to apply a different regulatory scheme to each gene relies on information that is encoded in the DNA sequence surrounding the gene itself. Genes consist of fragments of *coding sequence* which specify the content of the gene's end-product (the amino acid sequence in the case of protein coding genes). In close proximity to such sequences (but not necessarily in continguity to it), the genome contains *regulatory regions* that are used by the transcription control machinery to determine when and how to activate the gene. Such control regions contain short DNA sequences that are preferably bounded by *sequence specific TFs*, as well as other signals that are not completely understood today. The short DNA sequences that can be bound by TFs are called here TF binding sites (TFBSs). The composition and spatial organization of TFBSs near a gene determines, at each condition, which TFs will bind the gene's regulatory region. Biochemically, the concentration of the TF and the free energy of the binding site-TF interaction together determine the level of binding site occupancy. When a large number of possible TFBSs are present and several TFs are bound to the regulatory regions simultaneously, the reaction becomes very complex and can translate TF concentrations into transcriptional activity via rich logical functions [154]. Such functions cannot be determined directly from the sequence.

2.1.4 Sequence specific TFs recognize DNA motifs

Understanding of transcriptional switches begins in the characterization of individual TFBSs and the TF-DNA reactions on them. In theory, we would like to read the DNA sequence and compute the *affinity* of each TF to each location along the genome just as we can predict the protein sequence out of the DNA sequence of the gene coding for it. Unfortunately, predicting TF-DNA affinity is a very difficult task that cannot be accurately solved even for TFs whose structure is well understood. As a first approximation, a common simplified model represents the binding energy as the sum of individual energy contributions from the few positions that make up the binding site. According to this model, each TF has preferences for specific nucleotides at each position of the site. Given these preferences, which are summarized in a Position Weight Matrix (PWM) (also known as Position Specific Score Matrix (PSSM)), we can predict the binding affinity of candidate TFBSs along the genome. Although several studies suggest refinements of the position independence assumption [7], it is still the prevalent approach to modeling TF-DNA interactions. Importantly, PWMs are very easy to manipulate computationally and are a natural probabilistic/energetic substitute to the common combinatorial alternative that uses consensus sequences to determine which sites will be bound by a TF. We note that while binding affinity can at least be approximated from the sequence, the logic of the entire transcriptional switch is completely out of reach using sequence alone. The main reason for this limitation is the complexity of protein-protein interactions among TFs and between TFs and the chromatin, which cannot be predicted from protein sequence or even protein structure using today's methods.

2.1.5 Coordination of biological processes is facilitated by co-regulation at the transcriptional level

We have mentioned above that genomes contain thousands of genes and that each gene has a distinct transcriptional switch controlling its activation. We also stated that this switch is encoded into the sequence and structure of the genome around that gene. In order to perform complex tasks, coordinating dozens or hundreds of proteins, cells employ *transcriptional programs* that are modular and hierarchical. Such transcriptional programs are organizing a coordinated response of groups of co-regulated genes. For example, in eukaryotes, a small number of transcription factors are triggered for activation in specific phases of the cell cycle by the Cdk complex [123]. Small subsets of this set of TFs regulate the expression of hundreds of genes each, therefore ensuring that groups of genes that are required for specific activities (e.g. DNA synthesis) are expressed together. The regulatory regions of genes that participate in such transcriptional programs would therefore contain

TFBSs for specific TFs. Consequently, we should be able to associate TFBSs with specific transcriptional responses based on genomic activity profiles, as we shall see in the next section.

2.2 High throughput biology

The genomic revolution [85, 144] has changed the scale by which biological questions can be asked. Additional technological innovations have made it possible to perform experiments that capture cell activity on a genomic scale. These advances have contributed to the transformation of biology into an information science, where a large body of data is collected rapidly and needs to be analyzed and explained using mathematical models and algorithms.

2.2.1 Gene expression can be measured in a genomic scale using microarrays

The first, and currently most popular, functional genomic experimental technique emerged in the middle of the 90s and allows the measurement of gene expression on a genomic scale [116, 31]. The technology uses the complete genomic sequence of an organism and builds on the ability to miniaturize DNA hybridization experiments so that thousands such experiments can be performed on a single chip, in a single experiment. Using several technological alternatives (e.g., cDNA microarrays, oligonucleotide DNA chips) expression profiling is now a standard tool in many biological laboratories, with applications ranging from basic experiments on metabolism of yeast [100] to clinical assessment of cancer progression and prognosis [109]. Gene expression profiling comprehensively describes the "output" of transcriptional programs - the activity profiles that are produced by the regulatory network in a certain environment. Such "molecular phenotypes" are used frequently for the identification of genes that are involved in certain processes (in this context gene expression profiling is merely an efficient genetic screening procedure). As more and more expression profiles are obtained, the general organization of transcriptional programs can be revealed by comparisons of the responses to many different conditions [67, 45, 18].

2.2.2 TF genomic locations are measured by chromatin immunoprecipitation and microarrays

A more intimate view on the internals of transcriptional programs comes from combination of chromatin immunoprecipitation and DNA microarrays (ChIP on chip). Using cross-linking of TFs to DNA, immunoprecipitation and release from cross-linking, fragments of DNA that are bound by a TF in vivo can be isolated and hybridized to a microarray containing probes for all possible regulatory regions in the genome [111, 72, 122, 87]. The genomic regions where binding occur can thus be identified on a genomic scale, providing critical information on the topology of the transcriptional network. Advanced technologies are continuously improving the resolution and the magnitude by which ChIP on chip experiments are performed. Modern *tiling arrays* [19] allow millions of probes to be used and enable localization experiments to be performed in large genomes (e.g., humans).

2.2.3 Additional high throughput techniques are constantly emerging

Many other features of biological networks are subject to high throughput experimentation. Standard genetic screening, aiming at the identification of genes that are required for growth in a certain condition, is greatly accelerated by the availability of whole genome mutant libraries, by robotics and by image processing algorithms that create an efficient pipeline for the measurement of phenotypic effects of mutations in thousands of genes [54, 34]. Whole genome genetic analysis is also possible with more complex assays, where pairs of genes are tested for essentiality in a high throughput fashion [139]. A significant effort is directed toward the identification of physical interactions between proteins [142, 64, 56] on a large scale. For example, information on such interactions is revolutionizing our understanding of post-translational cascades leading to transcriptional switches.

2.2.4 Clustering algorithms identify groups of co-expressed genes

Genomic and functional genomics experiments are generating massive amounts of data. The success of experiments therefore depends not only on clever design and a polished protocol, but also on successful analysis of very large datasets. The first analytical method used to analyze a set of gene expression profiles was *clustering.* The idea is simple: instead of having to understand the behavior of thousands of individual genes, one can look at a much smaller set of *gene clusters.* A cluster is a group of genes that, at least in the context of the experiment, behave similarly. The algorithmic problem of clustering is well developed in general [61] and many algorithms were suggested for clustering biological data. Perhaps the most popular of these algorithms is hierarchical clustering [37], owing much of its success to the familiarity of its biologist users with phylogenetic trees. Other algorithmic alternatives include standard optimization approaches like k-means or self organizing maps [127], and also algorithms that were developed specifically for gene expression data, and take into account typical characteristics of these data [11, 121].

2.2.5 Motif finding algorithms identify regulatory elements in groups of co-regulated genes

Clusters of co-regulated genes, as identified by clustering of gene expression data, represent *transcriptional modules* - units of the transcriptional program in which a common regulatory mechanism drives the expression of a group of genes. As we outlined above, the transcriptional switch controlling each gene is at least partially determined by the sequence of regulatory regions around it, and more specifically by the presence of TFBSs that are targeted by specific TFs. Given a cluster of co-regulated genes, one can anticipate that common sequence motifs in the regulatory regions of genes in the cluster would carry functional importance and may correspond to the TFBSs regulating the common module's response. Indeed, numerous methods for motif finding were developed to try and identify sequence motifs in groups of regulatory regions, most of which use simplified assumptions on the structure of a motif, either being a combinatorial consensus or a PWM [138, 137, 123, 38]. In many cases, motif finding algorithms were able to characterize novel TFBSs and to reveal the structure of the transcriptional networks controlling important biological processes.

2.3 Computational models for transcriptional programs

The computational techniques we have outlined above allow the clustering of gene expression data and the discovery of enriched motifs in regulatory regions of genes in specific clusters. These methods are based on well developed computational problems (clustering and motif finding) that were adapted to the biological domain. Biologists are typically using a two-phase approach for the analysis of gene expression data, first clustering and then motif finding. A more integrative approach to the computational analysis of transcriptional programs requires new algorithms and models that are more specific to the biological domain.

2.3.1 Cluster based models for transcriptional regulation

One problem with the two-phase approach to gene expression analysis is the lack of integration between regulatory sequence data and gene expression profiles. Clustering algorithms use only the expression to determine the partition into clusters and this partition cannot be corrected given the results of the motif finding algorithm. By integrating the two problems into one, algorithms for simultaneous discovery of clustering and enriched motifs were shown to have improved performance in several cases [16, 148, 119]. The typical integrative approach simply searches for a clustering solution that is optimal with respect to a score that quantifies the clusters expression coherence and the existence of common DNA regulatory motifs. Additional integrative approaches combine gene expression data with ChIP on chip data [6, 87] or try to learn a model in which each cluster has a decision tree predicting the expression of genes in the module from the activity of a selected set of putative regulators [118].

2.3.2 Gene networks model regulation at the single gene level

Interpretation of transcriptional programs by means of modules (or clusters) allows us to learn a model for transcriptional programs using a reasonable number of parameters. The cluster-based approach also provides improved robustness for the considerable amounts of noise in gene expression data. However, the actual structure of transcriptional programs, as we outlined above, is too detailed to be fully expressed by a cluster-based model, as each gene has its own transcriptional switch. A more flexible approach for the computational learning of transcriptional programs is thus to assume the activity of each gene is an independent variable and the behavior of each variable is determined by a variable-specific regulation function. Such approaches, either deterministic [77, 90] or Bayesian [42] use established methods from computational learning theory and probabilistic graphical models, but are inherently confined by the large number of parameters that define the model, and by the relative scarcity of data for specifically determining all these parameters. One partial solution to the above problem is to report only on features of the model that can be learned robustly, using bootstrap on relations [42] or sub-networks [103].

2.3.3 Refined network models use biologically motivated constraints

A more sophisticated class of network models tries to assume additional constraints on the model so that learning would become more practical and require less parameters. Such constraints may be using other sources of data (e.g., ChIP on chip [60]) or specific structure imposed on transcriptional switches [99]. In a more complex family of models, prior knowledge on some of the regulation functions in part of the network can be assumed [48, 47], and the learning algorithms take into account a large number of hidden variables. Another important issue in network models is their temporal behavior. A steady state assumption is used frequently since correct temporal modeling would require understanding of the time scales in different regulatory interactions as well as rapid experimental sampling rate. Models that explicitly formulate the time-dependency of transcriptional regulation usually assume synchronous regulation and use dynamic versions of steady state models (as in dynamic Bayesian networks [71, 5]).

2.4 Evolution of transcriptional regulation

Biological research is conducted using a variety of model systems, from bacteria, through worms and up to mammals. Historically, selection of model organisms was guided by experimental considerations, choosing species in which it was possible to study specific processes in an efficient and affordable way. Recently, with the advent of genomic technologies and as functional genomics became an increasingly central challenge, new advantages for studying a diverse set of species emerged. Studying complex genomes is difficult since well defined functional elements (e.g., genes, TFBS) are only a tiny fraction of the genomic sequence. By analyzing a set of genomes together (using *comparative genomics*) we can recruit the rich and principled theory of evolution, and classify sequences by their evolutionary behavior (e.g., conserved/unconserved). Evolution, in this case, serves as a powerful source of "experiments" - we can look at a set of similar but not identical genomes and correlate differences in phenotype with differences in genotype.

The fundamental theory of molecular evolution [89] describes the relations between the evolutionary process at the DNA level, the fitness of individuals and the structure of the population. Much of the theoretical development and empirical studies in molecular evolution were focused on protein coding genes or completely non-functional elements like pseudo-genes or repetitive elements. Regulatory regions are different from both of these, since they are responsible for critical functions, but this function is not coded into the sequence in a dense and systematic way as codons in a protein coding gene. To fully exploit multiple genomes for the deciphering of transcriptional programs, one should adequately model the evolutionary dynamics of regulatory regions and correlate them with other data on the transcriptional programs in several species. Evolution of regulatory regions is thus an active field of research at both the experimental and theoretical levels [152].

2.4.1 Evolution of regulatory regions

Like any DNA sequence, regulatory regions evolve by a combination of two processes. First, the processes by which genetic material is transferred from generation to generation is noisy and involves *mutations*. Mutations, ignoring for now gross insertion or deletions, are commonly assumed to happen at the single nucleotide level and independently of each other, in a rate that may be a function of the lineage and the chromosomal locus. The second process affecting evolution at the molecular level is the *fixation* of mutations in the population. The fixation process, according to the widely accepted neutral theory [80], is believed to be mostly random, such that mutations that have no impact on the organism's fitness are continuously fixated in the population, in a rate that depends on the population size (or the *effective population size*, see [89]). In other cases, mutations with negative effects on fitness are being rapidly eliminated from the population in a process called *negative* or *purifying selection*. When affecting regulatory regions, mutations may a) have no effect on the transcriptional switch; b) change the binding capacity of an existing binding site, or even eliminate its functionality; c) change the binding capacity of a non functional site, thereby creating a new TFBS; d) have other, more indirect effect on the switch. The evolution of regulatory regions can be viewed as a sequence of such events, where a complex selection force, which depends on the function of the transcriptional switch and on the importance of certain TFBSs to its activity, affects the rate of evolution of different loci in the region.

2.4.2 Co-regulation and epistasis

An important question in the evolution of transcriptional programs (and of regulatory regions in particular) is the interplay between the functional interaction of genomic loci and their evolution. For example, assume a TF is regulating a gene through binding to the gene's regulatory regions. Both the protein coding sequence of the DNA binding domain of the TF and the sequence of the respective gene TFBS are evolving together. Any change in the DNA binding domain may change the fitness of the TFBS, and the effect of mutations in the TFBS's sequence on this fitness. The dependency of the evolution of TFs and their TFBSs is just one example of *epistasis* - non linear effect on fitness between pairs or among larger groups of loci [89]. Epistasis may be a major driving force in the evolution of transcriptional programs. Transcriptional networks involve numerous interactions between different factors, and the evolution of such complex entities cannot be understood without taking into account these dependencies. A transcriptional module, for example, where one or few TFs regulate a large group of genes through a set of similar TFBSs, implies considerable epistasis. Since the function of a transcriptional module requires co-regulation, an evolutionary change in the regulation of a module can either affect the TF's post-translational behavior, or otherwise involve numerous loci that have to somehow change together. In other words, the selective pressure on TFBSs of transcriptional modules should be increased because of the epistatic effects. The study of evolving transcriptional modules is just one example of the deep interaction between functional and evolutionary analysis of transcriptional programs.

Chapter 3

Biclustering

In this chapter we describe the algorithmic foundations that we shall later use to dissect large biological systems into modules. We shall employ a high level computational approach and treats the problem in the context of *biclustering*. In biclustering, one is given a large data matrix and the goal is to identify one or many submatrices with statistically surprising characteristics. In biological applications, we search for groups of genes with a common pattern of behavior across a group of conditions. We start this chapter by a survey of some of the biclustering models and algorithms that are in use today. We then outline the SAMBA graph-theoretic formalization of the biclustering problem, and present the statistical model used by SAMBA to guarantee the significance of biclusters. We describe the theory and practice of the SAMBA algorithm next, and show how SAMBA can be used to discover multiple biclusters, and how it eliminates redundancies. Biological applications of our methods will be presented in the next chapter.

Most of the results presented in this chapter were published in [134, 133, 136, 135]. The SAMBA algorithm was developed with Roded Sharan.

3.1 Introduction and motivation

Given a set of gene expression profiles, organized together as a *gene expression matrix* with rows corresponding to genes and columns corresponding to conditions (or samples), a common analysis goal is to group conditions and genes into subsets that convey biological significance. In its most common form, this task translates to the computational problem known as *clustering*. Formally, given a set of elements with a vector of attributes for each element, clustering aims to partition the elements into (possibly hierarchically ordered) disjoint sets, called clusters, so that within each set the attribute vectors are similar, while vectors of disjoint clusters are dissimilar. For example, when analyzing a gene expression matrix we may apply clustering to the genes (as elements) given the matrix rows (as attribute vectors) or cluster the conditions (as elements) given the matrix columns (as attribute vectors). Analysis via clustering makes several a-priori assumptions that may not be perfectly adequate in all circumstances. First, clustering can be applied to either genes or conditions, implicitly directing the analysis to a particular aspect of the system under study (e.g., groups of patients or groups of co-regulated genes). Second, clustering algorithms usually seek a disjoint cover of the set of elements, requiring that no gene or condition belong to more than one cluster.

The notion of a bicluster gives rise to a more flexible computational framework. A *bicluster* is defined as a submatrix spanned by a set of genes and a set of conditions (compare Figure 3.1). Alternatively, a bicluster may be defined as the corresponding gene and condition subsets. Given a gene expression matrix, we can characterize the biological phenomena it embodies by a collection of biclusters, each representing a different type of joint behavior of a set of genes in a corresponding set of conditions. Note that there are no a-priori constraints on the organization of biclusters and, in particular, genes or conditions can be part of more than one bicluster or of no bicluster. The lack of structural constraints on biclustering solutions allows greater freedom but is consequently more vulnerable to overfitting. Hence, biclustering algorithms must guarantee that the output biclusters are meaningful. This problem is particularly acute, since gene expression data typically suffer from high noise levels. DNA chips provide only rough approximation of expression levels, and are subject to errors of up to two-fold the measured value [1]. Any analysis method, and biclustering algorithms in particular, should therefore be robust enough to cope with significant levels of noise. This is usually done by an accompanying statistical model or a heuristic scoring method that define which of the many possible submatrices represent a significant biological behavior. The *biclustering problem* is to find a set of significant biclusters in a matrix.

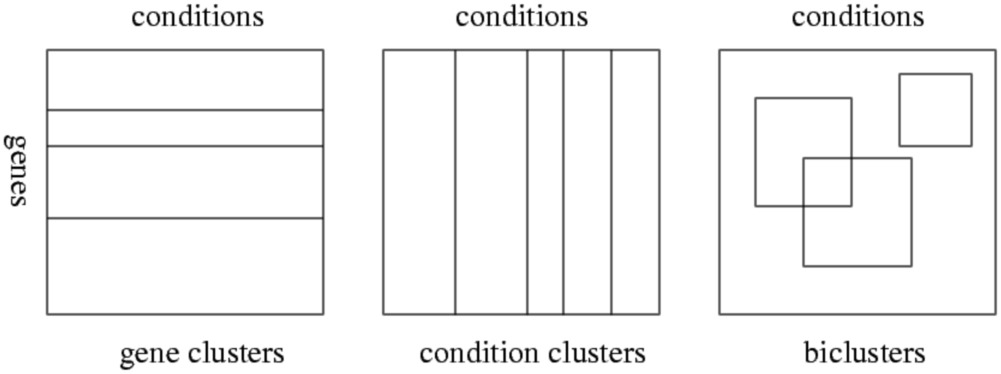


Figure 3.1: **Clustering and biclustering.** Clusters correspond to disjoint strips in the matrix. A row cluster must contain all columns, and a column cluster must contain all rows. Biclusters correspond to arbitrary subsets of rows and columns, shown here as rectangles. Note that since gene (condition) clusters are disjoint, the rows (columns) of the matrix can be reordered so that each cluster is a contiguous strip. Similar reordering of rows and columns that shows all the biclusters as rectangles is usually impossible.

3.2 Current approaches

In this section we outline some of the extant methods for biclustering biological (mostly gene expression) data. Throughout, we assume that we are given a set of genes V a set of conditions U, together with a matrix $E = (e_{vu})$ where e_{vu} is the expression level of gene v in condition u. We assume that the matrix is normalized, though some of the algorithms below perform additional normalization. A *bicluster* $B = (U', V')$ is defined by a subset of genes $V' \subset V$ and a subset of conditions (or samples) $U' \subset U$.

3.2.1 Cheng and Church's Algorithm

Cheng and Church were the first to introduce biclustering to gene expression analysis [20]. Their algorithmic framework represents the biclustering problem as an optimization problem, defining a score for each candidate bicluster and developing heuristics to solve the constrained optimization problem defined by this score function. In short, the constraints force the uniformity of the matrix, the procedure gives preference to larger submatrices and the heuristic is a relaxed greedy algorithm.

Cheng and Church implicitly assume that (gene, condition) pairs in a "good" bicluster have a constant expression level, plus possibly additive row and column-specific effects. After removing row, column and submatrix averages the residual level should be as small as possible. More formally, given the gene expression matrix E, a subset of genes I and a subset of conditions J, we define $e_{Ij} = \frac{\sum_{i \in I} e_{ij}}{|I|}$ (row subset average) $e_{iJ} = \frac{\sum_{j \in J} e_{ij}}{|J|}$ (column subset average) and $e_{IJ} = \frac{\sum_{i \in I, j \in J} e_{ij}}{|I||J|}$ (submatrix average). We define the *residue score* of an element e_{ij} in a submatrix E_{IJ} as $RS_{IJ}(i,j) = e_{ij} - e_{Ij} - e_{iJ} + e_{IJ}$ and the *mean square residue score* of the entire submatrix as $H(I,J) = \sum_{i \in I, j \in J} \frac{RS_{ij}^2}{|I||J|}$. The intuition behind this definition can be understood via two examples: a completely uniform matrix will have score zero. More generally, any submatrix in which all entries have the form $e_{ij} = b_i + c_j$ would also have score zero. Given the score definition, the *maximum bicluster problem* seeks a bicluster of maximum size among all biclusters with score not exceeding a threshold δ. The size can be defined in several ways, for example as the number of cells in the matrix ($|I||J|$) or the number of rows plus number of columns ($|I| + |J|$).

The maximum bicluster problem is NP-hard if we force all solutions to be square matrices ($|I| = |J|$) or if we use the total number of submatrix cells as our optimization goal (Reductions are from Maximum Balanced Biclique or Maximum Edge Biclique). Cheng and Church suggested a greedy heuristic to rapidly converge to a locally maximal submatrix with score smaller than the threshold. The algorithm (presented in Figure 3.2) can be viewed as a local search algorithm starting from the full matrix. Given the threshold parameter δ, the algorithm runs in two phases. In the first phase, the algorithm removes rows and columns from the full matrix. At each step, where the current submatrix has row set I and column set J, the algorithm examines the set of possible moves. For rows it calculates $d(i) = \frac{1}{|J|} \sum_{j \in J} RS_{I,J}(i,j)$ and for columns it calculates $e(j) = \frac{1}{|I|} \sum_{i \in I} RS_{I,J}(i,j)$. It then selects the highest scoring row or column and removes it from the current submatrix, as long as $H(I,J) > \delta$. The idea is that rows/columns with large contribution to the score can be removed with guaranteed improvement (decrease) in the total mean square residue score. A possible variation of this heuristic removes at each step all rows/columns with a contribution to the residue score that is higher than some threshold.

In the second phase of the algorithm, rows and columns are being added, using the same scoring scheme, but this time looking for the lowest square residues $d(i), e(j)$ at each move, and terminating where none of the possible moves increases the matrix size without crossing the threshold δ. Upon convergence, the algorithm outputs a submatrix with low mean residue and locally maximal size.

To discover more than one bicluster, Cheng and Church suggested repeated application of the biclustering algorithm on modified matrices. The modification includes randomization of the values in the cells of the previously discovered biclusters, preventing the correlative signal in them to be beneficial for any other bicluster in the matrix. This has the obvious effect of precluding the identification of biclusters with significant overlaps.

An application of the algorithm to yeast and human data is described in [20]. The software is available at http://arep.med.harvard.edu/biclustering.

3.2.2 Coupled Two-way Clustering

Coupled two-way clustering (CTWC), introduced by Getz, Levine and Domany [51], defines a generic scheme for transforming a one-dimensional clustering algorithm into a biclustering algorithm. The algorithm relies on having a one-dimensional (standard) clustering algorithm that can discover significant (termed *stable* in [51]) clusters. Given such an algorithm, the coupled two-way clustering procedure will recursively apply the one-dimensional algorithm to submatrices, aiming to find subsets of genes giving rise to significant clusters of conditions and subsets of conditions giving rise to significant gene clusters. The submatrices defined by such pairings are called *stable submatrices* and correspond to biclusters. The algorithm, which is shown in Figure 3.3, operates on a set of gene subsets $\mathcal{V}$ and a set of condition subsets $\mathcal{U}$. Initially $\mathcal{V} = \{V\}$ and $\mathcal{U} = \{U\}$. The algorithm then iteratively selects a gene subset $V' \in \mathcal{V}$ and a condition subset $U' \in \mathcal{U}$ and applies the one dimensional clustering algorithm twice, to cluster V' and U' on the submatrix $U' \times V'$. If stable clusters are detected, their gene/condition subsets are added to the respective sets $\mathcal{V}$, $\mathcal{U}$. The process is repeated until no new stable clusters can be found. The implementation makes sure that each pair of subsets is not encountered more than once.

Note that the procedure avoids the consideration of all rows and column subsets, by starting from an established row subset when forming subclusters of established column subsets, and vice versa. The success of the coupled two-way clustering strategy depends on the performance of the given one-dimensional clustering algorithm.

```
Cheng-Church($U$, $V$, $E$, $\delta$):
$U$ : conditions. $V$ : genes.
$E$ : Gene expression matrix.
$\delta$: maximal mean square residue score.
Define $e_{Ij} = \frac{\sum_{i \in I} e_{ij}}{|I|}$
Define $e_{iJ} = \frac{\sum_{j \in J} e_{ij}}{|J|}$
Define $e_{IJ} = \frac{\sum_{i \in I, j \in J} e_{ij}}{|I||J|}$
Define $RS_{IJ}(i,j) = e_{ij} - e_{Ij} - e_{iJ} + e_{IJ}$
Define $H(I,J) = \sum_{i \in I, j \in J} \frac{RS_{ij}^2}{|I||J|}$.
Initialize a bicluster $(I,J)$ with $I = U, J = V$.
Deletion phase:
    While ($H(I,J) > \delta$) do
        Compute for $i \in I$, $d(i) = \frac{1}{|J|} \sum_{j \in J} RS_{I,J}(i,j)$.
        Compute for $j \in J$, $e(j) = \frac{1}{|I|} \sum_{i \in I} RS_{I,J}(i,j)$.
        If $max_{i \in I} d(i) > max_{j \in J} e(j)$ assign $I = I \setminus \{argmax_i(d(i))\}$.
        Else $J = J \setminus \{argmax_j(e(j))\}$
Addition phase:
    assign $I' = I, J' = J$
    While ($H(I',J') < \delta$) do
        Assign $I = I', J = J'$
        Compute for $i \in U \setminus I$, $d(i) = \frac{1}{|J|} \sum_{j \in J} RS_{I,J}(i,j)$.
        Compute for $j \in V \setminus J$, $e(j) = \frac{1}{|I|} \sum_{i \in I} RS_{I,J}(i,j)$.
        If $max_{i \in I} d(i) < max_{j \in J} e(j)$ assign $I' = I \cup \{argmax_i(d(i))\}$.
        Else $J' = J \cup \{argmax_j(e(j))\}$
Report $I, J$
```

Figure 3.2: The Cheng-Church algorithm for finding a single bicluster.

We note that many popular clustering algorithms (e.g. K-means, Hierarchical, SOM) cannot be plugged "as is" into the coupled two-way machinery, as they do not readily distinguish significant clusters from non-significant clusters or make a-priori assumption on the number of clusters. Getz et al. have reported good results using the SPC hierarchical clustering algorithm [52]. The results of the algorithm can be viewed in a hierarchical form: each stable gene (condition) cluster is generated given a condition (resp. gene) subset. This hierarchical relation is important when trying to understand the context of joint genes or conditions behavior. For example, when analyzing clinical data, Getz et al. have focused on gene subsets giving rise to stable tissue clusters that are correlative to known clinical attributes. Such gene sets may have an important biological role in the disease under study.

The CTWC algorithm has been applied to a variety of clinical data sets (see, e.g., [114]), the software can be downloaded via the site http://ctwc.weizmann.ac.il.

```
TWOWAY($U$, $V$, $E$, $ALG$):
$U$ : conditions. $V$ : genes.
$E$ : Gene expression matrix.
$ALG$ : one-dimensional clustering algorithm. Inputs a matrix and outputs significant (stable) clusters of columns or rows
Initialize a hash table $weight$
Initialize $\mathcal{U}_1 = \{U\}$, $\mathcal{V}_1 = \{V\}$
Initialize $\mathcal{U} = \emptyset$, $\mathcal{V} = \emptyset$
Initialize the sets hierarchy table $H_V$ storing for gene clusters
the condition subsets used to generate them.
Initialize the sets hierarchy table $H_U$ storing for condition
clusters the gene subsets used to generate them.
While ($\mathcal{U}_1 \neq \emptyset$ or $\mathcal{V}_1 \neq \emptyset$) do
        Initialize empty sets $\mathcal{U}_2, \mathcal{V}_2$.
        For all $(U', V') \in (\mathcal{U}_1 \times \mathcal{V}_1) \cup (\mathcal{U}_1 \times \mathcal{V}) \cup (\mathcal{U} \times \mathcal{V}_1)$ do
            Run $ALG(E_{U'V'})$ to cluster the genes in $V'$:
                Add the stable gene sets to $\mathcal{V}_2$
                Set $H_V[V''] = U'$ for all new clusters $V''$.
            Run $ALG(E_{U'V'})$ to cluster the conditions in $U'$:
                Add the stable condition sets to $\mathcal{U}_2$
                Set $H_U[U''] = V'$ for all new clusters $U''$.
        Assign $\mathcal{U} = \mathcal{U} \cup \mathcal{U}_1$, $\mathcal{V} = \mathcal{V} \cup \mathcal{V}_1$
        Assign $\mathcal{U}_1 = \mathcal{U}_2$, $\mathcal{V}_1 = \mathcal{V}_2$
Report $\mathcal{U}, \mathcal{V}$ and their hierarchies $H_U, H_V$.
```

Figure 3.3: Coupled two-way clustering.

3.2.3 The Iterative Signature Algorithm

In the Iterative Signature Algorithm (ISA) [70, 13] the notion of a significant bicluster is defined intrinsically on the bicluster genes and samples – the samples of a bicluster uniquely define the genes and vice versa. The intuition is that the genes in a bicluster are co-regulated and, thus, for each sample the average gene expression over all the bicluster's genes should be surprising (unusually high or low) and for each gene the average gene expression over all biclusters samples should be surprising. This intuition is formalized using a simple linear model for gene expression assuming normally distributed expression levels for each gene or sample as shown below.

The algorithm, presented in Figure 3.4, uses two normalized copies of the original gene expression matrix. The matrix E^G has rows normalized to mean 0 and variance 1 and the matrix E^C has columns normalized similarly. We denote by $e^G_{V'u}$ the mean expression of genes from V' in the sample u and by $e^C_{vU'}$ the mean expression of the gene v in samples from U'. A bicluster $B = (U', V')$ is required to have:

$$U' = \{u \in U :\ |e^G_{V'u}| > T_C\sigma_C\}, V' = \{v \in V :\ |e^C_{vU'}| > T_G\sigma_G\} \tag{3.1}$$

Here T_G is the threshold parameter and σ_G is the standard deviation of the means $e^C_{vU'}$ where v ranges over all possible genes and U' is fixed. Similarly, T_C, σ_C are the corresponding parameters for the column set V'. The idea is that if the genes in V' are up- or down-regulated in the conditions U' then their average expression should be significantly far (i.e., T_G standard deviations) from its expected value on random matrices (which is 0 since the matrix is standardized). A similar argument holds for the conditions in U'. The standard deviations can be predicted as $\frac{1}{\sqrt{|U'|}}, \frac{1}{\sqrt{|V'|}}$ being a linear sum of $|U'|$ (or $|V'|$) independent standard random variables. Alternatively (and in fact, more practically), the standard deviations can be estimated directly from the data, correcting for possible biases in the statistics of the specific condition and gene sets used. In other words, in a bicluster, the z-score of each gene, measured w.r.t. the bicluster's samples, and the z-score of each sample, measured w.r.t. the bicluster's samples, should exceed a threshold. As we shall see below, ISA will not discover biclusters for which the conditions (3.1) hold strictly, but will use a relaxed version.

The algorithm starts from an arbitrary set of genes $V_0 = V_{in}$. The set may be randomly generated or selected based on some prior knowledge. The algorithm then repeatedly applies the update equations:

$$U_i = \{u \in U : \ |e^G_{V_i u}| > T_C \sigma_C\}, V_{i+1} = \{v \in V : \ |e^C_{v U_i}| > T_G \sigma_G\} \qquad (3.2)$$

The iterations are terminated at step n satisfying:

$$\frac{|V_{n-i} \setminus V_{n-i-1}|}{|V_{n-i} \cup V_{n-i-1}|} < \epsilon \qquad (3.3)$$

for all i smaller than some m. The ISA thus converges to an approximated fixed point that is considered to be a bicluster. The actual fixed point depends on both the initial set V_{in} and the threshold parameters T_C, T_G. To generate a representative set of biclusters, it is possible to run ISA with many different initial conditions, including known sets of associated genes or random sets, and to vary the thresholds. After eliminating redundancies (fixed points that were encountered several times), the set of fixed points can be analyzed as a set of biclusters.

The ISA algorithm can be generalized by assigning weights for each gene/sample such that genes/samples with a significant behavior (higher z-score) will have larger weights. In this case, the simple means used in (3.1) and (3.2) are replaced by weighted means and the algorithm can be represented using matrix operations.

The signature algorithm has been applied for finding cis-regulatory modules in yeast ([70]) and for detecting conserved transcriptional modules across several species ([14]). For software see http://barkai-serv.weizmann.ac.il/GroupPage/.

```
ISA($U$, $V$, $E$, $V_{in}$, $T_G$, $T_C$, $m$, $\epsilon$):
$U$ : conditions. $V$ : genes.
$E$ : Gene expression matrix.
$V_{in}$ : Initial gene set.
$T_G, T_C$: gene and condition $z$-score thresholds.
$m, \epsilon$: stopping criteria.
Construct a column standardized matrix $E^C$.
Construct a row standardized matrix $E^G$.
Initialize counters $n = 0, n' = 0$.
Initialize the current genes set $V' = V_{in}$
Initialize an empty condition set $U'$.
**While** $(n - n' < m)$ do
        Compute $e^G_{V'u} = \frac{1}{|V'|} \sum_{v \in V'} e^G_{vu}$ for $u \in U$.
        $U' = \{u \in U : \ |e^G_{V'u}| > \frac{T_C}{\sqrt{|V'|}}\}$
        Compute $e^C_{vU'} = \frac{1}{|U'|} \sum_{u \in U'} e^C_{vu}$ for $v \in V$.
        $V'' = V'$
        $V' = \{v \in V : \ |e^C_{vU'}| > \frac{T_G}{\sqrt{U'}}\}$
        if $(\frac{|V' \backslash V''|}{|V' \cup V''|} < \epsilon)$ then $n' = n$
        $n = n + 1$
Report $U', V'$
```

Figure 3.4: **The ISA algorithm for finding a single bicluster.** For simplicity we write the algorithm here as if $\sigma_G = \frac{1}{\sqrt{|U'|}}$ and $\sigma_C = \frac{1}{\sqrt{|V'|}}$, direct estimations of the standard deviation may give better results in practice (see text).

3.2.4 Spectral Biclustering

Spectral biclustering uses techniques from linear algebra to identify bicluster structures in the input data. Here we review the biclustering technique presented in Kluger et al. [83]. In this model, it is assumed that the expression matrix has a hidden checkerboard-like structure that we try to identify using eigenvector computations. The structure assumption is argued to hold for clinical data, where tissues cluster to cancer types and genes cluster to groups, each distinguishing a particular tissue type from the other types.

To describe the algorithm, suppose at first that the matrix E has a checkerboard-like structure (see Figure 3.5). Obviously we could discover it directly, but we could also infer it using a technique from linear algebra that will be useful in case the structure is hidden due to row and column shuffling. The technique is based on a relation between the block structure of E and the block structure of pairs of eigenvectors for EE^T and E^TE, which we describe next. First, observe that the eigenvalues of EE^T and E^TE are the same. Now, consider a vector x that is *stepwise*, i.e., piecewise constant, and whose block structure matches that of the rows of E. Applying E to x we get a stepwise vector y. If we now apply E^T to y we get a vector with the same block structure as x. The same relation is observed when applying first E^T and then E (see Figure 3.5). Hence, vectors of the stepwise pattern of x form a subspace that is closed under E^TE. This subspace is spanned by eigenvectors of this matrix. Similarly, eigenvectors of EE^T span the subspace formed by vectors of the form of y. More importantly, taking now x to be an eigenvector of E^TE with an eigenvalue λ, we observe that $y = Ex$ is an eigenvector of EE^T with the same eigenvalue.

$$Ex = \begin{bmatrix} 8 & 8 & 7 & 7 & 3 & 3 \\ 8 & 8 & 7 & 7 & 3 & 3 \\ 6 & 6 & 4 & 4 & 5 & 5 \\ 6 & 6 & 4 & 4 & 5 & 5 \end{bmatrix} \begin{bmatrix} a \\ a \\ b \\ b \\ c \\ c \end{bmatrix} = \begin{bmatrix} d \\ d \\ e \\ e \end{bmatrix} = y, E^Ty = \begin{bmatrix} 8 & 8 & 6 & 6 \\ 8 & 8 & 6 & 6 \\ 7 & 7 & 4 & 4 \\ 7 & 7 & 4 & 4 \\ 3 & 3 & 5 & 5 \\ 3 & 3 & 5 & 5 \end{bmatrix} \begin{bmatrix} d \\ d \\ e \\ e \end{bmatrix} = \begin{bmatrix} a' \\ a' \\ b' \\ b' \\ c' \\ c' \end{bmatrix} = x'$$

Figure 3.5: An example of a checkerboard-like matrix E and the eigenvectors of EE^T and E^TE. The vector x satisfies the relation $E^TEx = E^Ty = x' = \lambda x$. Similarly, y satisfies the equation $EE^Ty = E\lambda x = \lambda y$.

In conclusion, the checkerboard-like structure of E is reflected in the stepwise structures of pairs of EE^T and E^TE eigenvectors that correspond to the same eigenvalue. One can find these eigenvector pairs by computing a singular value decomposition of E. Singular value decomposition is a standard algebraic technique (cf. [106]) that expresses a real matrix E as a product $E = A\Delta B^T$, where Δ is a diagonal matrix and A and B are orthonormal matrices. The columns of A and B are the eigenvectors of EE^T and E^TE, respectively. The entries of Δ are square roots of the corresponding eigenvalues, sorted in a non-increasing order. Hence the eigenvector pairs are obtained by taking for each i the ith columns of A and B, and the corresponding eigenvalue is the Δ_{ii}^2.

For any eigenvector pair, one can check whether each of the vectors can be approximated using a piecewise constant vector. Kluger et al. use a one-dimensional k-means algorithm to test this fit. The block structures of the eigenvectors indicate the block structures of the rows and columns of E.

In the general case, the rows and columns of E are ordered arbitrarily, and the checkerboard-like structure, if E has one, is hidden. To reveal such structure one computes the singular value decomposition of E and analyzes the eigenvectors of EE^T and E^TE. A hidden checkboard structure will manifest itself by the existence of a pair of eigenvectors (one for each matrix) with the same eigenvalue, that are approximately piecewise constant. One can determine if this is the case by sorting the vectors or by clustering their values, as done in [83].

Kluger et al. further discuss the problem of normalizing the gene expression matrix to reveal checkerboard structures that are obscured, e.g., due to differences in the mean expression levels of genes or conditions. The assumed model for the data is a multiplicative model, in which the expression level of a gene i in a condition j is its base level times a gene term, which corresponds to the gene's tendency of expression under different conditions, times a condition term, that represents the tendency of genes to be expressed under condition j. The normalization is done using two normalizing matrices: R, a diagonal matrix with the mean of row i at the ith position; and C, a diagonal matrix with the mean of column j at the jth position. The block structure of E is now reflected in the stepwise structure of pairs of eigenvectors with the same eigenvalue of the normalized matrices $M = R^{-1}EC^{-1}E^T$ and M^T. These eigenvector pairs can be deduced by computing a singular value decomposition of $R^{-1/2}EC^{-1/2}$. Due to the normalization, the first eigenvector pair (corresponding to an eigenvalue of 1) is constant and can be discarded. A summary of the biclustering algorithm is given in Figure 3.6.

The spectral algorithm was applied to human cancer data and its results were used for classification of tumor type and identification of marker genes [83].

3.2.5 Plaid Models

The Plaid model [86] is a statistically inspired modeling approach developed by Lazzeroni and Owen for the analysis of gene expression data. The basic idea is to represent the genes-conditions matrix as a superposition of *layers*, corresponding to

```
Spectral($U$, $V$, $E$):
$U$ : conditions. $V$ : genes.
$E_{n\times m}$ : Gene expression matrix.
Compute $R = diag(E \cdot 1_m)$ and $C = diag(1_n^T \cdot E)$.
Compute a singular value decomposition of $R^{-1/2}EC^{-1/2}$.
Discard the pair of eigenvectors corresponding to the largest eigenvalue.
For each pair of eigenvectors $u, v$ of $R^{-1}EC^{-1}E^T$ and $C^{-1}E^TR^{-1}E$
with the same eigenvalue do:
    Apply $k$-means to check the fit of $u$ and $v$ to stepwise vectors.
Report the block structure of the p $u, v$ with the best stepwise fit.
```

Figure 3.6: **The spectral biclustering algorithm.**

biclusters in our terminology, where each layer is a subset of rows and columns on which a particular set of values takes place. Different values in the expression matrix are thought of as different colors, as in (false colored) “heat maps” of chips. This metaphor also leads to referring to “color intensity” in lieu of “expression level”. The horizontal and vertical color lines in the matrix corresponding to a layer give the method its name.

The model assumes that the level of matrix entries is the sum of a uniform background (“grey”) and of k biclusters each coloring a particular submatrix in a certain way. More precisely, the expression matrix is represented as

$$A_{ij} = \mu_0 + \sum_{k=1}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}$$

where μ_0 is a general matrix background color, and $\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$ where μ_k describes the added background color in bicluster k, α and β are row and column specific additive constants in bicluster k. $\rho_{ik} \in \{0, 1\}$ is a gene-bicluster membership indicator variable, i.e., $\rho_{ik} = 1$ iff gene i belongs to the gene set of the k-th bicluster. Similarly, $\kappa_{jk} \in \{0, 1\}$ is a sample-bicluster membership indicator variable. Hence, similar to Cheng and Church [20], a bicluster is assumed to be the sum of bicluster background level plus row-specific and column-specific constants.

When the biclusters form a k-partition of the genes and a corresponding k-partition of the samples, the *disjointness constraints* that biclusters cannot overlap can be formulated as $\sum_k \kappa_{jk} \leq 1$ for all j, $\sum_k \rho_{ik} \leq 1$ for all i. Replacing $\leq$ by $=$

would require assignment of each row or column to *exactly* one bicluster. Generalizing to allow bicluster overlap simply means removing the disjointness constraints.

The general biclustering problem is now formulated as finding parameter values so that the resulting matrix would fit the original data as much as possible. Formally, the problem is minimizing

$$\sum_{ij}[A_{ij} - \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}]^2 \tag{3.4}$$

where $\mu_0 = \theta_{ij0}$. If α_{ik} or β_{jk} are used, then the constraints $\sum_i \rho_{ik}\alpha_{ik} = 0$ or $\sum_j \kappa_{jk}\beta_{jk} = 0$ are added to reduce the number of parameters. Note that the number of parameters is at most $k+1+kn+km$ for the θ variables, and $kn+km$ for the κ and ρ variables. This is substantially smaller than the nm variables in the original data, if $k << max(n,m)$.

Estimating Parameters

Lazzeroni and Owen propose to solve problem (3.4) using an iterative heuristic. New layers are added to the model one at a time. Suppose we have fixed the first $K-1$ layers and we are seeking for the K-th layer to minimize the sum of squared errors. Let

$$Z_{ij}^{(K-1)} = A_{ij} - \sum_{k=0}^{K-1} \theta_{ijk}\rho_{ij}\kappa_{jk} \tag{3.5}$$

be the *residual matrix* after removing the effect of the first $K-1$ layers. In iteration K we wish to solve the following quadratic integer program.

$$\begin{array}{ll} min & Q^{(K)} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(Z_{ij}^{(K-1)} - \theta_{ijK}\rho_{iK}\kappa_{jK})^2 \\ s.t. & \sum_i \rho_{iK}^2\alpha_{iK} = 0, \;\; \sum_j \kappa_{jK}^2\beta_{jK} = 0 \\ & \rho_{iK} \in \{0,1\}, \;\; \kappa_{jK} \in \{0,1\} \end{array} \tag{3.6}$$

The proposed heuristic method to solve (3.6) is again iterative. To avoid confusion we call the iterations for fixed K *cycles*, and indicate the cycle number by a superscript in parentheses, e.g. $\theta^{(i)}$. The integrality constraints are ignored throughout, and the goal is to solve corresponding relaxation of it. A cycle is done as follows: Compute the best values of the θ parameters given fixed ρ and κ values; Compute the best values of the ρ parameters given new θ and the old κ values; Compute the best values of the κ parameters given the new θ and the old ρ values. In order to avoid "locking in" of the membership variables to 0 or 1, their values are changed only modestly on the first cycle, and they are allowed to become integral only at the final cycle.

The following optimal parameter values in the relaxed version of (3.6) are obtained by using Lagrange multipliers:

$$\mu_K = \frac{\sum_i \sum_j \rho_{iK}\kappa_{jK}Z_{ij}^{K-1}}{(\sum_i \rho_{iK}^2)(\sum_j \kappa_{jK}^2)} \tag{3.7}$$

$$\alpha_{iK} = \frac{\sum_j (Z_{ij}^{(K-1)} - \mu_K \rho_{iK}\kappa_{jK})\kappa_{jK}}{\rho_{iK}\sum_{jK}\kappa_{jK}^2} \tag{3.8}$$

$$\beta_{jK} = \frac{\sum_i (Z_{ij}^{(K-1)} - \mu_K \rho_{iK}\kappa_{jK})\rho_{iK}}{\kappa_{jK}\sum_{iK}\rho_{iK}^2} \tag{3.9}$$

So, in cycle s, we use these equations to update $\theta^{(s)}$ using the old values $\rho^{(s-1)}$ and $\kappa^{(s-1)}$. The values for ρ_{iK} and κ_{jK} that minimize Q are:

$$\rho_{iK} = \frac{\sum_j \theta_{ijK}\kappa_{jK}Z_{ij}^{K-1}}{\sum_j \theta_{ijK}^2\kappa_{jK}^2} \tag{3.10}$$

$$\kappa_{jK} = \frac{\sum_i \theta_{ijK}\rho_{iK}Z_{ij}^{K-1}}{\sum_i \theta_{ijK}^2\rho_{iK}^2} \tag{3.11}$$

At cycle s, we use these equations to update $\rho^{(s)}$ from $\theta^{(s)}$ and $\kappa^{(s-1)}$, and update $\kappa^{(s)}$ from $\theta^{(s)}$ and $\rho^{(s-1)}$. The complete updating process is repeated a prescribed number of cycles.

Initialization and Stopping Rule

The search for a new layer K in the residual matrix $Z_{ij} = Z_{ij}^{(K)}$ requires initial values of ρ and κ. These values are obtained by finding vectors u and v and a real value λ so that λuv^T is the best rank one approximation of Z. We refer the readers to the original paper for details.

Intuitively, each iteration "peels off" another signal layer, and one should stop after $K-1$ iterations if the residual matrix $Z_{ij} = Z_{ij}^{(K)}$ contains almost only noise. Lazzeroni and Owen define the *importance* of layer k by $\sigma_k^2 = \sum_{i=1}^{n}\sum_{j=1}^{p}\rho_{ik}\kappa_{jk}\theta_{ijk}^2$. The algorithm accepts a layer if it has significantly larger importance than in noise.

To evaluate σ_k^2 on noise, repeat the following process T times: Randomly permute each row in Z independently, and then randomly permute each column in the resulting matrix independently. Apply the layer-finding algorithm on the resulting matrix, and compute the importance of that layer. If σ_k^2 exceeds the importance obtained for all the T randomized matrices, add the new layer K to the model.

The complete algorithm is outlined in Figure 3.7.

Plaid models have been applied to yeast gene expression data [86]. The software is available at http://www-stat.stanford.edu/∼owen/plaid.

```
Plaid($U$, $V$, $E$, $S$):
$U$ : conditions. $V$ : genes.
$E$ : Gene expression matrix.
$S$: maximum cycles per iteration.
Set $K = 0$
adding a new layer:
    K=K+1
    Compute initial values of $\kappa_{jK}^{(0)}, \rho_{iK}^{(0)}$. Set $s = 1$
    **While** ($s \leq S$) do:
            Compute $\mu_K^{(s)}$, $\alpha_{iK}^{(s)}$, $\beta_{jK}^{(s)}$ using equations (3.7)- (3.9).
            Compute $\kappa_K^{(s)}$ using equations (3.11)
            Compute $\rho_K^{(s)}$ using equations (3.10)
            If $\rho_K^{(s)} > 0.5$ set $\rho_K^{(s)} = 0.5 + s/2S$, else set $\rho_K^{(s)} = 0.5 - s/2S$
            If $\kappa_K^{(s)} > 0.5$ set $\kappa_K^{(s)} = 0.5 + s/2S$, else set $\kappa_K^{(s)} = 0.5 - s/2S$
    If the importance of layer $K$ is non random then record the layer and repeat
    Else exit.
Report layers $1, \ldots, K - 1$.
```

Figure 3.7: **Inferring plaid models.**

3.3 Functional properties

We start our description of the SAMBA biclustering algorithm by introducing an approach for the abstraction of genomic information. The biclustering algorithms we described above assume that the analysis is focused on gene expression data.

In practice, however, it is often desirable to apply biclustering on a heterogeneous dataset, so that unexpected dependencies among experiments can be discovered. Data from many types of biological genome-wide experiments can be expressed as vectors of real-valued measurements (one value per gene). Depending on the experiment, the data entries may vary in range, order, statistical properties and reliability. Aiming at the application of combinatorial algorithms and analysis, the SAMBA framework abstracts all sources of information as discrete *properties*. Properties are Boolean characters of genes and may be used to express any kind of biological knowledge. Data from biological experiments are transformed into probabilities of genes having certain properties. For example, given a gene expression experiment we can define two properties, one for genes that are induced in the experiment and the other for genes that are repressed in the experiment. For each gene, we can use the microarray measurement to determine what is the probability that the gene was induced or repressed at the experiment (the third possibility, and usually the most probable, that the gene is neither induced nor repressed, is not defined as a property to keep the representation sparse). We note that the transformation from continuous measurements to discrete properties needs to be determined based on prior understanding of the biological phenomenon and that this transformation may degrade some of the information of the experiment. However, the SAMBA model (see below), guarantees that whatever the discretization scheme was, the biclusters that are detected will be significantly non random. In the following we shall assume that we are given a set of genes V, a set of properties U and a matrix of probabilities $\phi(u, v)$ specifying the probability of each gene to have each of the properties. We shall describe how we derive such probabilities from various types of experiments in practice when discussing applications in the next chapter (see e.g., Figure 4.1).

3.4 The SAMBA model

Given a set of genes and properties we form a bipartite graph $G = (U, V, A)$ (See Figure 3.8 for an example). In this graph, U is the set of properties, V is the set of genes, and $(u, v) \in A$ iff v have property u with nonzero probability. We also assign each edge with the probability of the respective gene to have the respective property, denoted as $\phi(u, v)$. A bicluster corresponds to a subgraph $H = (U', V', A')$ of G, and represents a subset V' of genes that share a set of properties U' (see Figure 3.8).

Later we shall define weights (not to be confused with the probabilities) for each property-gene pair in the graph, the *weight* of a subgraph (or bicluster) will then be the sum of the weights of gene-condition pairs in it, including edges and non-edges.

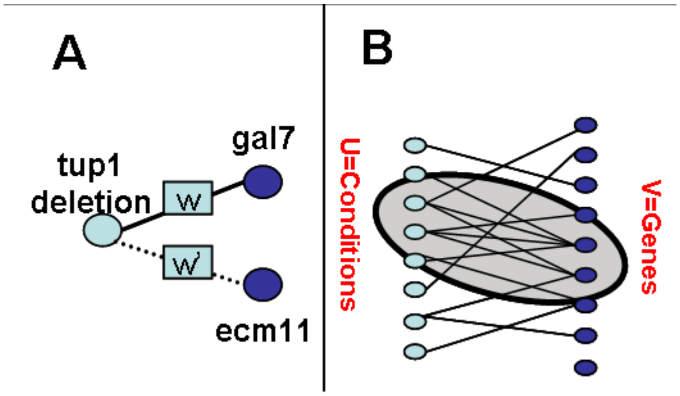


Figure 3.8: **The SAMBA graph**. Functional genomic data is modeled using a bipartite graph whose two sides correspond to a set of properties U and a set of genes V. An edge (u, v) indicates the association between gene v and property u. A statistical model assigns weights to the edges and non-edges of the graph. A) Part of the graph showing the gene expression property "induced in tup1 deletion" and its effect on the genes "gal7" (response) and "ecm11" (no response). B) A heavy subgraph (shaded) representing a significant bicluster.

In the following we develop statistical models for our bipartite graph representation of functional genomic datasets. Using these models we derive scoring schemes for assessing the significance of an observed subgraph. We note that we are aiming at scoring schemes that are *additive* - scores that can be decomposed across the edges and non-edges of the graph. This will allow us to reduce the biclustering problem to that of finding heavy subgraphs in a bipartite graph. We will first assume that all edges have probability 1 ("hard discretization") and will later explain how to generalize the model for arbitrary probabilities.

3.4.1 A Naive Model

Let $H = (U', V', A')$ be a subgraph of G. Denote $|U'| = m'$, $|V'| = n'$. Let $p = \frac{|A|}{|U||V|}$, and let $k' = |A'|$. Our first model assumes that edges occur independently and equiprobably with density p. Denote by $BT(k, p, n)$ the binomial tail, i.e., the probability of observing k or more successes in n trials, where each success occurs independently with probability p. Then the probability of observing a graph at least as dense as H according to this model is $p(H) = BT(k', p, n'm')$.

Our goal is to find a subgraph H with lowest $p(H)$. By bounding the terms of the binomial tail using the first one, assuming that $p < 1/2$, we obtain the following loose upper bound for $p(H)$: $p^(H) < \sum_i \binom{n'm'}{i} p^{k'}(1-p)^{n'm'-k'} < 2^{n'm'} p^{k'}(1-p)^{n'm'-k'} = p^*(H)$. Seeking a subgraph H minimizing $\log p^*(H)$ is equivalent to finding a maximum weight subgraph of G where each edge has positive weight $(-1 - \log p)$ and each non-edge has negative weight $(-1 - \log(1-p))$.

Note that $p(H)$ is a reasonable measure for the p-value of a subgraph only if $n'm' \ll nm$, as its calculation ignores the total number of edges in G. An accurate p-value for H is:

$$p'(H) = \frac{1}{\binom{nm}{k}} \sum_{i=k'}^{k} \binom{n'm'}{i} \binom{nm - n'm'}{k - i}$$

3.4.2 A degree based model

The simple model presented above, assume all edges to be equally probable. Unfortunately, in real biological datasets this assumption does not hold. Experiments can be relating to few genes or thousands of genes, leading to highly heterogeneous degree distribution for property nodes in our graph. The same is correct for genes - some genes are highly active and participate in numerous biological processes, other are extremely specific and can be associated with few (or no) properties.

We next develop a refined null model that takes into account the variability of the degrees in G, i.e., it incorporates the characteristic behavior of each specific condition and gene. Let $H = (U', V', A')$ be a subgraph of G and denote $\overline{A'} = (U' \times V') \setminus A'$. For a vertex $w \in U' \cup V'$ let d_w denote its degree in G. Our null model assumes that the occurrence of each edge (u, v) is an independent Bernoulli variable with parameter $p_{u,v}$. The probability $p_{u,v}$ is the fraction of bipartite graphs with degree sequence identical to G that contain the edge (u, v). This probability takes into account the variability (or degree) of the associated property and associated gene, and will typically match the statistical properties of biological data much better than the simpler model presented above. In order to compute $p_{u.v}$ we can use a Monet-Carlo simulation and sample a set of graphs with the required degree sequence. The

MC process proceeds in a sequence of edge-crossings, swapping between randomly chosen edges $(u_1, v_1), (u_2, v_2)$ and the corresponding non edges (u_1, v_2) and (u_2, v_1) (verifying that the latter are indeed non edges before the swap).

The likelihood of a subgraph H given the null model is thus $p(H) = (\prod_{(u,v)\in A'} p_{u,v}) \cdot (\prod_{(u,v)\in \overline{A'}}(1 - p_{u,v}))$. To score a subgraph for significance, we shall quantify its deviation from random behavior by comparing its likelihood according to the null model to its likelihood according to the true-bicluster model we define next. We assume that in a true bicluster, all node pairs should be edges with a constant probability p_c, and that $p_c > \max_{(u,v)\in U\times V} p_{u,v}$. This model reflects our belief that biclusters represent approximately uniform relations between their elements. The likelihood ratio for H is therefore:

$$L(H) = (\prod_{(u,v)\in A'} \frac{p_c}{p_{u,v}}) \cdot (\prod_{(u,v)\notin A'} \frac{1-p_c}{1-p_{u,v}})$$

The null model is rejected for high values of this ratio.

After taking the logarithm we get:

$$\log L(H) = \sum_{(u,v)\in A'} \log \frac{p_c}{p_{u,v}} + \sum_{(u,v)\in \overline{A'}} \log \frac{1-p_c}{1-p_{u,v}} \tag{3.12}$$

Setting the weight of each edge (u, v) to $\log \frac{p_c}{p_{u,v}} > 0$ and the weight of each non-edge (u, v) to $\log \frac{1-p_c}{1-p_{u,v}} < 0$, we derive an additive score for H, which as we stated above, is used to reduce the biclustering problem to the problem of finding the heaviest subgraph in G.

3.4.3 Incorporating edge probabilities

As we explained above, SAMBA transforms continuous experimental data into probabilistic observations on Boolean properties. The SAMBA graph is therefore a bipartite graph with edge probabilities. Recall that $\phi(u, v)$ is the probability that the edge (u, v) is present. We interpret these probabilities as defining a simple probability space over the deterministic bipartite graphs, where the probability of a graph equals the product of the probabilities for having (or not having) individual edges $Pr((U, V, A)) = \prod_{(u,v)\in A} \phi(u, v) \prod_{(u,v)\notin A}(1 - \phi(u, v))$. Given this probability space, we shall define our bicluster score as the expected log likelihood ratio $E(\log(L(H)))$ given a background model that uses the expected degrees for each node ($\hat{d_u} = \sum_{v \in V} \phi(u, v)$). Since $L(H)$ is decomposed over the edges, and using the linearity of the expectation, it is easy to see that:

$$E(\log(L(H))) = \sum_{u,v} (\phi(u,v) * \log \frac{p_c}{p_{u,v}} + (1 - \phi(u,v)) * \log \frac{1 - p_c}{1 - p_{u,v}}) \tag{3.13}$$

By setting the weight of each node pair (u, v) to $\phi(u, v) * \log \frac{p_c}{p_{u,v}} + (1 - \phi(u, v)) * \log \frac{1-p_c}{1-p_{u,v}}$ we can reduce the problem of finding subgraphs with optimal $E(\log(L(H)))$ to the problem of finding heavy subgraphs. Note that for small $\phi(u, v)$ edge weights will be negative, and for large $\phi(u, v)$ edge weights will be positive.

3.5 Finding the heaviest bipartite graph

In the previous section we have given an additive scoring scheme assigning weights to edges and non-edges. Discovering the most significant biclusters in the data reduces under this scoring scheme to finding the heaviest subgraphs in the bipartite graph. The *maximal weight bipartite subgraph* problem is defined given a complete bipartite graph with weights (positive and negative). The problem is to find the subgraph with maximal total edge weight. We note this problem is NP-hard, even if the edge weights are bounded, as can be shown by a simple reduction from CLIQUE.

Theorem 3.5.1 *For a weighted complete bipartite graph G and a number k, the problem of determining if G contains a subgraph of weight at least k is NP-complete, even if edge weights are bounded to $+1, -1$ and 0.*

Proof: By reduction from CLIQUE. Let $(G = (V, A), k)$ be an instance of CLIQUE. Let $V' = \{v' | v \in V\}$, we call v, v' a *twin pair*. Build a weighted complete bipartite graph $G' = (V, V', V \times V', w)$ where $w((v, v')) = 1$ for each $v \in V$, $w((u, v')) = w(v, u') = 0$ if $(u, v) \in A$, and $w((u, v')) = -1$ otherwise. If G has a clique of size at least k then clearly G' has a subgraph of weight at least k. Conversely, if G' has a subgraph of weight k then we can construct a $k-$clique as follows. Call a subgraph *symmetric* if it for each $v \in V$, v is part of the subgraph iff its twin v' is also part of it. W.l.o.g we assume that G' is symetric, otherwise we may remove the non-symmetric nodes without decreasing the score (without the symmetric edge, the total weight contributed by a node is non-positive). Now a subgraph of weight k must contain k' twin pairs and $k' - k$ negative edges. For each negative edge (v, u'), the (negative) edge (u, v') should also be part of the subgraph (by symmetry). Thus, by removing the four related nodes v, u, v', u' from the graph we do not decrease the total graph weight and reduce the number of negative edges. This can be done until we derive a graph of weight k without negative edges - which corresponds to a k-clique in the original graph G. ■

Using the deep result on CLIQUE hardness [63] and the fact that the reduction we presented above does not change the score function we optimize (and is thus gap preserving), we conclude that the maximum bipartite subgraph is not any easier to approximate than CLIQUE:

Corollary 3.5.2 *The maximal weigth subgraph problem on complete bipartite graphs with edge weights $\{+1, -1, 0\}$ is hard to approximate within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ unless $NP = ZPP$. In other words, if there exists $\epsilon > 0$ and a polynomial algorithm that approximate the problem within $O(n^{1-\epsilon})$ then $NP = ZPP$.*

3.6 Bounded degree biclustering

In this section we will develop practical intuition into the combinatorial problem of finding the heaviest bipartite subgraph by studying the special case in which one of the sides in the graph have bounded degrees.

3.6.1 Maximum Bounded Biclique

We start by describing an $O(|V|2^d)$-time algorithm to find a maximum weight biclique in a bipartite graph whose gene vertices have d-bounded degree. This algorithm will be a key component in our more involved algorithms that follow.

Let $G = (U, V, A)$ be a bipartite graph. We say that G has *d-bounded gene side*, if every $v \in V$ has degree at most d. Let $w : U \times V \to \mathcal{R}$ be a weight function. For a pair of subsets $U' \subseteq U, V' \subseteq V$ we denote by $w(U', V')$ the weight of the subgraph induced on $U' \cup V'$, i.e., $w(U', V') = \sum_{u \in U', v \in V'} w((u, v))$. The *neighborhood* of a vertex v, denoted $N(v)$, is the set of vertices adjacent to v in G. We denote $n = |V|$

throughout.

Problem 1 (Maximum Bounded Biclique) *Given a weighted bipartite graph G with d-bounded gene side, find a maximum weight complete subgraph of G.*

Theorem 3.6.1 *The maximum bounded biclique problem can be solved in $O(n2^d)$ time and space.*

Proof: Observe that a maximum bounded biclique $H^* = (U^*, V^*, A^*)$ in G must have $|U^*| \leq d$. Figure 3.9 describes a hash-table based algorithm that for each vertex $v \in V$ scans all $O(2^d)$ subsets of its neighbors, thereby identifying the heaviest biclique. Each hash entry corresponds to a subset of conditions and records the total weight of edges from adjacent gene vertices. The iteration over subsets of $N(v)$ is done by repeatedly changing the current subset S by adding or removing a single element, updating $w(S, \{v\})$ in constant time. Hence, the algorithm spends $O(n2^d)$ time on the hashing and finding U_{best}. Computing V_{best} can be done in $O(nd)$ time, so the total running time is $O(n2^d)$. The space complexity is $O(n2^d)$ due to the hash-table. ■

```
MaxBoundBiClique($U$, $V$, $A$, $d$):
Initialize a hash table $weight$; $weight_{best} \leftarrow 0$
**For** all $v \in V$ do
    **For** all $S \subseteq N(v)$ do
        $weight[S] \leftarrow weight[S] +$
                    $\max\{0, w(S, \{v\})\}$
        If ($weight[S] > weight_{best}$)
          $U_{best} \leftarrow S$
          $weight_{best} \leftarrow weight[S]$
Compute $V_{best} = \cap_{u \in U_{best}} N(u)$
Output ($U_{best}$, $V_{best}$)
```

Figure 3.9: An algorithm for the maximum bounded biclique problem.

Note that the algorithm can be adapted to give the k condition subsets that induce solutions of highest weight in $O(n2^d \log k)$ time using a priority queue (heap) data structure.

3.6.2 Finding Heavy Subgraphs

We now look for heavy subgraphs which are not necessarily complete. We start by giving weight 1 for an edge and weight -1 for a non-edge. Formally, given a bipartite graph $G = (U, V, A)$ define a weight function $w : U \times V \rightarrow \{-1, 1\}$ such that $w((u, v)) = 1$ for $(u, v) \in A$, and $w((u, v)) = -1$ for $(u, v) \in (U \times V) \setminus A$. Consider the following problem:

Problem 2 *(Maximum Bounded Bipartite Subgraph) Given a bipartite graph G with d-bounded gene side, find a maximum weight subgraph of G.*

Lemma 3.6.2 *Let $H^* = (U^*, V^*, A^*)$ be a maximum weight subgraph of G. Then every vertex in H^* is connected to at least half the vertices on the other side of H^*.*

Proof: Follows from the choice of weights, since if a vertex $v \in V^*$ has less than $\lceil |U^*|/2 \rceil$ neighbors, then removing v from H^* will result in a heavier subgraph. The proof for $u \in U^*$ is symmetric. ■

Corollary 3.6.3 *A maximum weight subgraph of G has at most $2d$ vertices from U.*

Proof: Since the degree of every vertex in V is bounded by d, this follows from the previous lemma. ■

Lemma 3.6.4 *Let $H^* = (U^*, V^*, A^*)$ be a maximum weight subgraph of G. For each set $X \subseteq U^*$ there exists a subset $Y \subseteq X$ with $|Y| \geq \lceil |X|/2 \rceil$ such that $Y \subseteq N(v)$ for some $v \in V^*$.*

Proof: Assume there exists $X \subseteq U^*$ such that all subsets $X \cap N(v), v \in V^*$ are of size smaller than $\lceil |X|/2 \rceil$. Then the weight of the subgraph induced on $(U^* \setminus X, V^*)$ exceeds that of H^*, a contradiction. ■

Corollary 3.6.5 *Let $H^* = (U^*, V^*, A^*)$ be a maximum weight subgraph of G. Then U^* can be covered by at most $\lfloor \log(2d) \rfloor$ sets, each of which is contained in the neighborhood of some vertex in V^*.*

Proof: Denote $|U^*| = t$. By Lemma 3.6.4 there exists a subset $Y \subseteq U^*$ with $|Y| \geq \lceil t/2 \rceil$, such that $Y \subseteq N(v)$ for some $v \in V^*$. The same holds for the set $U^* \setminus Y$, and we can continue in this manner until we cover U^*. By construction we have at most $\lfloor \log t \rfloor$ sets in the cover. Since $t \leq 2d$ by Corollary 3.6.3, the result follows. ■

Corollary 3.6.5 implies an algorithm to find a maximum weight subgraph. The algorithm tests all collections of at most $\lfloor \log(2d) \rfloor$ subsets of neighborhoods of vertices in V. Since there are $O(n2^d)$ such subsets we have:

Theorem 3.6.6 *The maximum bounded bipartite subgraph problem can be solved in* $O((n2^d)^{\log(2d)})$ *time.*

A *non-redundant* subgraph is one whose weight cannot be increased by removing any vertex from it. Theorem 3.6.6 can be generalized to give the k heaviest non-redundant subgraphs in $O((n2^d)^{\log(2d)} \log k)$ time.

We now extend Theorem 3.6.6 to graphs with more general weights: Suppose that edges in G have positive weights and non-edges have negative weights. Define $r = \max_{(u,v),(u',v') \in U \times V} |\frac{w(u,v)}{w(u',v')}|$. We call r the *maximum weight ratio* in G. Similarly to Lemma 3.6.4 we can show:

Lemma 3.6.7 *Let* $H^* = (U^*, V^*, A^*)$ *be a maximum weight subgraph of* G. *For each set* $X \subseteq U^*$ *there exists a subset* $Y \subseteq X$ *with* $|Y| \geq \lceil |X|/(r+1) \rceil$ *such that* $Y \subseteq N(v)$ *for some* $v \in V^*$.

Theorem 3.6.8 *Let* G *be a bipartite graph with* d*-bounded gene side. Suppose a weight function assigns positive and negative weights to edges and non-edges, respectively, such that the maximum weight ratio is* r. *Then the* k *heaviest non-redundant subgraphs in* G *can be found in* $O((n2^d)^{\log_{(r+1)/r}(rd)} \log k)$ *time.*

3.7 Searching for multiple biclusters

So far we have discussed the problem of finding the *best* bicluster subgraph under some plausible statistical definitions. In practice, however, we are rarely interested in just the best bicluster (which is often relatively easy to detect). Instead, we wish to identify a comprehensive set of significant biclusters in an attempt to extract all non trivial phenomena in the data. We can easily modify the combinatorial algorithm described above to find for each node in the bipartite graph the k heaviest maximal subgraphs containing it. This, as well as other heuristics (see below) will produce a large set of locally optimal biclusters. To facilitate downstream analysis, it is important to select from this large repertoire a non-redundant set of biclusters.

To formalize the notion of redundancy among biclusters, we revisit the definition of the SAMBA subgraph score $L(H)$. The definition used the log likelihood ratio of a uniform probability dense random subgraph model (a bicluster model) over a degree preserving random graph model. We can score a set of biclusters in a similar fashion, applying the log likelihood ratio to all of the edges covered by the biclusters. In other words, we assume that all edges (or non edges) that are covered by at least one bicluster are originating from a bicluster model and compare this model to the background as before. The structural constraint we used when searching for a single bicluster (nodes form for a submatrix), is relaxed so that we can detect unions of submatrices. Given a set of biclusters $H_1 = (U_1, V_1, A_1), \ldots, H_n = (U_s, V_s, A_s))$, we define $A'' = \cup_i A_i$ and $\overline{A''} = \cup_i \overline{A_i}$ where $\overline{A_i}$ is as before, $\overline{A_i} = (U_i \times V_i) \setminus A_i$. The log likelihood ratio for a set of biclusters is written as:

$$\log L(H_1, \ldots, H_s) = \sum_{(u,v) \in A''} \log \frac{p_c}{p_{u,v}} + \sum_{(u,v) \in \overline{A''}} \log \frac{1 - p_c}{1 - p_{u,v}} \tag{3.14}$$

The above definition tries to quantify how well a set of biclusters models the bipartite graph. To find a non redundant set of biclusters we wish to maximizes the L score while using a reasonable number of biclusters. This is formalized by introducing a cost $cost(H_i)$ to each bicluster and maximizing:

$$\log L(H_1, \ldots, H_s) - \sum_i cost(H_i) \tag{3.15}$$

Higher biclusters costs will result in smaller number of biclusters in the optimal solution (and lower redundancy). For example, two biclusters that share a common heavy subgraph, but have low or negative contribution in their difference will not be reported together. We shall provide a probabilistic interpretation to the biclusters' costs below.

The *bicluster set optimization problem* is defined given a bipartite graph G and a set of candidate biclusters H_i with their costs. The goal is to maximize the total cost as defined above. The problem is NP hard as can be seen by a trivial reduction from set cover. We can heuristically generate a solution using a greedy algorithm: we repeatedly select the highest scoring bicluster and update the scores of all others given the edges already covered by at least one bicluster. We can continue with local improvements (adding or removing biclusters) as long as we can improve the total score.

We shall next introduce an alternative representation of the model just defined which allows for a probabilistic interpretation of the bicluster set optimization problem. We use a random variable ϵ_e for each edge in the graph and a random variable β_b for each candidate bicluster. According to our background model, each edge appears independently with probability $p_{u,v}$. If we add to the model a bicluster β_b, we change the distribution of edges that belong to it. To describe the relations between biclusters and edges variables, we construct a Bayesian network [102] by connecting bicluster variables to their respective edges, i.e. setting β_b of the bicluster $b = (U_b, V_b, A_b)$ as the parent of all edges variables $\epsilon_{u,v}$ where $u \in U_b$ and $v \in V_b$. The conditional probability distribution of an edge variable given its associated bicluster variables is defined as a noisy 'OR' function:

$$Pr(\epsilon_{u,v}|\beta_{b_1}, \ldots, \beta_{b_d}) = \begin{cases} p_c & : \quad \epsilon_{u,v} = 1 \quad \text{and} \quad \beta_{b_1} = 1 \vee \beta_{b_2} = 1 \ldots \vee \beta_{b_d} = 1 \\ 1 - p_c & : \quad \epsilon_{u,v} = 0 \quad \text{and} \quad \beta_{b_1} = 1 \vee \beta_{b_2} = 1 \ldots \vee \beta_{b_d} = 1 \\ p_{u,v} & : \quad \epsilon_{u,v} = 1 \quad \text{and} \quad \beta_{b_1} = 0 \wedge \beta_{b_2} = 0 \ldots \wedge \beta_{b_d} = 0 \\ 1 - p_{u,v} & : \quad \epsilon_{u,v} = 0 \quad \text{and} \quad \beta b_1 = 0 \wedge \beta_{b_2} = 0 \ldots \wedge \beta_{b_d} = 0 \end{cases}$$

We complete the definition by introducing prior probabilities for the β variables $Pr(\beta_b = 1) = 1/(1 + e^{cost(b)})$. We now interpret the SAMBA bipartite graph as an observation on edge variables - assigning to each edge variable a value of 0 or 1. The bicluster set optimization problem we defined above is now analogous to finding the MAP (maximum a-posteriori) configuration of the β variables in the Bayesian network over biclusters and edges:

Claim 3.7.1 *The bicluster set optimization problem is equivalent to MAP inference in the Bayesian network built over edges and bicluster variables.*

Proof: Each assignment $\overline{\beta}$ of Boolean values to the β variables gives rise to a subset of biclusters. Given a SAMBA graph (interpreted as an assignment on the edge variables) the likelihood of this assignment can be written as

$$\begin{gathered}\log Pr(\overline{\beta}|G) = \\ \textstyle\sum_{\beta_b=1} log(1/(1+e^{cost(b)})) + \sum_{\beta_b=0} log(e^{cost(b)}/(1+e^{cost(b)})) + \\ \textstyle\sum_{\boldsymbol{Pa}_{\epsilon_{u,v}} \neq 0 \wedge \epsilon_{u,v}=1} \log p_c + \\ \textstyle\sum_{\boldsymbol{Pa}_{\epsilon_{u,v}} \neq 0 \wedge \epsilon_{u,v}=0} \log(1-p_c) + \\ \textstyle\sum_{\boldsymbol{Pa}_{\epsilon_{u,v}} = 0 \wedge \epsilon_{u,v}=1} \log p_{u,v} + \\ \textstyle\sum_{\boldsymbol{Pa}_{\epsilon_{u,v}} = 0 \wedge \epsilon_{u,v}=0} \log(1-p_{u,v})\end{gathered}$$

Where the $\boldsymbol{Pa}$ notation represents the assignment of parent β variables. By subtracting from the likelihood the expression $-\sum_b log(1+e^{cost(b)}) + \sum_{\epsilon_{u,v}=1} \log p_{u,v} + \sum_{\epsilon_{u,v}=0} \log 1 - p_{u,v}$ (which is constant once we fix the observed graph G) we obtain the original definition of a bicluster set score. Finding the MAP assignment is thus equivalent to finding the highest scoring bicluster set. ■

The probabilistic representation allows interpretation of the bicluster costs as prior probabilities and naturally suggests the conditional marginal probabilities $Pr(\beta_b = 1|\beta_{b'}, G)$ as ways to quantify the notion of redundancy (is bicluster b significant given that we assume bicluster b'?). The MAP inference problem however, remains a hard optimization problem and we must resort to heuristic solutions.

3.8 SAMBA heuristic

Using the above techniques, the SAMBA practical implementation works as follows:

- Form the bipartite graph and calculate vertex pair weights by estimating the background edge probabilities in the degree preserving random graph model as described above.
- In practice, when processing thousands of experiments from many different sources, some of the properties may be similar due to errors in the construction of the compendium. Dependencies between properties are also present in the graph when generating several properties from the same experiments (see next chapter). We reduce the effect of such dependencies by computing the correlation between each pair of properties and searching for an independent

set in the graph of properties in which edges connect properties whose correlation is higher than a threshold. We find an independent set using a greedy algorithm with a short local search to improve the results. The subsequent step is performed on the subgraph induced by the reduced node set.

- Apply the hashing technique of the algorithm in Figure 3.9 to find the heaviest bicliques in the graph. In fact, we look for the k best bicliques intersecting every given condition or gene. This can be done efficiently using a standard heap data structure. Since our graph does not have bounded degrees, we ignore genes with degree exceeding some threshold D, and hash for each gene only subsets of its neighbors of size ranging from N_1 to N_2.

- For each of the bicluster seeds in the heap, perform a hill climbing procedure by iteratively applying the best modification to the bicluster (addition or deletion of a single vertex) so as to maximize the total weight. In this phase we use again the entire graph.

- Use the set of locally optimal biclusters to construct an instance of the bicluster set optimization problem and solve it using greedy local search as described in the previous section. Report the optimal set of biclusters as the final solution.

The SAMBA C++ implementation was optimized for handling large datasets. For example, using a standard PC (PIII 3.0GHz) it can process a graph of more than 15,000 properties and 6000 genes in less than an hour. SAMBA is also part of the Expander suite [120] distribution and is available for both Linux and Windows.

3.9 Approximating the bicluster p-value

In this section we shall develop an alternative formulation for computing the statistical significance of a bicluster. The method computes a "p-value" for a given bicluster B, i.e., the probability of finding at random a bicluster with at least the weight of B. We will use this method for validation, so that the algorithmic aspects of computing it would not be discussed. Let $H = (U', V', A')$ be a subgraph. Suppose at first that U' is fixed, and we wish to compute the probability of observing H, given that its weight is maximum among all subgraphs over the same set of conditions U'. To this end, we note that H is obtained by taking into V' all vertices $v \in V$

whose weights $w(\{v\}, U')$ are positive. Let $f_{U'} : V \rightarrow \mathcal{R}$ be a function defined as $f_{U'}(v) = \max\{0, w(\{v\}, U')\}$. For each $v \in V$ we can view $f_{U'}(v)$ as a random variable. The weight of H is just $w(H) = \sum_{v \in V} f_{U'}(v)$, a sum of independent random variables. These variables can be shown to satisfy the requirements of Liapunov's generalization of the Central Limit Theorem (cf. [29]), implying that when $|V|$ is sufficiently large, the weight of H is approximately normally distributed. Hence, we can compute the expectation and variance of $w(H)$ and derive a p-value $p(H)$ for observing a subgraph with such weight.

We next have to accommodate for the fact that the subset U' is optimized by the algorithm. For that, we apply Bonferroni's rule and compute an upper bound on the p-value: $p^*(H) = p(H) \sum_{i=1}^{\lceil (r+1)d \rceil} \binom{m}{i}$, since we are trying all subsets of U of size at most $\lceil (r+1)d \rceil$, where r is the maximum weight ratio in the graph. Henceforth we call $\log p^*(H)$ the *significance value* of H.

Figure 3.10 presents our analysis of the empirical behavior of the likelihood and significance scores, supporting the use of likelihood ratios as reliable figures of merrits that are easy to compute. large datasets.

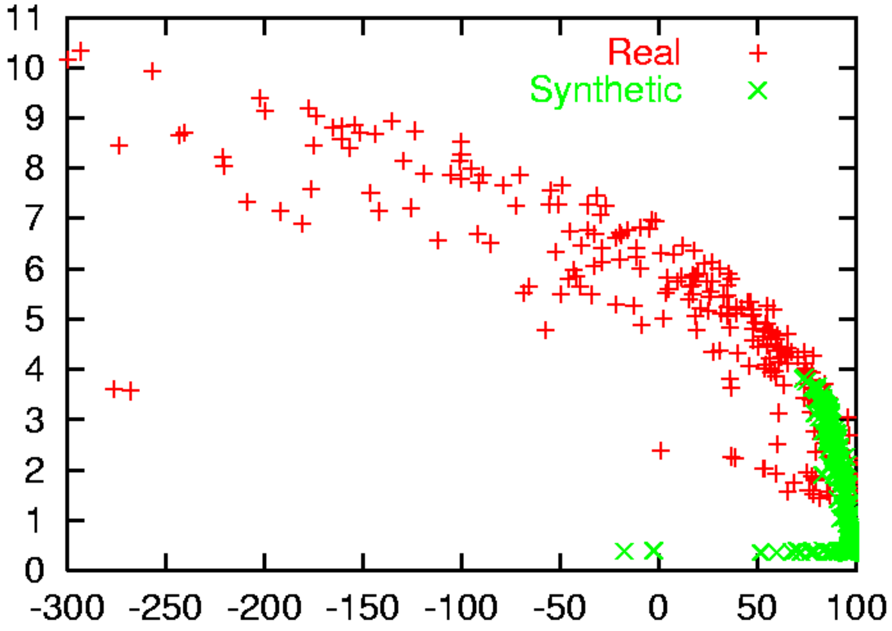


Figure 3.10: **Comparing likelihood ratio to approximated p-values**. X-axis - significance value. Y-axis - likelihood score. A set of biclusters generated using real yeast data (red) exhibit quadratic dependency between the likelihood and p-value. A set of biclusters generated using randomized data (green) is clearly limited in both its p-values and likelihood scores.

Chapter 4

Functional modules

In this chapter we apply the algorithms we described above to analyze a large compendium of yeast data. Using biclustering, we shall reveal groups of genes that form *functional modules* in the yeast genome. We will then show how functional modules can be exploited to A) improve our understanding of the function of individual genes, B) construct models for the transcriptional regulation of modules and C) identify organizational principles affecting the system as a whole. Given our set of modules, we shall then develop an integrative methodology for analyzing genome-wide datasets in the context of the existing compendium and exemplify how this methodology provides rich and informative basis for exploring transcriptional regulation.

Most of the results presented in this chapter were published in [133, 136]. Biological experiments were performed by Prof. Martin Kupiec's lab at the Department for Biotechnology and Molecular Biology, Tel-Aviv university. Parts of the visualization tools for SAMBA were programmed with Israel Steinfeld.

4.1 A yeast data compendium

We have chosen to analyze the modular organization of the yeast *S. cerevisiae*, being one of the best characterized model systems and the species on which many of the novel experimental techniques are being developed. We have built a compendium of yeast functional data including profiles from 52 gene expression studies, 5 transcription factor location studies, 3 synthetic lethality studies and data on protein interactions from the GRID database. Overall, our dataset consists of 1760 different genomewide experiments, the complete list of (hyper linked) references for all data sources is available on the web (www.cs.tau.ac.il/∼rshamir/qubic/)[1]

Our algorithmic framework, as described in the previous chapter, transforms all sources of information into properties which are analyzed in the context of one weighted bipartite graph. We used several types of data sources: gene expression profiles, TF location profiles, growth sensitivity profiles and protein interaction data. We transformed the experimental measurements into probabilistic properties as follows:

- **cDNA gene expression data** (e.g., [123]). First, we rank-normalized the data of each condition. We then generated from each experiment 3 properties (”high”, ”medium or stronger”, ”low or stronger”) for up-regulation and 3 properties for down-regulation. Each property is defined by a *translation function* converting the physical measurement (log expression fold-change) to a probability of having the property. The translation functions we used are the piecewise linear functions shown in Figure 4.1.

- **Affymetrix gene expression data** (e.g., [18]). For each study we selected a reference condition and transformed all data to log the ratio of each condition vs. the reference condition. We then treated the data as for cDNA arrays.

- **TF location data** (e.g. [59]). We generated three properties (”strong binding”, ”medium or stronger binding”, ”low or stronger binding”) for each ChIP profile. The p-values generated by the error model of the original studies were ranked normalized and transformed to properties probabilities using the translation function shown in Figure 4.1.

- Growth phenotypes of mutant libraries(e.g. [54]). We ranked normalize each condition profile and generated two properties (’high sensitivity’, ’medium sensitivity’) using the translation functions described in Figure 4.1.

- Protein interaction. We used interaction data generated by several technologies (two-hybrid and TAP tagging for physical interactions, synthetic lethality

[1] Some of the results reported below were generated using an earlier version of the compendium, including data from only 30 gene expression, TF location and protein interaction studies.

> screens for phenotypic interactions). For each protein and each type of technology we have generated one property (”interacting with protein p in experiment X”). We omitted properties that could be associated with less than 15 gene products and used hard discretization (property probability equals 1 if interaction was detected and 0 otherwise).

We applied biclustering to the combined heterogeneous data set and derived a set of 1248 statistically significant modules. Significance was validated using randomized control tests (see Figure 4.2). Recall that each module consists of a set of genes and a set of properties, such that the genes have significant and correlated values over the set of properties. For example, a bicluster may be defined by a set of genes that are (1) co-expressed in several conditions, (2) are targeted by the same transcription factors, and (3) their protein products are likely to interact with a certain protein. To understand the biology behind specific modules we automatically associated them with known processes and regulatory mechanisms. We assigned modules to biological processes using functional enrichment tests based on the SGD GO annotation (see Appendix A). We also searched for known and novel enriched cis-elements in the promoters of the genes in each module and manually annotated the discovered motifs (see Appendix B). When discussing modules, we use the module number, the primary biological process associated with it (when available) and the module's number of genes and properties. For example, module #524 (RNA processing, 76x211). A single biological process may be represented by several modules of varying sizes and specificities, but under the assumptions described in the previous chapter, the algorithm guarantees that no two modules are similar.

4.2 Synergism between different sources of data

We wished to test how much synergism exists among the experimental data from different studies and to what extent the SAMBA framework exploits such synergism. The distribution of module dimensions (Figure 4.3B) indicates that the comprehensive compendium gives rise to specific modules, with 10-50 genes supported by 20-100 conditions. The distribution of the number of studies contributing properties to each module (Figure 4.3C) demonstrates a high level of synergism in the multi-study data compiled. 86% of the modules used data from more than one study and 68% used data from three studies or more, showing that indeed, infor-

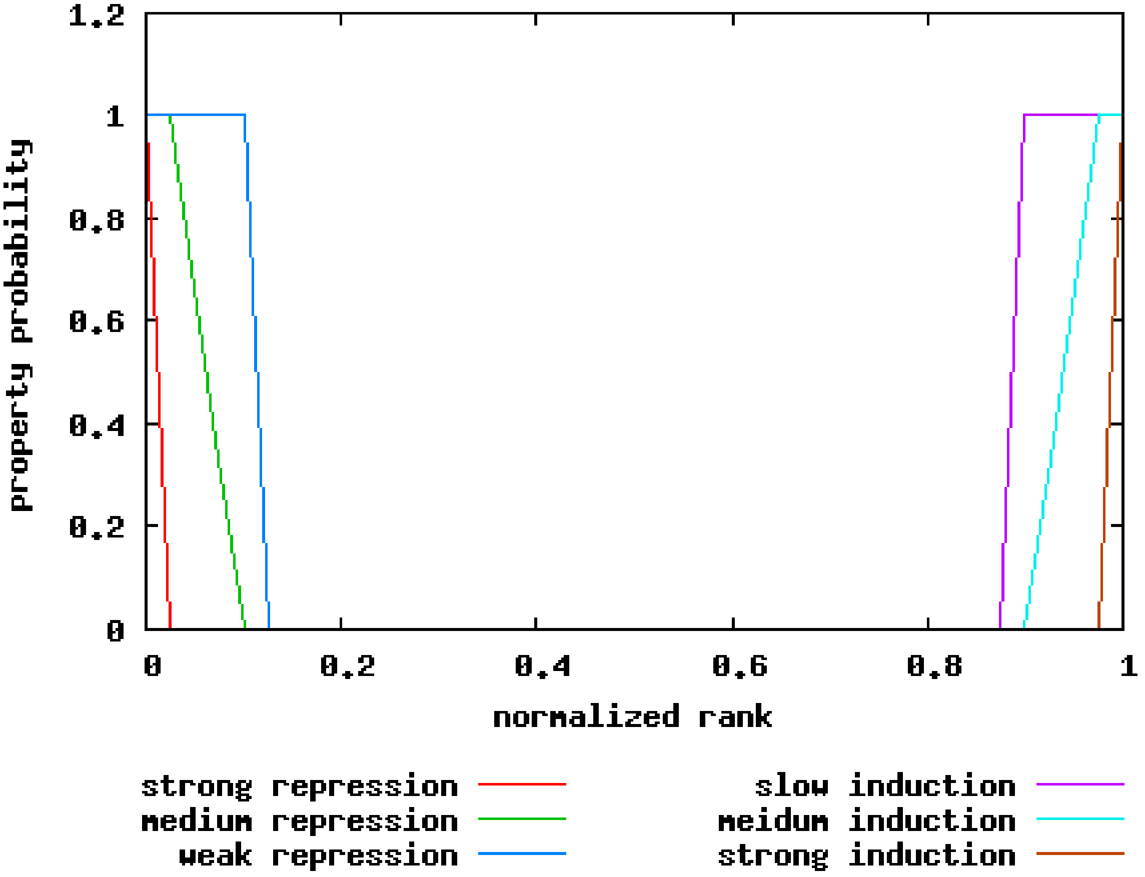


Figure 4.1: **Translating measurments to property probabilities**. We used piecewise linear functions to transform ranked normalized values (from gene expression, ChIP and growth rate experiments) into probabilities of having properties. The six functions plotted above correspond to six properties for different levels of up or down regulation in gene expression. For ChIP we used only the three functions corresponding to ”repression” (so that genes with small ChIP p-values were assigned with the property). For growth sensitivity we used only the functions ”strong induction” and ”medium induction” and applied them to the sensitivity scores reported in the original studies [54].

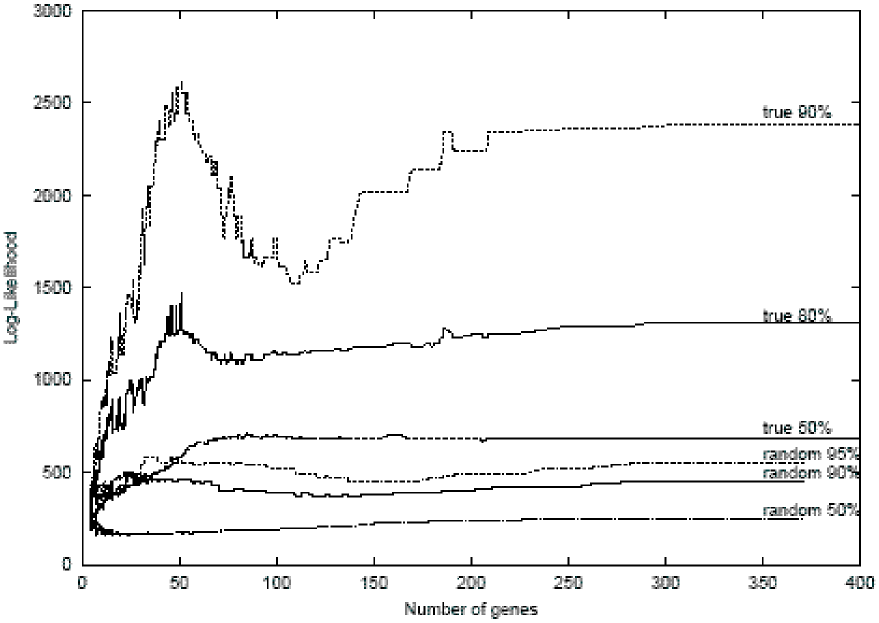
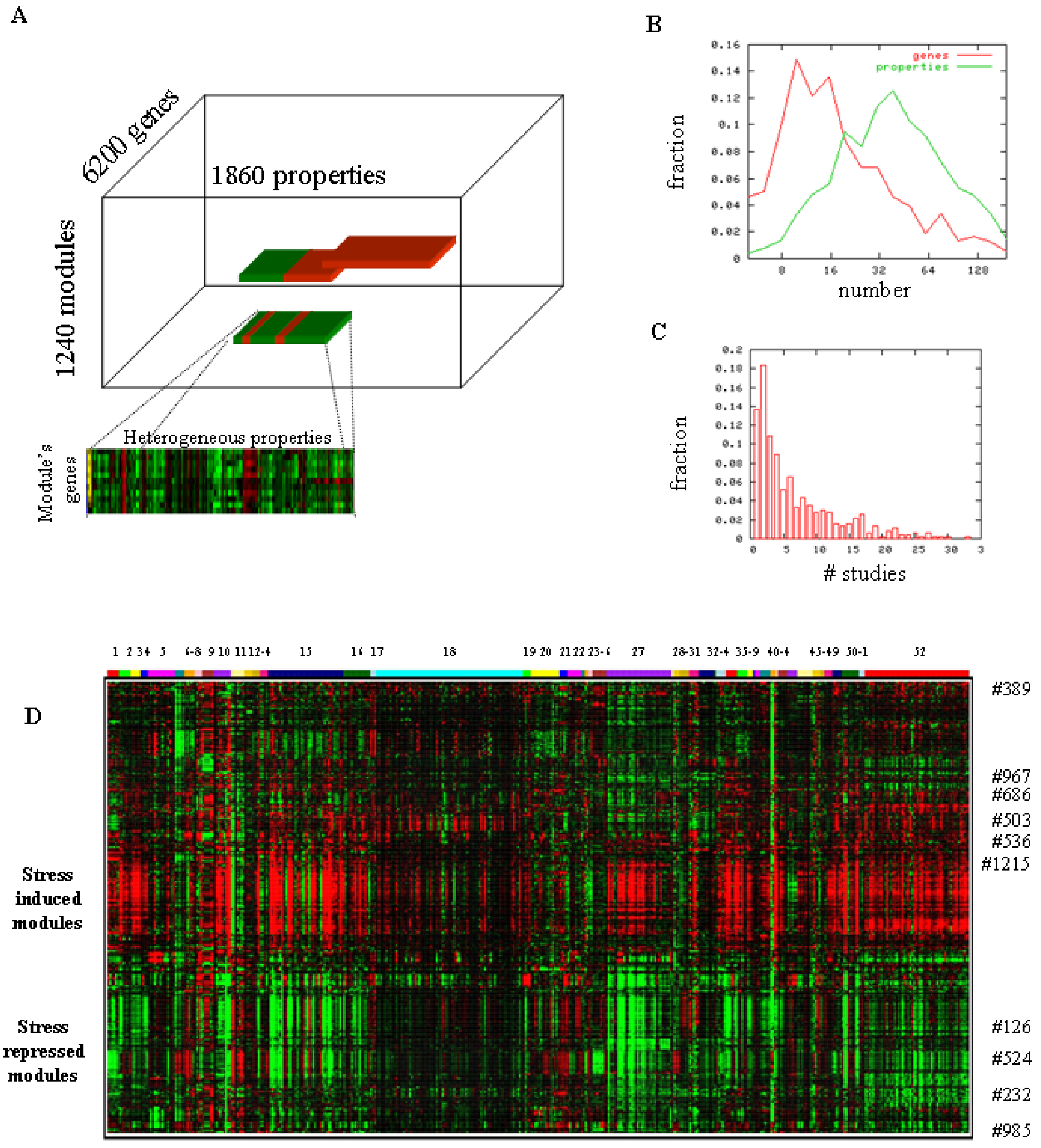


Figure 4.2: **Likelihood of true and random modules.** We applied SAMBA to real and shuffled datasets and plotted the distribution of minus log likelihood scores of modules (Y axis) as a function of the size of the module gene sets (X axis). Shuffled datasets preserved the same degree distribution of the resulting SAMBA graph. For each size, we computed the distribution of the scores of modules of that size, and plotted the $k - th$ percentiles (k = 90%, 80% and 50% for true modules, 95%, 90% and 50% for modules in randomized data). The score distribution was computed on the largest 300 modules not exceeding that size. This was repeated for the original data and for randomly shuffled data. It is evident that the scores of a large fraction of the modules detected in the original data exceed the maximal random score. The graph also shows that modules with 40-70 genes carry particularly high information content; indeed, many biological processes in yeast consist of several dozens of interacting proteins or co-regulated genes.

mation was extracted from multiple datasets and is not biased by one predominant source. A global representation of the compendium and its dissection into modules is obtained by clustering the mean module expression across all experimental conditions. Since the same gene may be part of several modules, such clustering allows the "unfolding" of the function of pleiotropic genes and differs substantially from a standard gene-by-condition clustering. The resulting representation (Figure 4.3D) shows how two opposite environmental stress responses (ESRs [45]) dominate the entire compendium. This response to stress is so strong and widespread, that other, condition-specific regulatory programs are hard to detect without the combination of multiple studies and the application of sensitive algorithms. As we shall see below, separating the general stress response into specific modules and comparing their activities in different conditions provides further insights into the complex regulation of this biological process.

4.3 Biclusters can integrate heterogeneous data

While integration of data from several studies is clearly very effective in extending the scope of the analysis, integration of data from different technologies, relating to completely different biological mechanisms and processes is a much bigger challenge. In general, true integration of such data requires a model based approach (as in, e.g., [48, 47]). Nevertheless, as we shall exemplify below, in several cases the high level approach we are taking in this chapter can successfully exploit heterogeneous data and discover functional modules that are otherwise indistinguishable from other genes. The most straightforward kind of integration is between gene expression and TF location data. For example, the methionine biosynthesis module (Figure 4.4) groups together a highly specific set of genes related to methionine metabolism. The module is supported by diverse expression profiles measured in many knockout strains and stress conditions. Mere expression profiles, however, do not suffice to identify that module, and the binding profiles of Met4, Met32 and Cbf1 are needed to separate it from other amino acid biosynthesis modules. The integration of protein interaction data into the module identification process provides additional information on relations that are not observed at the transcription level alone. It also allows the interpretation of modules in terms of complexes and cascades. For example, a module related to the proteosome complex (Figure 4.5)



1. Ideker 01
2. Yoshimoto 02
3. Nautiyal 02
4. Carrol 02
5. Roberts 00
6. Natarajan 01
7. Lyons 00
8. Williams 02
9. Primig 00
10. Causton 01
11. Jelinsky 00
12. Chu 98
13. Derisi 97
14. Ferea 99
15. Gasch 00
16. Gasch 01
17. Gross 00
18. Hughes 00
19. Ogawa 00
20. Spellman 98
21. Sudarsanam 00
22. Zhu 00
23. Myers 99
24. Angus-Hill 01
25. Fernandes 04
26. Gerver 04
27. Orourke 04
28. Kornor 04
29. Bernstein 00
30. Chitikila 02
31. Cohen 02
32. Fleming 02
33. Lascaris 03
34. Mutka 01
35. Shamji 00
36. Koehler 03
37. Kuruvilla 02
38. Shcherbakova 01
39. Robertson 00
40. Travers 00
41. Yale 01
42. Geisberg 01
43. Gelling 04
44. McCammon 03
45. Schuller 03
46. Boer 03
47. Linde 03
48. Santiago 03
49. Baetz 01
50. Segal 03
51. Shakoury 03
52. Mnaimneh 04

Figure 4.3: **Integrating yeast functional data.** A) Bicluster analysis. The SAMBA biclustering algorithm analyzes an integrated dataset to discover an extensive collection of modules. Each module consists of a set of genes and is supported by a set of functional properties. B) Modules' dimensions. The distribution of the number of genes and properties in each module indicates that modules are characterized by specific sets of genes (10-50) and a large number of different experiments (20-100) C) Synergism among studies. The graph shows the distribution of the number of studies contributing to each module. 86% of the modules use data from more than one study. D) The Module-Condition view. To obtain a global view of the behavior of our modules across all conditions, we clustered the module mean values across all conditions. Rows represent modules and columns represent conditions, with numbered colored bars indicating the study reporting each condition (the full references of the studies are available on the website). We show low means in green and high means in red. The global view reveals that the massive repression and induction of genes in stressful conditions dominates the compendium. Using integrative analysis we can dissect this response into components and study their specific regulation. The numbers on the right refer to modules addressed in this chapter.

is based on the combination of dense protein interaction data with expression data indicating up-regulation in many stress conditions. Protein interactions help in separating this module from other stress responsive genes, and also shed light on the cellular mechanisms forming it.

4.4 Understanding module's regulation

Defined by data from many different experiments, modules can characterize highly specific biological phenomena. Module #126 (Figure 4.6A) consists of 11 genes related to cytokinesis and daughter-specific expression. Of these genes, *DSE1-4, SCW11, CTS1, EGT2, AMN1* and *BUD9* are known to be localized to the daughter cell during late mitosis, and are associated with cell wall separation and exit from mitosis [26]. *SUN4* is also known to be involved in cell septation [143] and *PRY3* encodes a cell-wall specific protein of unknown function. The association of these genes into a single module was based on gene expression data from 261 conditions taken from 30 different studies, and the transcription factor location profiles of the cell cycle regulators Ace2, Swi5 and Fkh2. Indeed, Ace2 and Swi5 are known to have positive and negative effects, respectively, on the transcription of some of the genes in this module [33]. Fkh2 is known to regulate genes required for the G2/M transition and has been implicated (together with Ndd1 and Mcm1) in the regulation of the *SWI5* and *ACE2* genes [122], but its direct association to cytokinesis genes, to the best of our knowledge, was not noted before. This possible role for Fkh2 is supported by evidence for its involvement in the regulation of pseudohyphal growth [155] and by its synthetic lethality with *CLA4* [57], a gene involved in polarization and budding

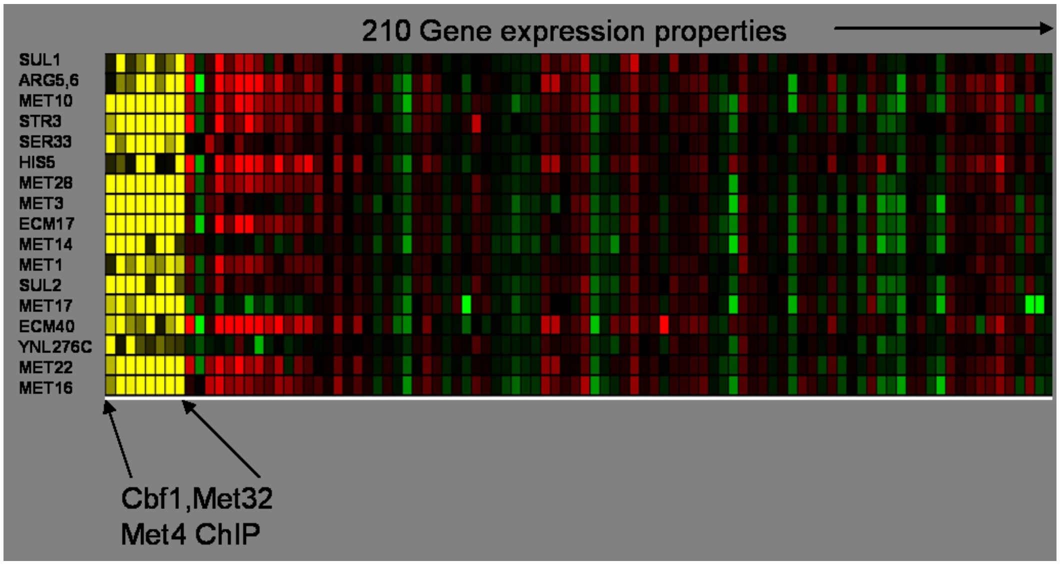


Figure 4.4: **Methionine biosynthesis module.** The integration of data from diverse sources enables the detection of modules in fine granularity. In this example, a group of genes related to methionine biosynthesis is is characterized by 8 binding profiles from 3 TFs (Cbf1, Met4, Met32) that are part of the Cbf complex (yellow columns) with many expression profiles measured in a wide variety of genetic and environmental stimulations (red/green). Note the uncharacterized gene, YNL276C, that was assigned to the module based on the combination of properties, although its expression profiles are only moderately correlated with the module's profile. Interestingly, a *ynl276c* mutant strain was also shown to be moderately sensitive to growth on minimal media.

which functions in a cascade regulating exit from mitosis [65]. The association of Fkh2 with cytokinesis genes may reflect the need to inhibit the function of these genes until mitosis is completed or during transition to pseudohyphal growth.

The wealth of functional information used to construct the module enabled us to explore the behavior of this important transcriptional program across many different experimental conditions. In particular, we analyzed the behavior of the module genes in experiments perturbing different transcriptional co-activators and co-repressors [126, 4, 49] to try and refine our understanding of the mechanisms of transcriptional regulation used in timing the mitotic events. The module exhibits a statistically significant response in several such experiments (Figure 4.6B). Strong induction is observed upon perturbation of the SWI/SNF chromatin remodeling complex (T-test, $P < 0.0001$ in minimal media, $p < 0.001$ in rich media, for both mutants). Strong repression was observed in an experiment that inactivated the

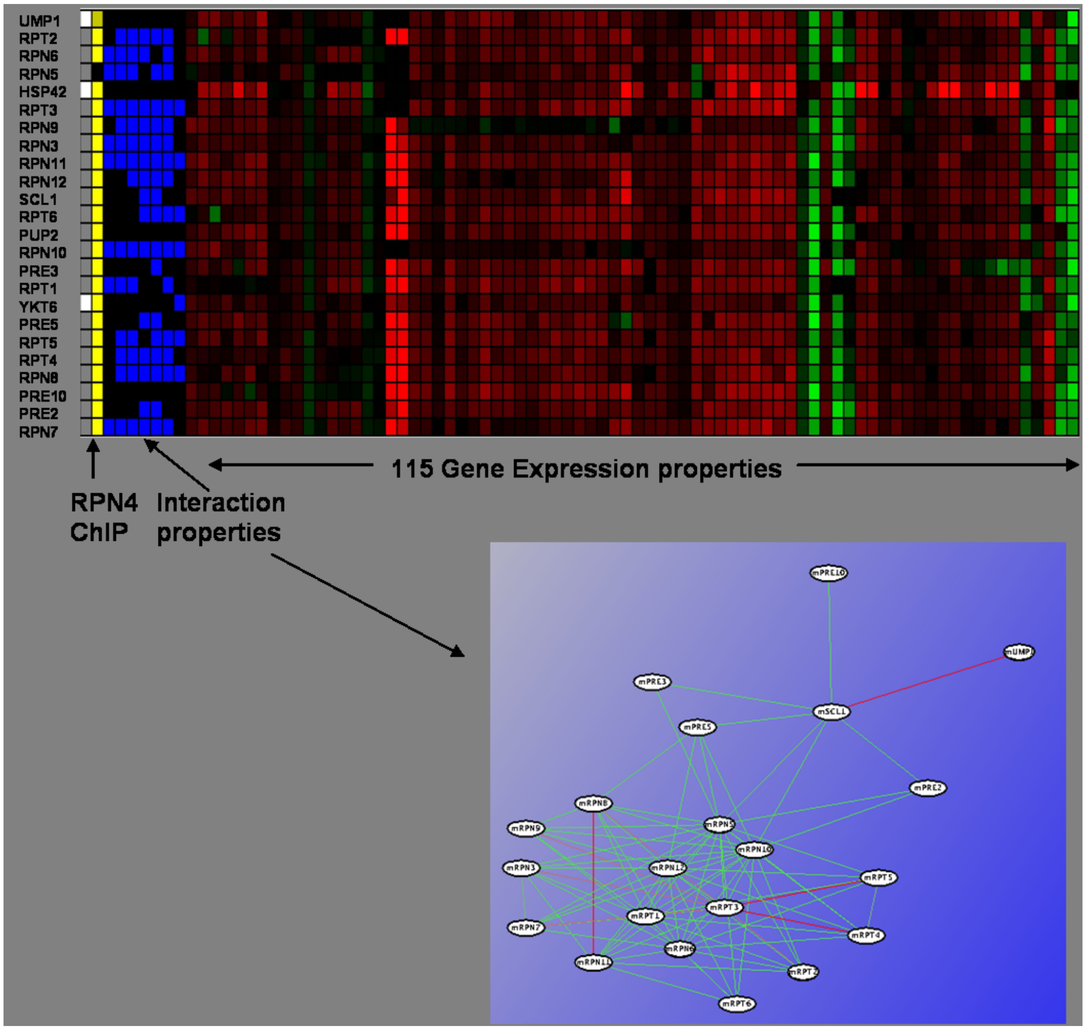


Figure 4.5: **Incorporating protein interactions and expression.** A module related to the proteosome complex is detectable by a combination of expression data and protein interactions. The interactions help in isolating this module, and also suggets the role of the observed cases of module induction/repression.

RSC factor Rsc3 ($P < 0.0004$), but no effect was detected when the RSC factor Rsc30 was inactivated ($P < 0.89$). In addition, a strain knocked out for NC2 activity (*BUR6* deletion) exhibited strong increase in the expression of this module ($P < 0.0002$). Interestingly, the behavior of module #126 in the SWI/SNF, RSC and NC2 experiments is unique among all the modules derived by SAMBA (Table 4.1) suggesting that the particular combination of co-factors uncovered may define the particular regulatory behavior of this module. Taken together, our analysis suggests that the module is controlled by an extended regulatory program that includes the well-known Ace2/Swi5 and Fkh2 transcription factors, and a unique combination of co-activators and co-repressors (Figure 4.6C). The cytokinesis module thus exemplifies the power of our methodology to unravel the complex regulation of a group of coordinated genes.

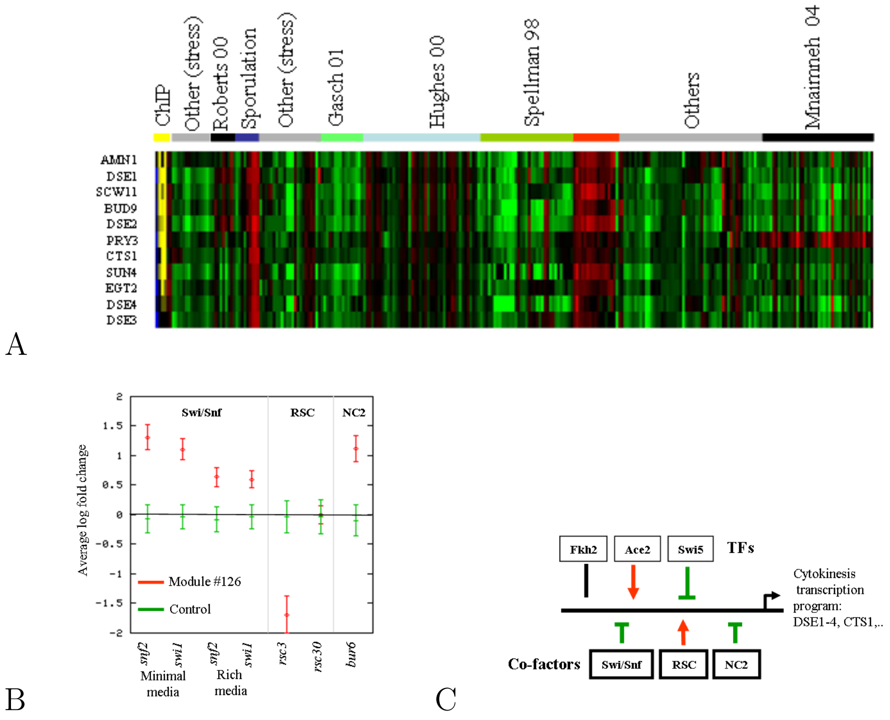


Figure 4.6: **Regulation of a transcriptional module: cytokinesis module**. A) Module #126 is defined by data from 30 different studies and contains a highly coherent set of 11 genes, all but two of which are known to be involved in late mitosis and in cell septation. B) Regulation by co-activators and co-repressors. We plot the average module expression (red) and the background genome-wide mean and standard deviation for a random set containing 11 genes (green) under conditions in which the Swi/Snf complex are not expressed in minimal and rich media [126], in conditions blocking components of the RSC complex [4] and in a strain lacking Bur6, a component of the NC2 co-factor [49]. There is significant induction in all Swi/Snf conditions and in the NC2 experiment, indicating a possible negative role for these co-factors in regulating the module. There is also a significant repression in the *rsc3* strain (but not in the *rsc30* strain), indicating that RSC may have a positive role in the regulation of the module. C) Extended regulatory model for the cytokinseis module. See text for details.

Gene	**Process**	**snf1 minimal media**	**swi2 minimal media**	**rsc3**	**rsc30**	**bur6**
DSE1	Mod #126	2.44	2.24	-2.42	-0.22	0.79
DSE3	Mod #126	0.88	0.78	-1.88	0.38	1.11
DSE2	Mod #126	1.83	1.60	-2.75	0.41	1.33
AMN1	Mod #126	1.42	1.13	-1.97	0.01	0.99
EGT2	Mod #126	1.75	1.47	-1.18	-0.80	1.01
CTS1	Mod #126	1.37	0.99	-1.55	-1.00	1.58
BUD9	Mod #126	0.85	0.73	-0.92	0.05	1.24
SCW11	Mod #126	2.22	1.80	-2.74	-0.26	0.84
YRO2	cell wall	3.72	3.86	-2.04	-0.74	1.23
WSC4	cell wall	2.53	2.42	-1.57	0.48	1.58
YGP1	cell wall	1.96	2.03	-1.30	-0.32	1.01
ENO1	glycolysis	0.96	1.11	-1.55	-0.62	0.78
TDH1	glycolysis	1.69	1.54	-2.74	-0.59	2.13
PGM2	glycogen	0.86	0.92	-2.16	-0.61	4.35
GSY2	glycogen	1.69	1.19	-1.55	-0.75	2.21
GCV1	glycogen	1.26	1.57	-2.14	0.76	1.70
ADE17	purines	0.77	0.84	-1.35	0.52	1.51
YER067W	unknown	1.29	1.31	-1.66	-0.61	2.83
YOL106W	unknown	1.63	1.54	-1.79	-0.54	1.43

Table 4.1: **cytokinesis cofactor profile.** The table shows all genes that are A) induced (log2(fold change) > 0.7) in *swi2, snf1* (in minimal media) and *bur6* (YPD) B) repressed (log2(fold change) < -0.9) in *rsc3* and C) not significantly responding ($-1 <$ log2(fold change) < 1) in *rsc30*. Only 18 genes satisfy this criterion, eight of which belong to module #126 and additional three are cell wall associated. Out of the remaining eight genes, six are structural enzymes in glycogen or related metabolism and two are uncharacterized.

4.5 Annotating unchracterized genes

As we have shown above, bicluster analysis may identify very specific functional modules using combination of data from many studies and technologies. One possible use of such modules is to perform "guilt by association" and predict the function of genes. Uncharacterized genes in modules showing enrichment ($p < 0.01$ and over

40% of the annotated genes) for genes known to be related to one biological process are likely to participate in the same process. We tested the specificity of this approach by performing a five-way cross validation: We repeatedly applied SAMBA to datasets in which one fifth of the known gene annotations were hidden and tested the specificity of predicting the function of these genes. Overall we obtained 40% to 100% specificity for a variety of classes including mating (GO:0007322, 65%), amino acid metabolism (GO:0006520, 40%), sporulation (GO:0030435, 55%), glucose metabolism (GO:0006006, 100%), lipid metabolism (GO:0006629, 92%) and more (Figure 4.7). Average specificity ranged between 58% and 78%, depending on the strictness threshold used for annotation. In many cases, the classification errors result from ambiguous annotation terms or too general categories. This may represent missing information rather than misclassification. For example, stress response and cell cycle are very general categories that intersect many other processes. Stress-annotated genes are often also related to carbohydrate metabolism and transport, so our classification for such genes may reflect an additional function and not an error. In total, our scheme generated putative functional annotations for 874 uncharacterized yeast genes. The complete list is available on the web (www.cs.tau.ac.il/∼rshamir/qubic). We note that although SAMBA was not designed specifically for performing functional annotation, and although our annotation scheme was very simple once the biclusters are given, we perform well relative to other approaches [153, 140]. This further supports the importance of integrating data sources and analyzing them together.

As an additional test of our annotation accuracy we analyzed, in collaboration with Martin Kupiec's lab, five yeast strains deleted for ORFs predicted by SAMBA to be involved in mating. Quantitative mating experiments showed that four of the strains (*YDR429c, YBR223c/TDP1, YNL106c/INP52* and *YOL106w*) exhibited reduced mating ability, compared to the wild type, confirming the involvement of these genes in the mating process. A fifth ORF, *YFL027c/GYP8*, exhibited mating ability indistinguishable from that of the wild type control.

4.6 The global organization of the yeast system

The combined analysis of gene expression and TF binding location was used before to study the transcriptional network of specific processes (e.g., cell cycle [122,

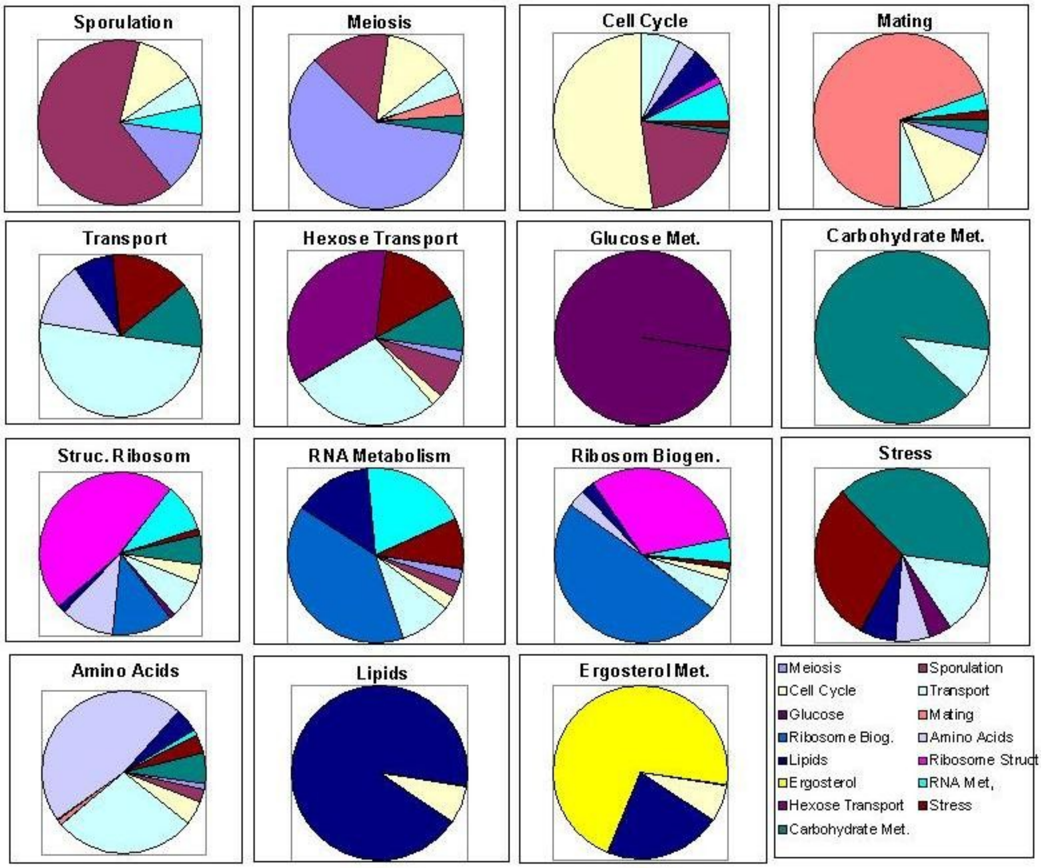


Figure 4.7: **Cross validation of functional annotation.** 874 uncharacterized genes were assigned with putative annotations based on functionally enriched SAMBA modules. We tested the specificity of the annotation process using five-way cross validation, repeatedly hiding one fifth of the annotations and trying to recover them. Each sub-figure plots the distribution of true functions in genes annotated by SAMBA with a specific function. Sub-figures that show partition into several sizeable functions are often a result of overlapping biological processes. For example: sporulation and meiosis, transport and amino acid metabolism, ribosomal structural proteins and ribosome biogenesis.

87, 6]). SAMBA enables the simultaneous analysis of the entire network and the exploration of the relations among TF binding profiles, biological processes and DNA regulatory motifs in a single map (Figure 4.8; for an interactive version see www.cs.tau.ac.il/∼rshamir/biobic). To form the transcriptional network map we generated nodes for all GO processes that were significantly over represented in at least one module. We also added nodes for TFs having ChIP profiles. We associated a TF node with a process node whenever there existed a module annotated with the process ($p < 0.01$) which has the TF binding profile as one of its properties.

By analyzing the transcriptional network map, we could characterize the regulation of several processes, mostly those that are active in the same condition that were used in the ChIP experiments. Cell cycle modules, as previously observed [122, 87], are associated with a combination of known TFs acting in a cyclic fashion. Amino acid metabolism modules are associated with combinations of the master regulator Gcn4 and module specific regulators (Cbf1-Met4-Met31 for methionine and sulfur, Arg80 and Arg81 for arginine). Respiration modules are regulated by Hap2-5, and protein biosynthesis genes are associated with Rap1, Fhl1 and others.

Systems that are activated during stress or developmental processes have weaker support of binding profiles [87]. For example, sporulation modules are associated with Sum1 but not with other important meiotic regulators (Ndt80, Ume6) [107]. Modules that are repressed by the environmental stress response are either well explained by Rap1 binding (the ribosomal protein module), or are highly enriched by the RRPE and PAC binding sites [9] without a matching ChIP profile.

We next turned to explore the global organization of the yeast system as revealed by the association of different modules into one functional network. To this end we constructed and analyzed two graphs. The **gene graph** contains as nodes all yeast genes, with an edge between two genes whenever they are both contained in some module. The **module graph** (Figure 4.9) contains as nodes all the modules, with an edge between two modules whenever their gene set intersection is larger than a 33% threshold. Since the gene graph is induced by gene modules (cliques in the graph), it is expected to have a modular structure. The module graph, on the other hand, could not be pre-assumed to exhibit modularity. To systematically analyze the topology of the two graphs, we computed their clustering coefficients [110]. The clustering coefficient of a node is the fraction of the pairs of its neighbors that have edges between them. High average clustering coefficient is an indication of modularity. For example, a tree graph has value zero, and a complete graph has value 1. As expected, the average cluster coefficient of the gene graph is a high 0.473. Interestingly, the module graph also has a very high average clustering coefficient of 0.49. We have sampled 1000 random graphs with the same degree distribution and computed the mean coefficient 0.0398 and standard deviation 0.008, supporting this graph is significantly more modular than expected by chance.

We note that it is very difficult to completely rule out algorithmic artifacts that

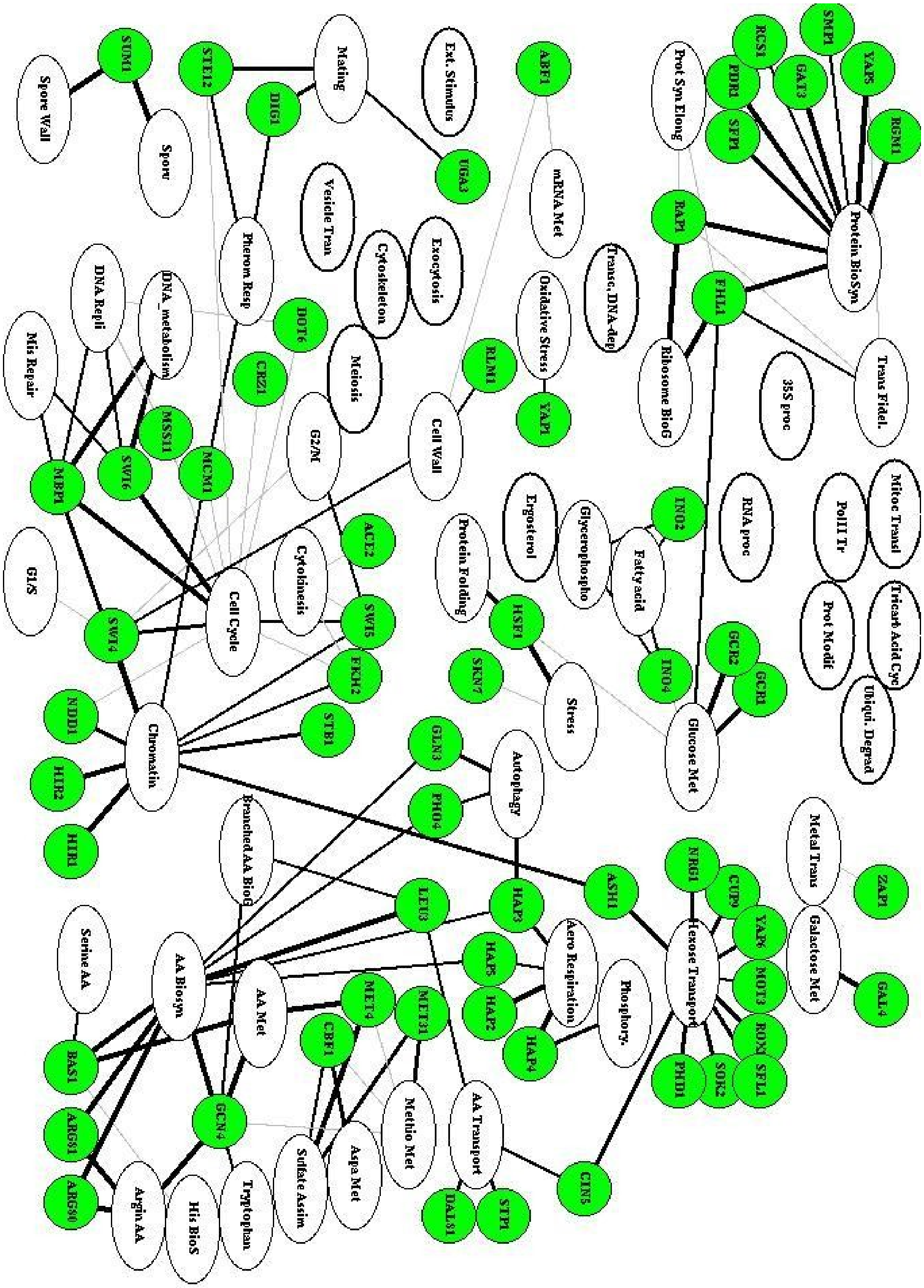

Spore Wall
SUM1
Sporu
STE12
DIG1
Mating
UGA3
Ext. Stimulus
ABF1
mRNA Met
Prot Syn.Elong
RCS1
PDR1
SFP1
GAT3
SMP1
YAP5
RGM1
Protein BioSyn
RAP1
FHL1
Ribosome BioG
Trans Fidel.
35S proc
Transc. DNA-dep
Oxidative Stress
YAP1
RLM1
Cell Wall
Exocytosis
Cytoskeleton
Vesicle Tran
Pherom Resp
DOT6
CRZ1
Meiosis
G2/M
DNA Repli
DNA_metabolism
Mis Repair
MSS11
SWI6
MCM1
MBP1
G1/S
SWI4
Cell Cycle
Cytokinesis
ACE2
SWI5
FKH2
STB1
Chromatin
NDD1
HIR2
HIR1
Protein Folding
HSF1
SKN7
Stress
Ergosterol
Glycerophospho
Fatty acid
INO2
INO4
RNA proc
PolII Tr
Mitoc Transl
Prot Modif
Tricarb Acid Cyc
Ubiqui. Degrad
GCR2
GCR1
Glucose Met
GLN3
PHO4
Autophagy
Branched AA BioG
LEU3
HAP3
ASH1
NRG1
CUP9
YAP6
MOT3
ROX1
SFL1
SOK2
PHD1
Hexose Transport
Metal Trans
ZAP1
Galactose Met
GAL4
Serine AA
AA Biosyn
AA Met
HAP5
HAP2
HAP4
Aero Respiration
Phosphory.
MET4
MET31
CBF1
BAS1
ARG81
ARG80
GCN4
Argin AA
His BioS
Tryptophan
Sulfate Assim
Aspa Met
Methio Met
AA Transport
DAL81
STP1
CIN5

Figure 4.8: **Functional modules and their transcription factors in the yeast system.** Modules with significant functional enrichment for a particular process ($p < 0.01$) are grouped and plotted as an oval with the process name. TFs with binding profiles associated with any of these modules are marked as green circles and connected to the associated process. Modules may be enriched in more than one process and thus contribute to several regions in the map. The thickness of the connecting lines is inversely proportional to the p-value of the functional enrichment in the associated module. The map was automatically generated by SAMBA using no prior biological knowledge. Key abbreviations: Met: Metabolism, AA: Amino Acid, Tran: Transport. An interactive version of this figure is available on our website.

generate a modular graph just due to a systematic errors (e.g., in redundancy elimination). Nevertheless, we can analyze biologically characterized cases in which known larger modules are divided into smaller, more specialized modules, and compare what is known to the module map topology. Indeed, in some cases (mostly in metabolism) modules are themselves organized into super-modules (Figure 4.9). The overall organization in those cases is thus hierarchical: Genes are grouped into modules that are clustered into super-modules. Note that genes can participate in more than one module, and modules can be part of more than one super-module.

Given this hierarchical architecture, it is important to characterize the genes that connect different super-modules and tie together different processes. To this end, we recalculated the average cluster coefficient in the gene graph over sets of genes annotated with each GO entry. A class with low average coefficient contains genes that are more likely to bridge different super-modules. Indeed, the classes with lowest coefficients are related to signaling (e.g., G-protein coupled receptor with value 0.27 and MAP kinase with 0.29) and transport (e.g., iron transporter with value 0.21 and phospholipid transport with 0.25). Closer examination of genes with low cluster coefficient may help in identifying genes that have multiple functions and improve our understanding of the way in which different biological processes are organized together.

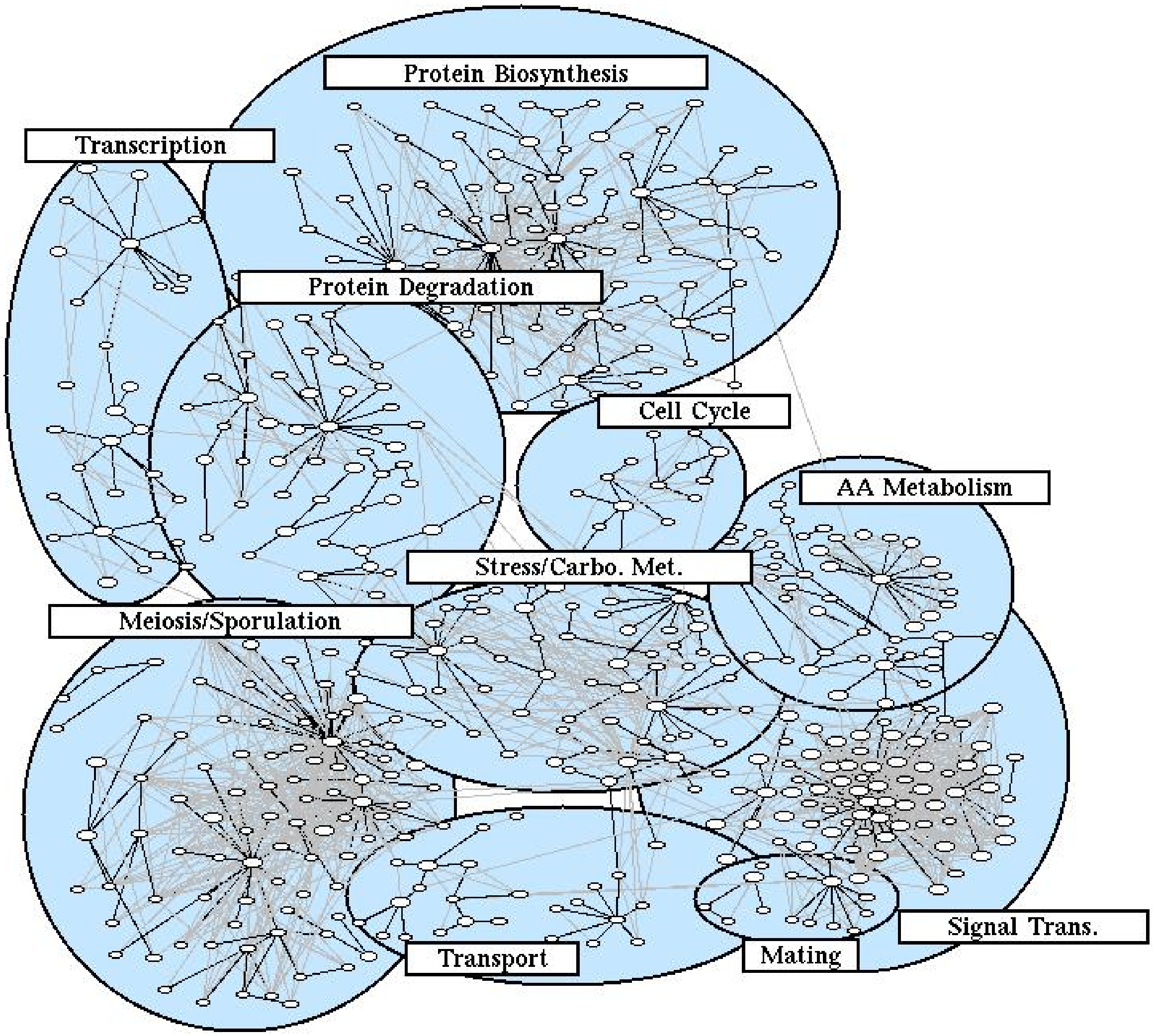


Figure 4.9: **Organization of the yeast molecular network.** The module graph was generated by connecting two modules (small ovals) if over one third of the genes in one (the smaller) are present in the other. We used our module annotations to manually classify regions in the graph (shown as colored large ovals). In several regions, the graph reflects a hierarchical organization that arranges modules in clusters. Some of the clusters (e.g., protein biosynthesis) are organized in more than two hierarchical levels: large modules are composed of several smaller modules, giving a star-like topology.

4.7 Module based analysis of the yeast galactose system

Using the data compendium, and the comprehensive collection of functional modules derived from it by SAMBA, we shall next show how to analyze a single high throughput dataset given the entire compendium. The yeast galactose utilization pathway is among the best-characterized biological systems. In a systematic set of experiments, Ideker et al. [68] measured the transcriptional response of yeast strains knocked out for a set of enzymes and regulators involved in galactose metabolism. The data were then clustered and analyzed in light of the known Gal4-Gal80-Gal3 regulatory circuit. We used the galactose dataset as a test case for an integrated methodology: instead of clustering yeast genes given their expression in the galactose data set only, we screened the complete set of modules, which are based on almost 2000 experiments, for modules that are responsive in at least one of the conditions analyzed by Ideker et al.. Since the data defining the modules are relevant to many different aspects of the yeast regulatory network, we were able to interpret galactose-related conditions from a broad perspective. We depict the effect of galactose-related conditions on several central modules in Figure 4.10 (interactive visualization of all modules is available on the web (www.cs.tau.ac.il/∼rshamir/qubic). As expected, the strongest effects are well known and were easily observed using clustering of the galactose dataset alone. For example, module #389 (Galactose metabolism, 20x160), the classical Gal4 regulon, consists mainly of enzymes required for the utilization of galactose (*GAL1,2,7,10*) and is strongly repressed when galactose is lacking from the medium or when knockouts in the GAL pathway compromise its yield. The response of other modules, however, is less predictable and reveals novel regulatory relations between different processes.

A first surprising effect revealed by our analysis is the repression of module #524 (RNA processing, 76x211) in *gal4* strains, in both galactose-containing (Paired T-test, *gal4*+galactose/*wt*+galactose $P < 10^{-21}$) and galactose-free media (*gal4*-galactose/*wt*-galactose $P < 10^{-22}$). The repression of this module in mutants lacking structural enzymes is much weaker, and so is the response of the wild type strain to lack of galactose (*gal4*+galactose /*wt*-galactose,$P < 10^{-10}$). Moreover, in three strains knocked out for *GAL80* (the Gal4 inhibitor), grown in medium lacking galactose, we observe induction of module #524 (*gal80*/*wt* $P < 10^{-17}$, *gal80gal2*/*wt*

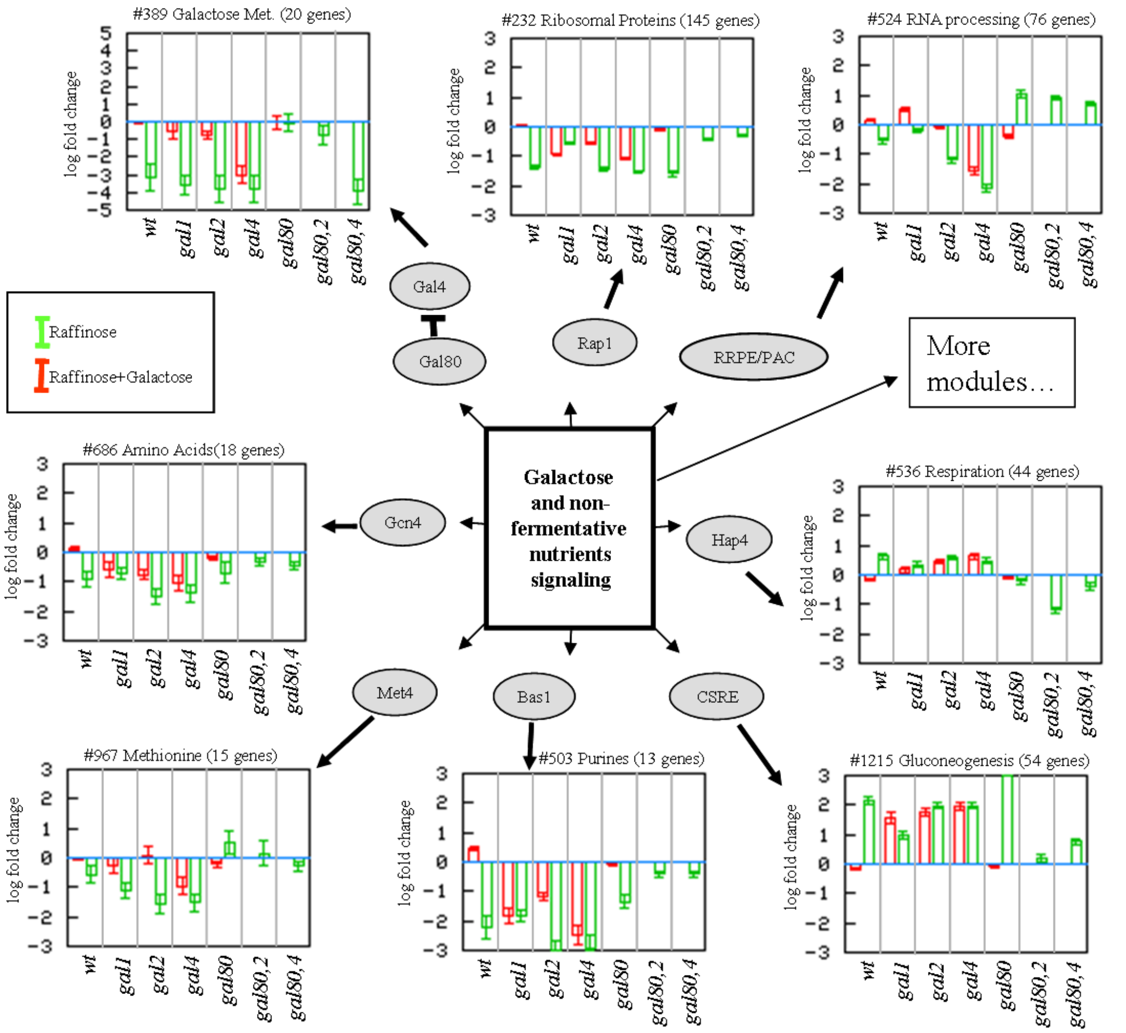


Figure 4.10: **Revisiting the galactose system.** Response of selected modules to disruptions in the GAL system. We plot the mean and standard deviation of the expression of several key modules that our algorithm associated with conditions from the galactose dataset [68]. For each module, we plot the behavior in four galactose related mutants and two double mutants grown with (red) and without (green) galactose. Module #389 (galactose metabolism), is strongly repressed when galactose is lacking or when the GAL pathway yield is compromised. Modules #232 (ribosomal proteins) and #524 (RNA processing) are repressed when growth is slower. Interestingly, module #524 is particularly repressed when *GAL4* is knocked-out, and is induced when *GAL80* is knocked-out and galactose is not available (right-most bars). Modules #536 (Respiration) and #1215 gluconeogenesis) are induced when galactose is not available or not processed. Here again, the *GAL80* mutants exhibit altered behavior (module #536 is repressed). Modules #503 (Nucleotides), #686 (Amino acids) and #967 (Methionine) are repressed when growth is slower.

$P < 10^{-25}$, *gal80gal4/wt* $P < 10^{-20}$). This result includes the double mutant *gal4gal80*, implying that the effect is Gal4-independent. The induction of module #524 is particularly interesting given the slow growth and transcriptional repression of module #232 (ribosomal proteins, 145x269) in the *gal80* strains. Across the entire compendium, the expression of modules #524 and #232 is tightly coupled, as both are strongly repressed under general stress conditions [45]. The correlation between the mean expressions of the two modules across 1500 gene expression conditions is indeed very high (Pearson=0.73, Figure 4.11). The marked difference between the expression of the two modules in the *gal80* and *gal4* experiments (Figure 4.12) represents a regulatory discrepancy whose mechanistic causes are still unclear. Module #524 is regulated by the two highly enriched cis-elements PAC (GCGATGAG) and RRPE (GAAAATTTT), but it is still not known which factors bind these sites. Module #232 is regulated by Rap1 and possibly by additional factors [95]. Some interaction between these factors, their co-activators/repressors and the Gal4/Gal80 circuit may account for the mutants altered response.

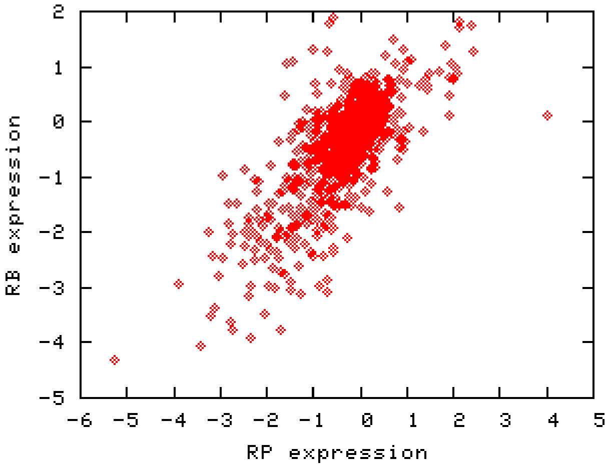


Figure 4.11: Correlation of module #232 (Ribosomal Proteins) and module #524 (RNA processing). The graph shows the mean expression of the two modules across more than 1500 gene expression profiles.

Mutations in genes of the galactose pathway and changes in the carbon source have an extensive effect on the yeast metabolism as a whole. The transcriptional regulation of nonfermentative metabolism involves a complex network of transcriptional regulators, co-activators and co-repressors (reviewed in [117]). Many of the modules that were associated with the galactose dataset are linked to different metabolic activities. Using data from different studies we can dissect the general metabolic

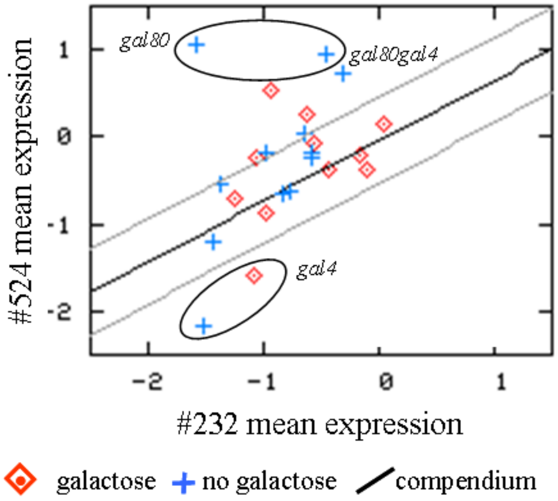


Figure 4.12: **Disrupted coupling of two stress-related modules.** The plot shows the mean expression of module #232 (Ribosomal proteins) and module #524 (RNA processing) in the galactose pathway experiments, together with the linear regression line of the dependency between the mean expression levels of the two modules over the entire compendium (Methods). Broken lines indicate 1 STD. The *gal80* mutants exhibit increased expression, while the *gal4* mutants (excluding *gal80gal4*) exhibit decreased expression relative to the compendium trend, supporting a possible involvement of the gal80-gal4 circuit in the regulation of the modules.

response into basic building blocks, thereby shedding light on the regulatory interactions that gave rise to it (compare Figure 4.10). Overall, we observe two types of behavior. Modules #1215 (Gluconeogenesis, 54x86) and #536 (Respiration, 44x156) are generally induced in conditions in which the yield of the galactose pathway is compromised. Modules #503 (Purine metabolism, 13x198), #686 (Amino acid biosynthesis, 18x150) and #967 (Methionine metabolism, 15x156) are repressed under these conditions. This general trend fits well with our understanding of the yeast regulatory program. Yeast cells respond to the lack of galactose-based energy by increasing the activity of the respiratory pathway and adapt to slower growth by reducing biomass production. Given these general, well-documented trends, the behavior of the *gal80* strains again remains unexplained. Module #536 (respiration), for example, is repressed in *gal80, gal2gal80* and *gal4gal80* strains in the absence of galactose (Figure 4.13), although there is no yield from the galactose pathway under these conditions. The repression cannot be explained by constitutive expression of

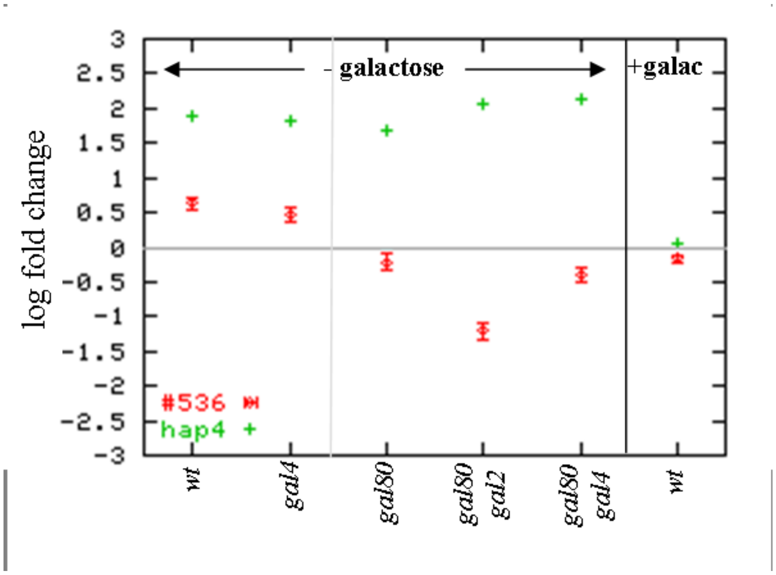


Figure 4.13: **Hap4-independent repression of module #536 in *gal80* strains.** We plot the mean expression of module #536 (respiration) and the expression of the gene coding for its direct regulator Hap4 in selected conditions. When galactose is not available, module #536 is induced via increased expression of *HAP4*. Similar effect is observed in several other conditions, for example in the *gal4* strain. In *gal80* strains we observe repression (or lack of induction) of the module, although the HAP4 gene is expressed at high levels.

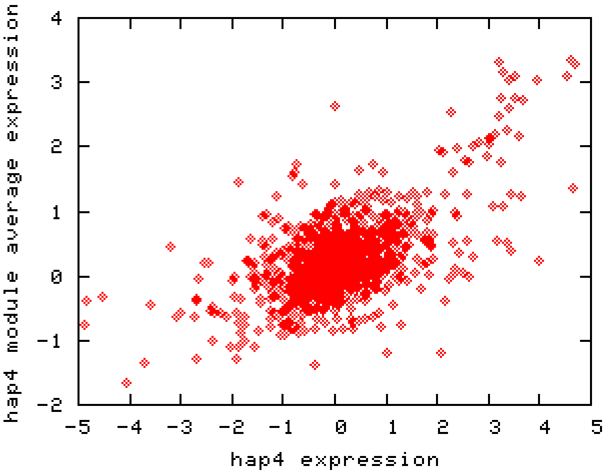


Figure 4.14: Correlation between expression of the HAP4 gene and of the Hap4- regulon. The graph shows the expression of the HAP4 gene and the mean expression of module #536 across more than 1500 gene expression profiles.

GAL genes, given that expression is reduced also in the *gal4gal80* double mutant. Module #536 is regulated by the Hap2-5 complex, and *HAP4* is itself part of the module [117]. There is a strong correlation between Hap4 expression and expression of module #536 across the entire compendium (Pearson = 0.65, Figure 4.14). Nevertheless, in the three conditions in which *GAL80* is inactivated, Hap4 is strongly induced while its module exhibits significant repression, suggesting the involvement of other factors in the repression of the respiratory genes. Other modules show different deviant responses to the gal80 knockout. For example, a Met4/31 module (#967) is induced in the *gal80* strain, in contrast to its general repression in other conditions with reduced energy flux. Given the involvement of Gal80 in the repression of SAGA recruitment to Gal4 binding sites [17] and the similar acetylation patterns found in the Gal4-, Hap4- and Met4- activation sites [28], we hypothesize that in media without galactose addition, Gal80 is capable of affecting the recruitment of co-activators or co-repressors for factors other than Gal4. Overall, our results provide an explanation of the slow growth phenotype of the gal80 strain, suggesting that deletion of this central regulator has far reaching implications, most notably breaking of the coupling between ribosomal proteins and RNA processing modules, and the blocking of Hap4-dependent activation of the respiration module.

4.8 Module based analysis of the yeast response to osmotic shock

In response to hyper-osmotic stress, yeast cells activate a combination of signaling pathways and transcriptional programs (reviewed in [66]. We applied our analysis framework to a set of 129 expression profiles obtained in experiments that tested the response of *S. cerevisiae* to varying levels of osmotic stress in strains knocked out for Hog1, Ssk1 and Ste11, three important proteins in the HOG pathway [101]. The response to high levels of osmotic stress is widespread and involves at least one fifth of the yeast genome. We found that this massive response can be dissected into finer transcriptional programs that govern specific modules (Figure 4.15). For example, modules #232 (ribosomal proteins, 145x269) and #524 (RNA processing, 76x211) are strongly repressed in 0.5M KCl. In the wild type repression peaks at 20 minutes and is alleviated in a HOG1-dependent manner after 40 minutes. This joint effect was noted before based on standard clustering analysis. Using the

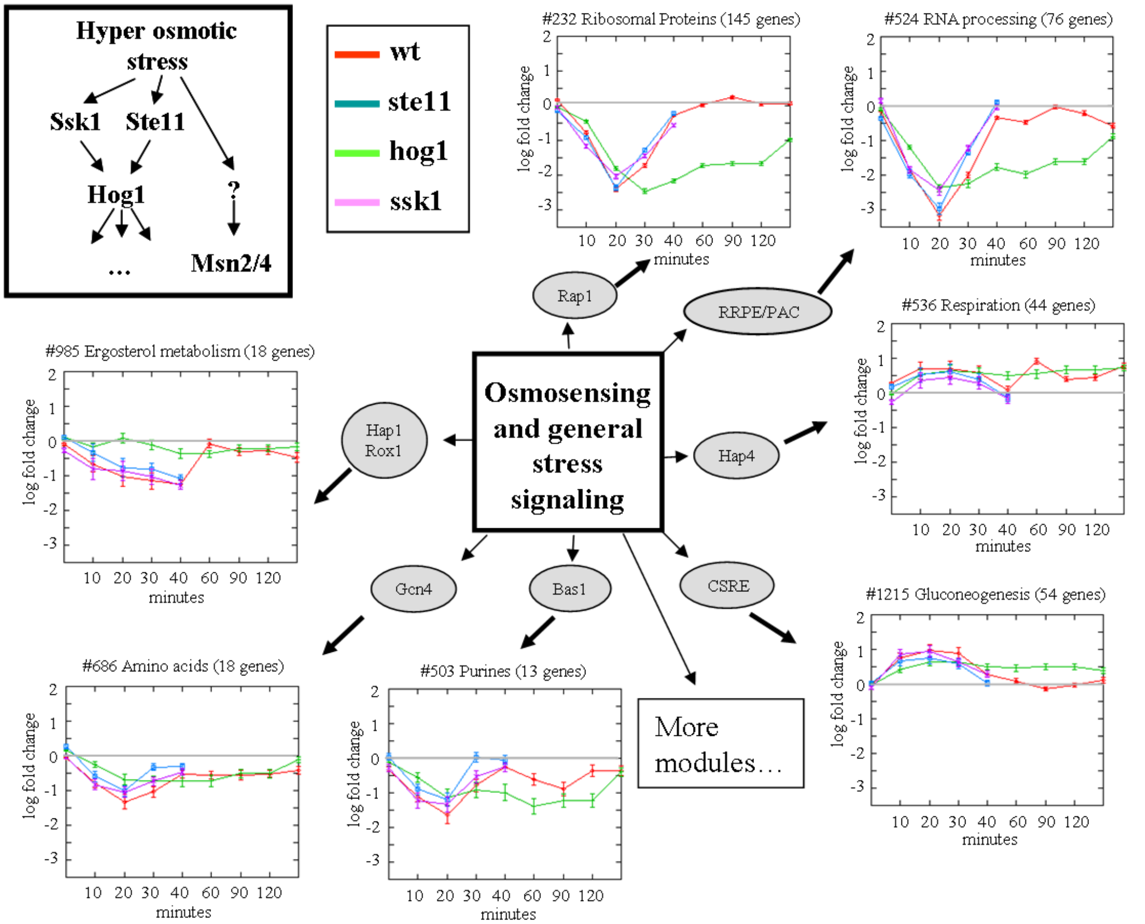


Figure 4.15: **Revisiting the response to hyper-osmotic stress.** Outline of the hyper osmotic stress signaling pathway is shown on the upper left part of the figure. Two Hog-dependent (Ssk1, Ste11) and one Hog-independent (Msn2/4) pathways mediate the hyper osmotic stress signal. We plot the average expression of several modules that were associated with osmotic stress conditions, in several strains knocked-out for key players in the HOG pathway. The graphs show modules' mean expression time courses after treatment in 0.5M KCl. In general, modules #232 (Ribosomal proteins) and #524 (RNA processing), #686 (Amino Acids), #503 (purines) and #985 (ergosterol) are repressed as part of the environmental stress response, with peak response observed at 20 minutes and re-establishment of normal transcription after 40-60 minutes. Modules #536 (respiration) and #1215 (gluconeogenesis) are induced with similar kinetics. Specific modules show particular deviation from these two general trends as discussed in the text.

compendium, we uncover a refined regulatory program. In module #524, the hog1 and ssk1 strains exhibit reduced repression in the presence of 0.5 M KCl (Paired T-test, *hog1/wt* $P < 10^{-20}$, *ssk1/wt* $P < 10^{-14}$, Figure 4.8a), but no reduction is observed for *ste11* (*ste11/wt* $P < 0.14$). Derepression by a *hog1/ssk1* knockout is also noticeable in a medium containing 0.125M KCl, (*hog1/wt* - $P < 10^{-28}$, *ssk1/wt*, $P < 10^{-20}$, Figure 4.8b), and the effect is almost identical for the two knockouts (*hog1/ssk1* $P < 0.002$). Our analysis thus suggests that in medium/low osmotic shock, an Ssk1/Hog1-transmitted signal represses the RNA processing module activity, whereas during high osmotic shock, a Hog1- independent pathway is repressing the module additively to the Ssk1/Hog1-mediated effect (Figure 4.16).

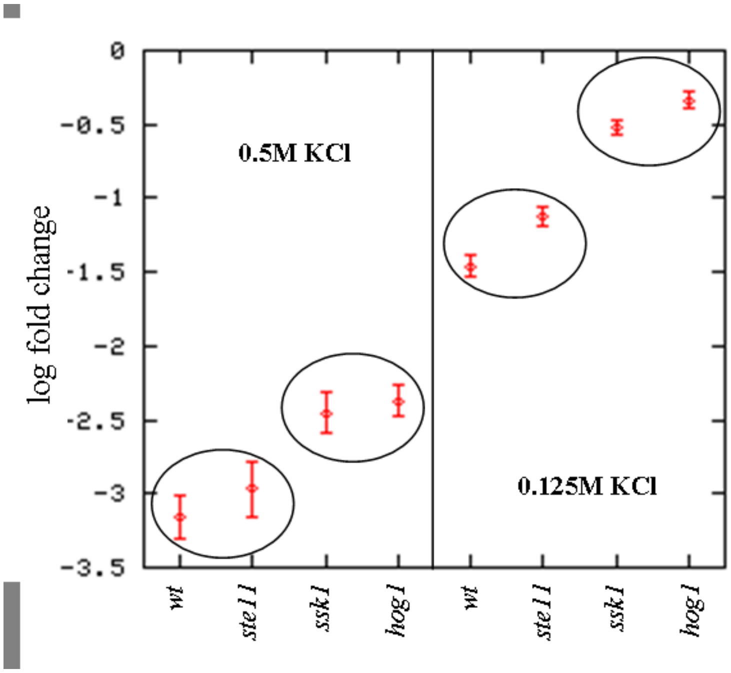


Figure 4.16: **Multiple signals additively regulate module #524.** We plot the mean expression of module #524 and its standard deviations in four strains (*wt, hog1, ste11, ssk1*) under two levels of hyper-osmotic shock (0.5M KCl, 0.125M KCl). There is marked difference between the *ssk1* and *hog1* strains and the *wt, ste11* strains, suggesting the existence of two regulatory mechanisms. An osmotic stress-specific, Ssk1/Hog1-mediated signal represses the module in both low and high levels of osmotic shock. In high osmotic shock a second, Hog1-independent signal (which is probably related to the general environmental stress response) is active in parallel to the Hog1 signal and contributes additively to the repression of the module.

Similar decomposition of the general stress response into components is possible for the set of stress-induced genes. Two of the transcriptional modules that are activated in general stress conditions (and specifically in the 0.5M KCl experiments) are module #536 (Respiration, regulated by Hap4) and module #1215 (Gluconeogenesis). Interestingly, while the response of both modules is remarkably similar in the early phases of the osmoregulation program (0-40 minutes), only the Respiration module shows a strong secondary induction after 60 minutes (Figure 4.18. Examination of the expression of the *HAP4* gene, which is generally coupled to the module's expression level (Figure 4.14), also reveals an increase after 60 minutes, supporting the hypothesis that module #536 undergoes two consecutive inductions, one via some common mechanism (which also affects module #1215) and a second, which occurs later and is facilitated by the increased levels in *HAP4* expression. This second wave of regulation is the adaptive response of yeast cells that have recovered from the osmotic shock, in preparation for further growth.

Analysis of the behavior of module #985 (Ergosterol biosynthesis, 18x69) provides another example for the power of the integrative approach. A clear Hog1-dependent repression is observed. This result is in sharp contrast to the general ESR pattern, in which only derepression is Hog1 dependent. Previous work has shown that ergosterol-related genes respond strongly to osmotic shock [66, 112, 113]. Our analysis suggests that their repression directly depends on Hog1 through an unknown signaling pathway, that does not involve Ssk1 or Ste11.

4.9 A new paradigm for analyzing genomewide experiments

After almost a decade of microarray-based experiments, a revision of the paradigm for their computational analysis is appropriate. In the previous two sections, we introduced a method for the simultaneous analysis of new high throughput datasets given a large compendium of diverse functional data. We have shown that the integrative approach greatly extends our understanding of the regulation of biological processes and allows the decomposition of seemingly global responses into characterized regulatory programs of specific biological modules. The methodology we envision (Figure 4.19) relies on a growing compendium of public datasets and on robust algorithms for revealing biological correlations present within these data. Given the data of a new study, its integration with the large body of prior data allows us to recast the new experiments in terms of a) the behavior of already characterized modules and b) new modules that are discovered for the first time upon the addition of the new data. Using this approach, backed by appropriate community effort for modules nomenclature (e.g., based on Gene Ontologies), the results of high throughput experiments will be easier to assess and share, as it will be clear what in the new experiments is new and what confirms previously published evidence.

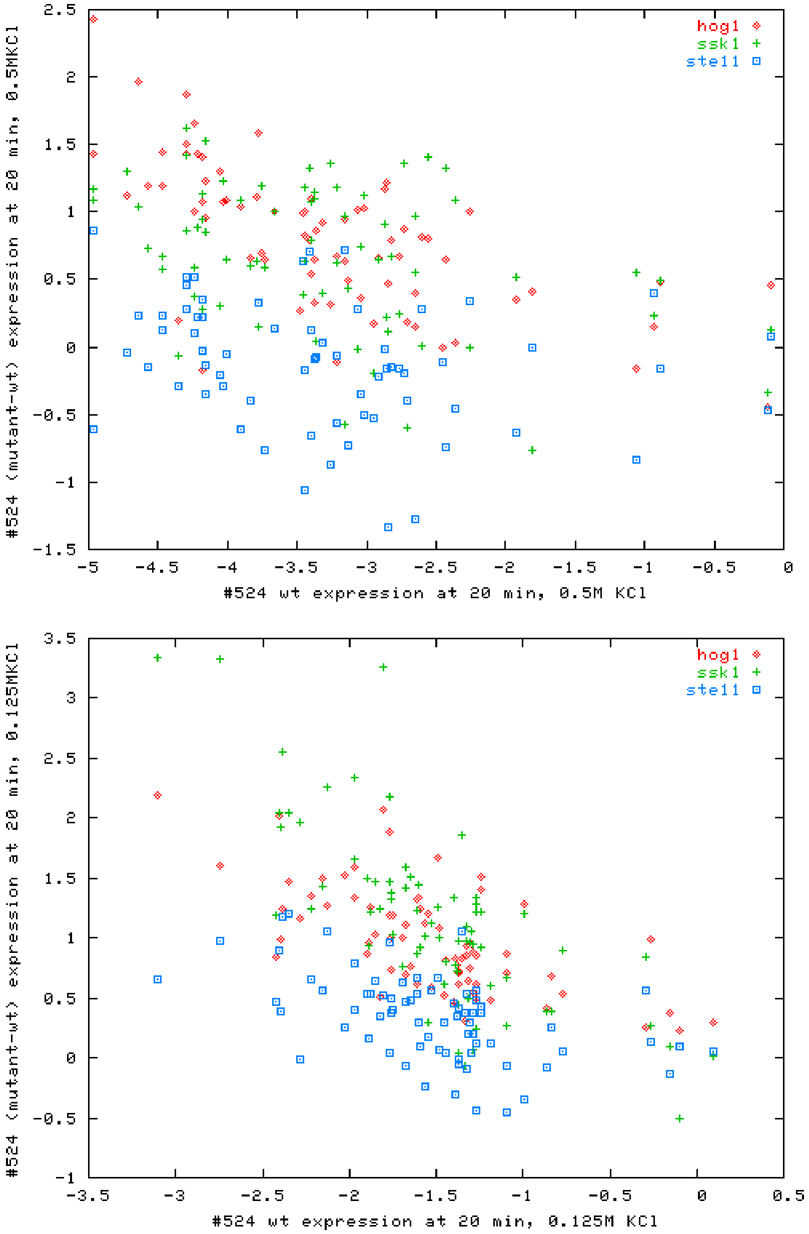


Figure 4.17: **Hog1-Ssk1 dependent repression of module #524**. We plot the change in expression of all the genes in module #524 (RNA processing) in the *wt* strain (X axis) and in the *hog1, ssk1* and *ste11* strains (Y axis), 20 minutes after treatment in KCl. The strains that disrupt the osmotic stress-specific signaling pathway (*ssk1, hog1*) exhibit reduced repression in both high (a) and low (b) KCl doses. The *ste11* strain behaves similarly to the wildtype.

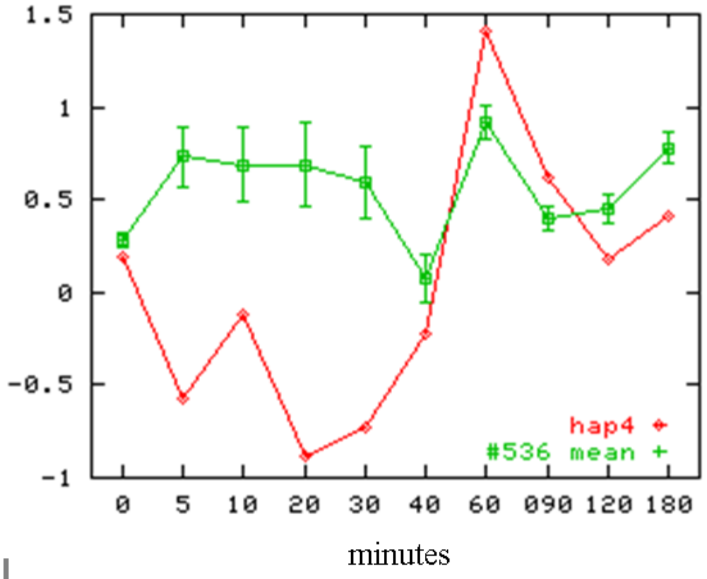


Figure 4.18: **A two-phase regulatory program for Module #536.** We show the time courses of the mean expression of module #536 (respiration) and its main regulator gene *HAP4*, when treated in 0.5M KCl in the *wt* strain. The module exhibits weak and poorly correlated induction, which is Hap4 independent, during the primary phase of the osmoregulation program (0-40 minutes). A second phase is observed at 60-180 minutes, where a tightly correlated induction is facilitated by increase in HAP4 expression.

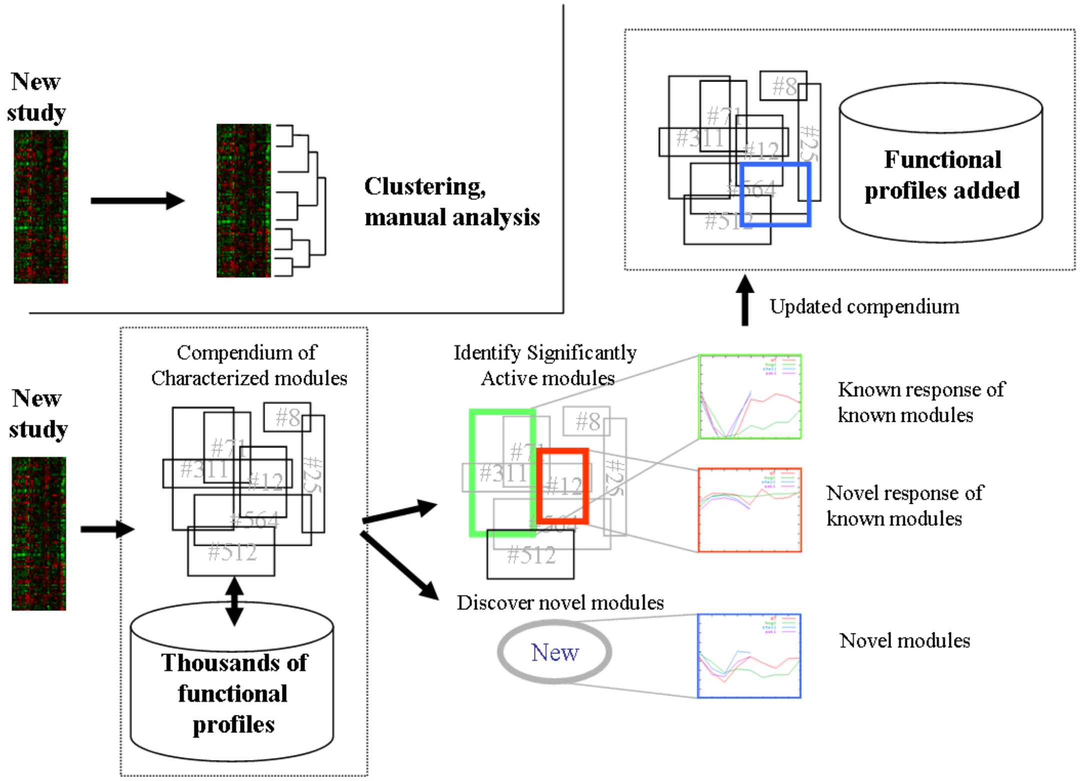

New study
Clustering, manual analysis
Functional profiles added
#8
#71
#311
#12
#25
#564
#512
Updated compendium
New study
Compendium of Characterized modules
Identify Significantly Active modules
Known response of known modules
Novel response of known modules
Thousands of functional profiles
Discover novel modules
New
Novel modules

Figure 4.19: **A paradigm for analyzing functional genomic experiments**. According to the current prevalent paradigm (top part), novel data are analyzed in isolation, typically using clustering and expert manual analysis of specific clusters. We suggest an approach (lower part) in which the community maintains the current publicly available datasets and the set of biological modules revealed by them. Modules may cover all aspects of biological processes and their regulation, as revealed, for example, by our biclustering algorithm. Using this resource, novel datasets can be represented in terms of the behavior of known and novel modules, providing an objective and transparent method for understanding, communicating and reusing high throughput data.

Chapter 5

Modeling transcriptional programs

We have seen above how to use functional genomics datasets to dissect large biological systems into simpler building blocks (functional modules). Our methods, however, could not offer a systematic way for describing the regulatory processes driving each functional module or the mechanisms differentiating them from each other. In fact, our approach to biclustering was deliberately high level, allowing integration of as much data as possible but losing much of the data's semantics. In this chapter we propose a mechanistic, multi-level model for transcriptional programs. We will introduce the model and study computational problems related to its automated learning from data. We will exemplify how to use the algorithms for learning transcriptional regulation by analyzing again the yeast galactose pathway, this time using a more detailed and systematic approach than the one used in Chapter 4.

5.1 Introduction

A transcription program can be described by the following 3-level model (compare Figure 5.1). First, certain transcription factors attain specific concentrations in specific post translational conformations. Second, as a result of sequence signals in the target gene's promoter and of the concentrations of certain TFs, the DNA in the proximity of that gene undergoes chromatin modifications and TF binding. We say that TFBSs in the gene's promoter become *active* in such cases. Third, the spatial organization of the active TFBSs and the interactions among bound TFs determine the rate of RNA polymerase recruitment and regulate the rate of transcription initiation.

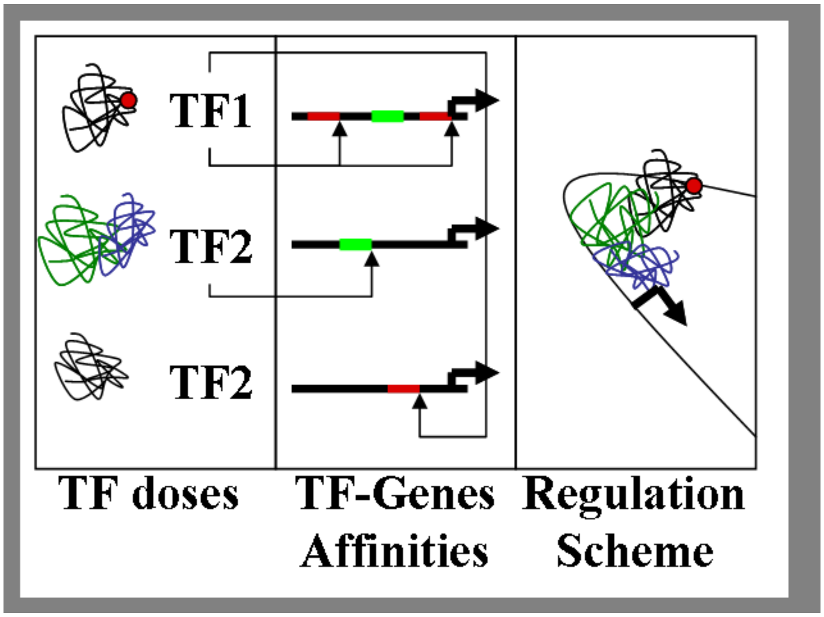


Figure 5.1: **The three components of a transcription program.** Left: TFs are transcriptionally and post translationally regulated to attain specific doses. Center: target gene promoters contain signals that enhance chromatin modifications and bind specific TFs. Right: the relative positions of TFs in their binding sites and their conformational properties induce a combinatorial regulation scheme for each gene.

The real computational challenge of building a consistent and predictive model for transcription is shaped by the nature of each of these three layers, but many of the current methods focus on one level and ignore the others. There are methods that try to model the combinatorial regulation of genes or of clusters of genes (e.g., [105, 118]) or methods that try to infer the activity of TFs [16], but true integrative modeling is still in its infancy.

In this chapter we develop a model for transcriptional programs that combines these aspects of transcription regulation. The model is constructed by representing a) TF concentrations (or doses), b) TF-gene affinities and binding site activities, and c) gene expression. The model ties the three levels together by two types of functions. It defines functions that determine binding site activity given TF doses and TF-gene affinities (these are called *dose-affinity-response functions*, and are specific for each TF). The model also defines functions that determine the rate of transcription given the activities of TFBSs in the gene promoter (these are called *combinatorial logic schemes*). In each biological condition, we infer (or use measured) concentrations of

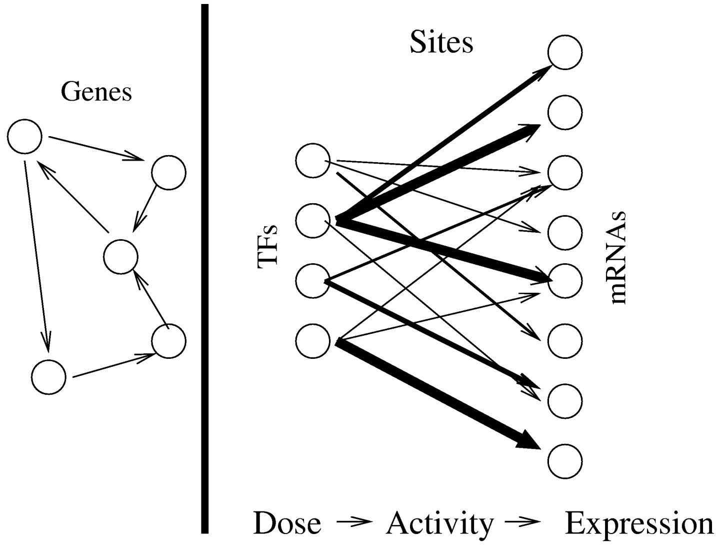


Figure 5.2: **Transcription program models**. Common gene network models (left) study a set of genes and the interactions among them. Genes and their proteins are indistinguishable and mRNA expression rates are identified with the active protein doses. Here we use a more detailed model that employs a bipartite topology (right). The model represents transcription factors and their concentrations, promoter binding sites and their affinities (edges in the graph; affinity is schematically indicated by the edge thickness) and the regulated genes and their expression rate. The structure of our model allows for integration of diverse experimental data types (expression, TF binding location) and for clear, mechanistic interpretation of the results.

active TFs. We use the dose-affinity-response functions to compute the activity of putative TFBSs at each promoter. We then use the site activities as the inputs to the combinatorial logic schemes and predict the rate of gene expression.

The chapter is organized as follows. We start by formally defining the model. We then describe algorithms for polynomial learning of dose-affinity-response functions given full data. We describe heuristics for inferring TF doses next and then show how to learn the entire model. We then discuss experiments with yeast data, building a model for the regulation of galactose related genes and showing how the integrative approach coherently combines TFs, affinities and expression into one model.

The Results in this chapter were published in [131, 132].

5.2 A model for transcription programs

We first define a model for transcriptional programs. The model is a generalization of simple genetic networks [90, 42]. In such networks, one studies a set of genes using a directed graph. The topology encodes the regulatory relations among genes, namely, the set S of genes with arcs into g are the *regulators* of g. g is also called the *regulatee* of S. For each regulatee one defines a function (logic) which determines the rate of transcription of the regulatee given the levels of its regulators. Basic network models make two critical assumptions on the regulatory system: a) the regulators states can be represented using the expression rate of the genes encoding them, and b) the relation between a potential regulator and a regulatee is a binary attribute (either there is an arc or there is none). The model we shall define and use below relaxes both assumptions in an attempt to build a model which can follow more faithfully our knowledge on transcriptional switches.

5.2.1 The topology

The transcription program model involves entities of three types. *Active TF*s represent transcription factors in their catalytically active from. *TF binding site*s represent loci at genes promoters where TFs bind. *mRNA*s represent the rate of transcription for genes in the system - the end product of the transcription program. We tie the three types of entities using a simple bipartite graph.

Definition 5.2.1 *A* **transcription program topology** *is a bipartite graph* $M = (T, V, S)$ *consisting of a set* T *of active TFs, a set* V *of genes and a set* S *of* **binding sites** *(or edges) connecting TFs and genes. We use the term* **regulators set** *of* $g \in V$*, or* $N(g)$ *to denote the set of edges adjacent to* g*, or the set of TFs adjacent to* g*, depending on the context.*

Note that we shall use V to denote both the genes and their mRNAs. For convenience, we shall use the term gene throughout for both. In each biological condition, we assume nodes and edges in M attain certain values. Active TFs are assigned **doses** (or concentrations). A binding site has a certain level of **activity** (or reaction) between the corresponding TF and gene's promoter. Genes have a **rate of transcription** (or of expression). A transcription program model is built over the topology by defining the way gene expression levels are determined once

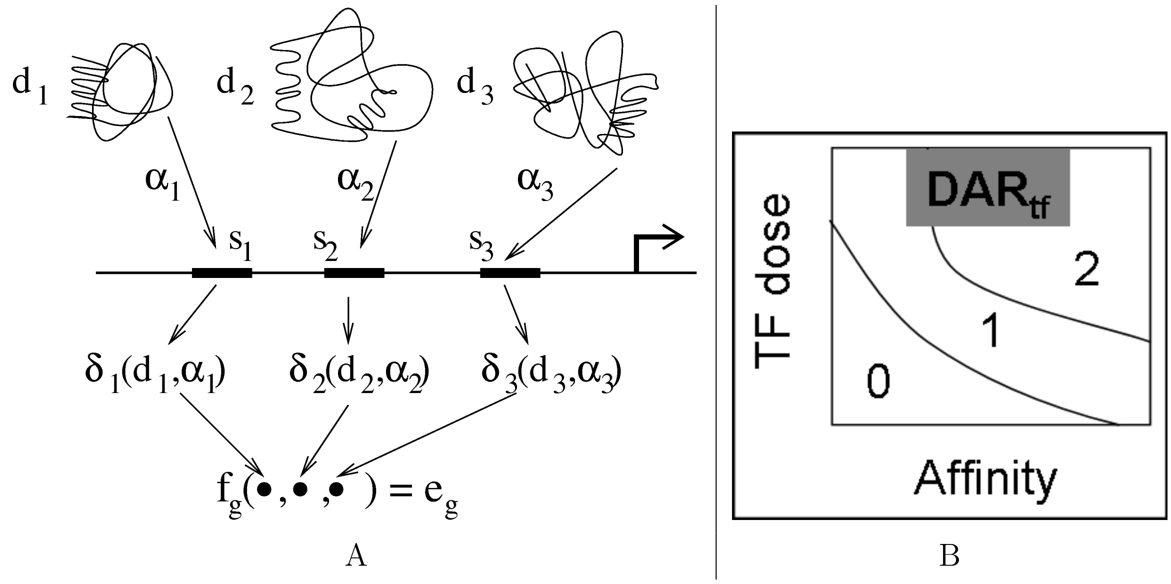


Figure 5.3: **Transcription program models.** A: The 3-layered mechanistic model of transcription regulation. The transcription factor proteins (top) attain their active states in particular doses d_i. Sites in the promoter region of the target gene g (s_i, center) allow binding of the TFs. Each site's activity is determined by the TF dose and the affinity α_i of the TF and the gene, via the dose-affinity-response function δ_j. Transcription rate e_g is then determined by these site activities via the regulation logic f_g. The site affinity is a constant value for each particular TF-gene pair. The doses - and hence the activities - vary depending on the experimental conditions. B: A typical DAR function. This function has three activity levels, denoted 0, 1, and 2. The activity level is determined as a function of the TF dose and the site's affinity. Note that the function is monotone increasing in each coordinate. DAR functions are specific to each TF, and are applied to all the TF-gene sites that include it.

their binding sites activity levels are known, and by specifying how site levels are determined once their associated TFs levels are given.

5.2.2 Scoring a topology

We first focus on the dependency between sites and genes. We assume that both site activities and transcription rates attain values in a discrete alphabet $\mathcal{C} = \{1 \ldots C\}$ and that their regulatory logic is a collection of unconstrained combinatorial functions. Specifically, the regulation of g is assumed to be via a **regulation function** $f_g : \mathcal{C}^{|N(g)|} \to \mathcal{C}$ which determines the transcription rate of g as a function of the activity levels of its regulator sites. The concrete functions will not play a major role here, but their nature is implicitly used in the model. In reality, the levels of sites are hidden variables that will be predicted by the model from TF doses. We shall temporarily assume that sites levels, in addition to gene expression levels, are measured across a set of experimental conditions and given as input. We will show how to score the fit between these input data and the model topology. We will then turn to our main problem, which is the optimization of the regulator functions.

Assume we are given a topology M and a set E of experimental conditions, where each condition $u \in U$ has a vector of gene expression levels e^u. $e^u_g \in \mathcal{C}$ is the expression level of gene g in condition u and $E = \{e^u | u \in U\}$. Suppose we also have the set A of (measured or predicted) activity levels r^u_s of each site $s \in S$ under each condition $u \in U$. A **topology score** $\phi(M, E, A)$ will be a real valued function using M and E to evaluate a site activity matrix A. Put differently, the score should assess the dependencies among site levels and gene expression levels given the topology M. ϕ is closely related to commonly used gene network scoring schemes, but with altered roles for topology and regulators values: Most network schemes score the topology, while in our setting we score the regulator activity values given a fixed topology. We call a score **decomposable** if it can be expressed as a sum of separate contributions from individual genes, each depending only on the value of the gene and its regulators sites, i.e., $\phi(M, E, A) = \sum_{v \in V} \overline{\phi}(r_{N(v)}, e_v)$. In what follows we assume that the score is decomposable.

Denote by n^v_r the number of conditions $u \in U$ in which v's regulator sites $N(v)$ attain the specific combination of activity values r (r is a vector of size $|N(g)|$). Denote by $n^{v,j}_r$ the number of conditions meeting the previous criterion which also have $e^u_v = j$. Also denote by n the number of conditions $|U|$ and by $n^{v,j}$ the number of times e^u_v equals j. We next show how several known network evaluation schemes can be used as topology scores.

- Consistency [130]. This simply sums over all genes the maximal possible number of correct model predictions: $\sum_{v \in V} \sum_r \max_j n^{v,j}_r$.
- Mutual Information [69, 104]: $\sum_{v \in V} I(r_{N(v)}, e_v)$ where $I(r_{N(v)}, e_v) = -\sum_j \frac{n^{v,j}}{n} log \frac{n^{v,j}}{n} + \sum_r (\frac{n^v_r}{n} log(\frac{n^v_r}{n}) - \sum_j \frac{n^{v,j}_r}{n} log(\frac{n^{v,j}_r}{n}))$.
- Bayesian scores [42]: We can interpret the models probabilistically and attempt at inferring a joint distribution of the sites and genes. In this case,

standard Bayesian scores (e.g., BDe) can be naturally applied.

- funcFit, or the mutual information p-value score [46]. The score equals the χ^2 value of the standard mutual information $I(r_{N(v)}, e_v)$. The χ^2 statistic is used with $(C-1)(R-1)$ degrees of freedom, where R equals the number of r vectors for which $n_r^g > 0$.

5.2.3 The Dose-Affinity-Response model

We now turn to describe the model that ties up TFs and sites. The model computes site activities given TF doses, using additional experimental information on the nature of relations between TFs and promoter binding sites. We denote by α_s the **affinity** of the site interaction represented by the edge s (recall that a binding site in our abstraction is a pair (TF,gene) so the affinity of a site actually refers to the affinity of the TF to the gene's promoter). Affinities are unbounded non negative integers and represent the strength of interaction (or biochemical reaction constant) between the DNA and the TF. Note that if several binding sites are present in the promoter of a certain gene, we will indicate this by increased affinity and not by several edges. We next define the function that computes site activities from TF doses and site affinities.

Definition 5.2.2 *A* **Dose-Affinity-Response (DAR) function** *is a non-decreasing function $\delta : Z^+ \times Z^+ \to \mathcal{C}$ computing for each TF dose and site affinity a site activity level. A* **DAR model** *is a triplet (M, α, Δ) where M is a transcription program topology, α is a set of site affinities α_s for each $s \in S$, and Δ is a set of DAR functions δ_t for each $t \in T$.*

Hence, if TF t has a DAR function δ_t, and $s = (t, g)$ has affinity α_s, then the activity of site s is determined by the dose d of t via $\delta_t(d, \alpha_s)$. Note that the affinity is a constant value per site, while the site's activity will vary (monotonically) with t's dose. The monotonicity of DAR functions means that for a fixed dose, response increases with affinity, and for a fixed affinity, response increases with dose. Suppose we are given a set of conditions U, where for each $u \in U$ we have a vector of gene expressions levels e^u and a vector of TF doses d^u. We can compute site activities r^u using the DAR functions by simple substitution:

$$r_s^u = \delta_{t_s}(d_{t_s}^u, \alpha_s). \tag{5.1}$$

Any topological score can be applied to the computed activities and the given gene expression to evaluate the fit of a DAR model to the experiments. We can now define our main model optimization problem w.r.t a topology score ϕ:

Definition 5.2.3 The DAR optimization problem. *We are given a transcription program topology M, site affinities α_s, and a set of experiments $U = (e_v^u, d_t^u)$. The goal is to find a set of DAR functions, giving rise to site activities $A = \{r_s^u | u \in U, s \in S\}$, such that the topology score $\phi(M, E, A)$ is optimized.*

A **transcription program model** is a DAR model along with concrete regulation functions for all regulatees. Given a DAR model and a set of experiments, it is easy to complete it to a transcription program model using majority voting: For the regulatee v and the combination r of activity levels of the sites $N(v)$, $f_v(r)$ is the most often observed expression rate of v when the inputs combination is r. This is the standard way by which generative gene network models are inferred once their topology is determined [104]. Note that the resulting functions will often be incomplete, as not all regulators combinations will be available.

5.2.4 Characteristics of the model

The DAR models we have introduced above serve several purposes and overcome some of the shortcomings of existing network models. Most importantly, we replace the "black box" used before to express the regulatory relations among genes with a detailed mechanistic model that incorporates new direct experimental data (e.g., TF-gene affinities) when available. The price we have to pay is the addition of hidden variables that increase the model's degrees of freedom and may lead to over-fitting. We control this effect by adding structure to the models, namely, constraints on the topology and restricted classes of DAR functions. A main structural constraint is the utilization of monotone DAR functions only. True active TFs tend to behave monotonically, while artifacts are unlikely to manifest such consistent pattern. We have observed this phenomenon in several cases, as shown in Figure 5.4. The integration of gene expression and TF affinities via the dose-affinity-response function thus models current biological understanding.

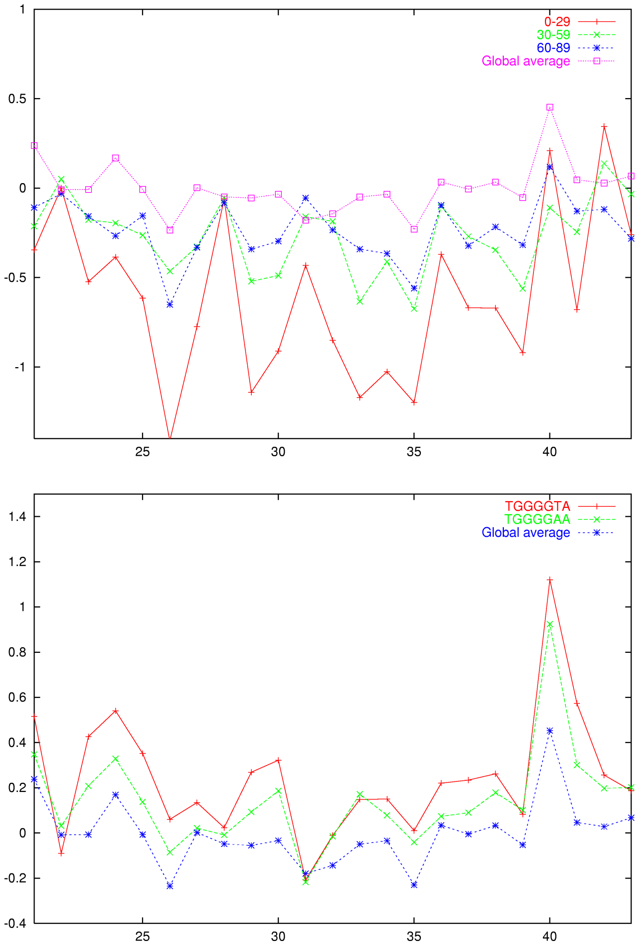


Figure 5.4: **Effects of affinity on activity.** X axis - different experimental conditions from [68]. Y axis - mean expression log fold change. Left: gcn4 bounded genes show variable response as a function of binding strength. We sorted all yeast genes by gcn4 binding affinity from [87]. We collected the first, second and third groups of 30 genes and plotted the mean expression of each group over the different conditions. The magnitude of repression in gcn4 bounded genes depends on gcn4 affinity. The analysis, although ignoring important effects such as combinatorial regulation, strongly supports the hypothesis of monotonic relation between affinity and response in gcn4 regulated genes. Furthermore, we detect multiple levels of response, underlying the importance of using more than two response types. Right: mig1 bounded genes are undetectable in [87] but exhibit similar behavior as seen by consensus based testing of mean expression. We plotted the average expression of yeast genes with the mig1 consensus TGGGGTA and its minor perturbation TGGGGAA. Both sets manifest significant induction of expression, compared to the global mean, but the exact mig1 consensus is responding stronger.

Most of the existing network learning algorithms [42, 130, 69, 104] employ discrete models. Such models are simpler computationally and enable easier statistical interpretation than in continuous (e.g., kinetic) models. High throughput experiments are inherently noisy, and thus they can rarely be translated to exact physical units. The experiments usually provide a normalized quantity which is only indicative of the actual physical quantity sought. For example, gene expression arrays results cannot be represented in absolute units of mRNA molecules per cell. Discrete models pre-process experimental information to transform a set of incomparable quantities into a small discrete set of semantically meaningful values in a small alphabet. The discretization process is known to be problematic in several respects: arbitrary thresholds must be set in advance and critical information may be lost. The model we describe here is a compromise between discrete and continuous models: Gene expression is discretized as before, but TF doses and affinities are used as ranked integers. The assumption is that although we cannot interpret most experimental results in exact physical units, we can quite safely compare them. We thus assume the DAR functions take ranked integers as inputs and output a discrete binding site activity. Using this approach, we avoid the need to arbitrarily determine a TF affinity threshold that would simplistically divide the genes into two sets of ”regulated” and ”non regulated” ones. The representation of TF doses as ordered values enables their estimation using statistical tests for hypotheses like ”a TF dose in condition X is larger than its dose in condition Y”. A similar approach to gene expression may be exploited in future work.

5.3 Model optimization

The monotonicity we impose on DAR functions is not only a biologically meaningful and powerful constraint, but also can be turned into an algorithmic advantage and enable efficient algorithms for the solution of the DAR optimization problem. We will deal with the case in which we optimize the DAR of one TF while fixing all others. We represent the dose-affinity plain as a matrix with a column for each affinity value and a row for each dose value (see Figure 5.5). Our algorithm will represent single DAR optimization as a longest path problem in an appropriate grid graph built over that matrix. The idea, in the simple case where site activities are binary, is that since DARs are assumed to be monotone, we only need to determine one dose threshold for each affinity level (or vise versa) and ensure those thresholds are non-increasing with affinity. The threshold decision at each affinity level can be scored separately (since the topology score is decomposable), and the total score for a DAR function will equal the sum of contributions from each affinity level. We next define the algorithm formally.

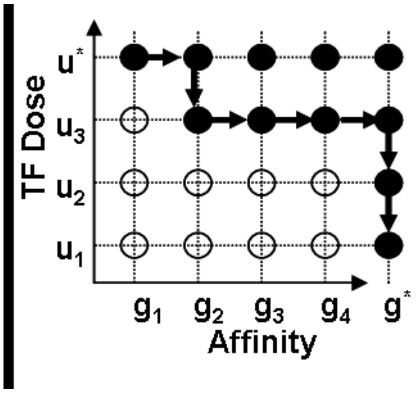


Figure 5.5: **Single DAR optimization for binary activity levels.** The grid graph represents the two dimensional dose-affinity plain of the DAR function. Arcs are directed such that paths may not go upward or to the left. A path in the graph is equivalent to a monotone DAR function, where nodes on or above the path have activity level 1 (black) and nodes below have activity 0 (white). Arc costs are set so that the longest path through the graph corresponds to an optimal DAR function.

Proposition 5.3.1 *The optimization of a single DAR function with k site activity levels can be done in $O(vu^k(log(vu^k)+k))+O(vu^k\Phi)$ time, where v is the number of genes, u is the number of conditions, and Φ is the time complexity of $\overline{\phi}$ computation. In particular, the problem is polynomial for fixed k.*

Proof: Assume first that $k=2$. We have a partial DAR model (M,α,Δ') where Δ' defines all DAR functions except δ_t which is the target of the optimization. Let $g_1 \dots g_n$ be the genes regulated by t via sites $s_1 \dots s_n$ respectively. We assume the indices are sorted by increasing site affinities $a_1 = \alpha_{s_1} < \dots < a_n = \alpha_{s_n}$. Let $u_1 \dots u_{|U|}$ be the set of experiments sorted by increasing doses of t, $d_t^1 < \dots < d_t^{|U|}$. For simplicity, we assume that all values (a_i, d_i) are distinct. We add an artificial condition $u_{|U|+1}$ and an artificial gene g_{n+1}. We build a grid digraph on the pairs (u_j, g_i) and add horizontal arcs $((u_j, g_i), (u_j, g_{i+1}))$ and vertical arcs $((u_j, g_i), (u_{j-1}, g_i))$. The horizontal arc from (u_j, g_i) to (u_j, g_{i+1}) represents the determination of δ_t in the i'th affinity level $\delta(*, a_i)$. The function is set to 0 for doses strictly smaller than d_j and 1 otherwise. Given this dose threshold, we can determine the activity vector for the site s_i and may calculate $\overline{\phi}(r_{N(g_i)}, e_{g_i})$. This score will be used as the horizontal arc cost. We set the cost of vertical arcs to 0. It is now clear that directed paths in the matrix induce monotone DAR functions (and that different DARs induce distinct paths). Moreover, for each function, the score of the model with that DAR function equals the sum of arc costs along the path that induces it. We find the optimal DAR function by solving a longest path problem in the directed acyclic graph.

To handle $k > 2$ activity levels we shall use a similar construction in higher dimension. A monotone DAR function with k activity levels divides each column in the dose-affinity plain into k segments (some possibly empty). We uniquely represent this sub-division by a non decreasing sequence $h_1 \leq \ldots \leq h_{k-1}$ where δ_t attains level i in the dose range $[h_i, h_{i+1})$ (we set $h_0 = u_1, h_k = u_{|U|+1}$). To fully define a monotone DAR function, we must determine for each affinity level a_j a sub-division $h_1^j \ldots h_{k-1}^j$ such that $h_i^j \geq h_i^{j'}$ whenever $j < j'$. We thus define a digraph on pairs (h, a_i) where h is any sub-division and a_i an affinity level. We add arcs $(((h_1, \ldots, h_{k-1}), a_l), ((h'_1, \ldots, h'_{k-1}, a_l))$ whenever $h_i = h'_i$ for all i except one index j for which $h_j = h'_j + 1$, and we set their costs to 0. These arcs are analogous to the vertical arcs in the binary case. We also add arcs $((h, a_i), (h, a_{i+1}))$ and set their costs to $\overline{\phi}(r_{N(g_i)}, e_{g_i})$ using the DAR function $\delta(*, a_i)$ induced by the subdivision h to determine site activities $r_{N(g_i)}$. We can again optimize the DAR function using the longest path algorithm. For the implementation we may use a Fibonacci heap based algorithm [41] which requires $O(M + NlogN)$ time on a graph with N vertices and M arcs. In our case $N = O(vu^k), M = O(kvu^k)$. We also have to precompute all arc costs by applying $\overline{\phi}$ once for each node in $O(N\Phi)$. The total time complexity follows. ■

Note that in practice the number of activity levels is small (we used $k = 3$ in the analysis in Section 5.5), so the exponential dependence of the complexity on k is not a practical obstacle. However, optimizing several DAR functions simultaneously is NP-hard, as we now show.

Proposition 5.3.2 *DAR optimization is NP-hard when optimizing simultaneously an arbitrary number of TFs, if consistency is used as the topology score. This is true even if the set of regulators of each gene is of size 3.*

Proof: We will show that the DAR problem (in decision version) is NP complete using a reduction from 3SAT. We are given an instance I of 3SAT with variables $x_1, \dots x_n$ and clauses $c_1, \dots c_m$. Let $c_i = (z_{i1} \vee z_{i2} \vee z_{i3})$, and let the variables that correspond to these literals be x_{i1}, x_{i2} and x_{i3}, respectively. We construct a topology with a TF for each variable and a gene for each clause. For each i, we add sites between c_i and x_{ik} for $k = 1, 2, 3$. All sites have the same affinity. We now construct the set of experiments. We use three dose levels, denoted by 0, $\frac{1}{2}$, and 1 and binary expression levels 0 and 1. For each clause c_i we add eight experiments in which all doses are set to 0 except for the doses of x_{i1}, x_{i2} and x_{i3}. Those are set so that each of the eight possible 0/1 dose combinations is covered once. We have $8m$ such experiments in total, with the property that for each truth assignment of each clause, there exists at least one experiment in which the doses of the clause variables coincide with their truth assignment values. The gene expression rates at the $8m$ experiments are defined using the clauses, as follows: For each of the eight experiments of each c_i, the expression rate of each gene c_j is set to the truth value of clause c_j when substituting the dose values (0 as false, 1 as true) of x_{j1}, x_{j2} and x_{j3} in that experiment into c_j. Precisely, let u be an experiment with doses $d_1^u, \dots, d_n^u$. For clause, set $e_i^u = 1$ iff $(d_{i1}^u \vee d_{i2}^u \vee d_{i3}^u) = TRUE$. Finally, add one experiment, denoted by u', in which all doses equal $\frac{1}{2}$ and all expression rates equal 1. The total number of experiments is thus $8m + 1$. The required bound for the total score is $(8m + 1)n$. Note that this score requires perfect consistency. This concludes our reduction which is evidently polynomial. We claim that the instance i is satisfiable iff the DAR optimization problem has solution with consistency score $(8m + 1)n$.

Assume first I is satisfiable with values $x_i = w_i$. Define a DAR solution in which $\delta_{x_i}(0) = 0, \delta_{x_i}(1) = 1$ and $\delta_{x_i}(\frac{1}{2}) = w_i$ for all i. Clearly all functions are monotone. We claim that the resulting consistency score is optimal. To see this we should prove that for each two experiments and each gene, identical site activities imply identical expression rates. This is true by construction for all pairs that do not include u', since their expression rates were determined by the clause in a consistent way. For a pair of experiments that includes u', we must ensure that for each gene, all experiments with site activities of input TFs that equals $\delta_{x_i}(\frac{1}{2})$ have expression level 1 (since in u' all expression rates were set to 1). The latter is true since $\delta_{x_i}(\frac{1}{2})$ were set to a satisfying truth assignment w_i.

To show the other direction, assume we have a solution for the DAR problem with $\phi = (8m + 1)n$. In particular, this solution specifies values $\delta_{x_i}(\frac{1}{2})$ for all x_is.

We shall use these values as our truth assignment. To see why this assignment satisfies all clauses, observe that the experiment u' gives rise for each gene to a specific site activity combination which is also present in some other experiment u'' (since we have covered, for each gene, all possible dose combinations). Since the expression value of all genes in u' is 1 and expression values in other experiments are determined by the clauses, consistency is possible only if $\delta_{x_i}(\frac{1}{2})$ satisfies all clauses. This conclude the proof. ■

The proof above would apply to another topology score if the optimal DAR solution in the reduction under that score is the same as for the consistency score. This is indeed the case for the three other scores defined in the previous section, and so the hardness result holds for those scores.

In practical settings, a heuristic to globally optimize a model is to repeatedly apply the polynomial algorithm for single DAR optimization to one TF at a time. We start with some arbitrary set of DAR functions and repeatedly select one TF, reoptimize its DAR function and add it to the current set of functions. Since single DAR optimization is solved to optimality, the new function obtained must have a score that is equal or higher than that of the previous one. Hence, the whole process is monotonically improving, and convergence to a local optimum is guaranteed. In the biological analysis reported below, our implementation converges within seconds.

5.4 Estimating TF doses

The algorithms described in the previous section rely on the knowledge of the active TF doses - the concentration of TFs in their catalytically active conformations. Since detailed measurements on protein abundance and conformation are not available yet, we must find a way to estimate TF doses. We do so by combining expression and sequence data. Our working assumption (see, e.g., [16]) is that TFs have specific DNA binding motifs, and that we can estimate the active TF doses by analyzing the expression of genes with such motifs in their promoters. We show below how to use this assumption to generate initial estimation of the TF doses. We also show how to rectify initial bias in dose estimates, by reevaluating the doses with respect to a given transcription program model, and, in later stages of the procedure, by alternating between model optimization and dose optimization.

In Section 5.4.1 we introduce an extension to motif models, and then show in Section 5.4.2 how to assign an activity value and statistical significance to such motifs. Section 5.4.3 describes an algorithm to screen and optimize active motifs . Finally, Section 5.4.4 discusses the dose optimization problem, when a model is given.

5.4.1 A location-dependent motif model

A position weight matrix (PWM) is a standard way for representing DNA motifs (see appendix 2 or [36]). A PWM P is a vector of distributions over $ACGT$. We denote by $P(i,t)$ the probability of observing character t in position i of the motif. In practice, many binding sites motifs tend to concentrate in particular regions within the promoter. To model this phenomenon, we introduce a distribution of locations for the motif: A *localized PWM* (LPWM) is a PWM with an additional location distribution p_l, where $p_l(j)$ is the probability of having the motif starting at position j in the target promoter. The likelihood of an LPWM match with a sequence s in location j is simply the product of profile probability and location probability: $Pr(P,s,j) = p_l(j)\prod_{0\leq i\leq l} P(i,s[i+j])$.

The *matching likelihood* of a string s and an LPWM P is $ML(P,s) = \max_j Pr(P,s,j)$. For each gene, we extract its promoter by taking a fixed-length sequence preceding the gene's start codon. Typical promoter lengths for yeast are 500-1000 bases. The **matching likelihood** of P with gene g, denoted $ML(P,g)$ is $ML(P,s)$ where s is the gene's promoter sequence. Denote the set of genes with matching likelihood exceeding a threshold T by $J_T^P = \{g \in V \; | ML(P,g) \geq T\}$.

5.4.2 Estimating motif activity

The standard way of assessing the activity of a putative motif given a normalized expression profile is to calculate the Z score of the mean expression taken over all genes with the motif in their promoter [16, 21]. Here, we improve this strategy by a) replacing the normalization procedure by calculation of log likelihood ratios using the random graph models introduced in SAMBA (see Chapter 3) and b) assessing the p-value of the log-likelihood ratio in order to improve the statistical significance of obtained motif activities.

For the sake of the discussion here one can view the SAMBA approach as a method for replacing the raw value in the genes- by-conditions expression matrix by new weights. The method uses a positive model and a null model. The former assigns probabilities to sets of co-expressed genes (as manifest by a bicluster). The null model, which corresponds to uncorrelated, random sets of genes, takes into account the prevalence of each gene's (and condition's) expression. Thus, the presence of a commonly-expressed gene in a set would be more probable under the null model than that of a rarely expressed gene. The SAMBA weights are defined so that the sum of weights of a set of genes S in a given condition, denoted $W(S)$, is $-log(p_+(S)/p_0(S))$, where $p_+(S)$ (resp., $p_0(S)$) is the probability of observing S under the positive (resp., null) model. Since the null model takes into account the data properties in a detailed way, one gets a more sensitive evaluation than using mean expression.

To further improve sensitivity, we use re-sampling of random gene sets to assess the empirical distribution of $W(S)$ for each condition and gene set S of size i. That distribution is not normal, and we compute its mean $\mu(i)$ and maximal deviation $\beta(i)$. We define the activity score of a gene set J and a condition j as $AS(J, j) = |W(J) - \mu(|J|)|/\beta(|J|)$. Finally, given an expression matrix on a set of conditions I, and a motif P, the **activity score** for the motif P is:

$$AS(P) = \max_{0 \leq T \leq 1} \sum_{i \in I} AS(J^P_T, i) \tag{5.2}$$

Given the threshold $Th(P)$ that maximized $AS(P)$, we can compute the activity of P on each of the conditions $i \in I$ as $AS(P, i) = AS(J^P_{Th(P)}, i)$.

5.4.3 Screening and optimization of motifs

We shall next present an algorithm for activity optimization of an LPWM. The algorithm is an EM-like heuristic with alternating phases of PWM update and gene set optimization. Similar motif refinement procedures were previously used, e.g., in multiple sequence alignment [3] and motif finding. We start from an initial LPWM P_0 (possibly random). The first phase computes the activity score of P_0 and determines the threshold T_0 for which $\sum_{i \in I} AS(J^{P_0}_{T_0}, i)$ is optimal. For each gene $j \in J^{P_0}_{T_0}$ we calculate its score contribution as $x^0_j = AS(J^{P_0}_{T_0}) - AS(J^{P_0}_{T_0} \setminus j)$. In the second phase we find for each gene $j \in J^{P_0}_{T_0}$ with positive x^0_j a position of P_0 in the promoter s_j that maximizes the matching likelihood, and use these positions as a gap-less alignment from which we extract the profiles P_1 for the next iteration. P_1 is formed by weighted counting in which gene j has a weight x_j^0, so higher activity genes have larger effect on the new PWM. We continue iterating the two phases until $AS(P_k)$ does not improve. The algorithm is described in Figure 5.6.

```
AS-EM($P_0$, $w_{ij}^c$, $s_j$):
$k = 0$
while($AS(P_k) > AS(P_{k-1})$)
        calculate $AS(P_k)$, extract optimal $J_{T_k}^{P_k}$
        For each $j \in J_{T_k}^{P_k}$ calculate $x_j = AS(J_{T_k}^{P_k}) - AS(J_{T_k}^{P_k} \setminus j)$.
        Let $Pos = \{j \in J_{T_k}^{P_k} | x_j > 0\}$
        For each $j \in Pos$ calculate $o_j = argmax_{l<|s_j|} Pr(P_k, s_j, l)$.
        Let $X = \sum_{j \in Pos} x_j$
        Initialize $P_{k+1}$ with uniform prior
        for $l = 0$ to $P_k$'s length and for each $j \in J_{T_k}^{P_k}$
            $P_{k+1}(l, s_j(o_j + l)) = P_{k+1}(l, s_j(o_j + l)) + x_j/X$
            $p_{k+1}(o_j) = p_{k+1}(o_j) + x_j/X$
        smooth $p_{k+1}$
        next $k$
return $P_k$
```

Figure 5.6: **LPWM optimization algorithm.**

We use the AS-EM algorithm as a subroutine in a platform for discovery of active motifs. We combine an exhaustive search for active k-mers and subsequent optimization of the highest scoring seeds. The combinatorial search examines all short DNA sequences with one possible gap, scans the entire set of promoters with the motif, extracts a set of genes and scores them using the AS scheme. We use a precomputed hash of short k-mers matches to speed up the procedure. The entire workflow is as follows:

1. Generate the SAMBA weights; sample random gene sets; assess likelihood distributions; sample random motif scores and determine score significance thresholds (for single and multiple testing)

2. Exhaustively screen all k-mers with a single gap of size $\leq l$
3. Use AS-EM to optimize each gapped k-mer with AS value above the random level
4. Cluster similar LPWMs and output a concise set of motifs

5.4.4 Refining transcription factor dose estimations

The initial estimation of TF doses using the expression of regulated genes may be strongly biased by effects of combinatorial regulation. For example, whenever TF A is positively regulating a set of genes and part of the set is also negatively regulated by TF B, we shall assign lower activity to A in cases where B is in action. We try to overcome such misleading initial values by tuning the TF activities given a TP model. The **TF dose optimization problem** is defined given a DAR model (M, α, Δ), input expression profiles E and a score ϕ. The goal is to find TF doses d_t^u optimizing ϕ (recall that given the DARs δ_t and the doses d_t^u we can calculate site activities r and apply the topological score ϕ). One can show that this problem is NP-hard by a simple reduction from SAT. We can heuristically approach it using hill climbing, optimizing the dose of one TF in one condition at each step. The optimal value of d_t^u, fixing the rest of the parameters, is derived by maximizing the combination of score contributions from all the model's genes: $argmax_{x\in\{d_t^{u'}, u'\neq u\}} \sum_{v\in V} \overline{\phi}(r_{N(v)}, e_v)$, where the site activities r are defined as usual 5.1.

By alternating between model optimization and dose optimization, we finally converge to a locally optimal solution. Our empirical studies show that the rough initial estimation of TF doses using motif activity scores generates a good starting point for this iterative algorithm.

5.5 Results

Here we report on the application of the framework developed in this chapter to analyze experimental data related to carbohydrate metabolism in yeast using 61 relevant expression profiles that were selected from [68, 45, 67, 92]. We applied SAMBA to transform gene expression to log likelihood weights (using probability

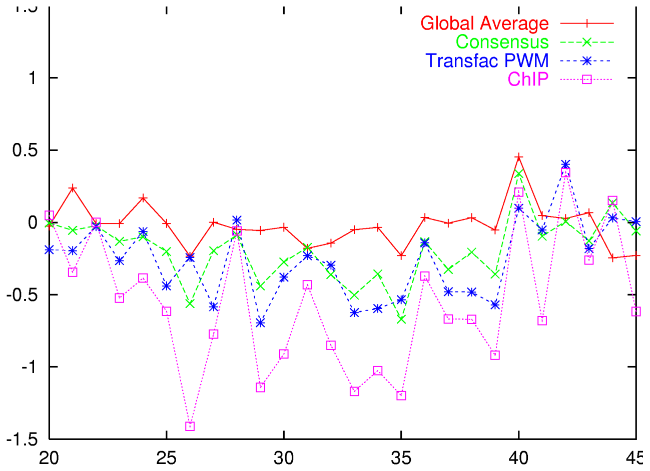


Figure 5.7: **Different sources for TF affinities.** We plot the mean expression of several groups of genes in galactose related conditions from [68]. X axis: different conditions. Y axis: Mean log expression ratio. The ChIP group consists of the 30 genes with strongest binding to gcn4, according to location analysis [87]. The Transfac PWM group consists of the 30 genes with highest matching likelihood of their promoter to Transfac PWM M00038. The consensus group consists of all yeast genes with the exact motif TGACTCA in their promoter. Since all groups manifest significant co-expression compared to the global mean, all these sources of information may be used as affinities to some extent. However the direct physical measurements outperform other methods.

$P_c = 0.6$ for the positive model, see Chapter 3). To assess p-values of weights we randomized 10000 gene sets of each size from 1 to 2000 genes and calculated their total weight. We used 500 bases upstream from the start codon as the promoters.

5.5.1 Comparing activity score and mean expression

To compare the performance of the activity score to the mean expression methods we used the 32 yeast PWMs from Transfac [149] version 5.1. Following [70], the mean-based score of a set of genes $G \subset V$ was computed as follows: Each condition in the expression matrix was normalized to mean 0 and standard deviation 1. Let s_c^G be the mean expression of genes in G on condition c. Let s^G be the mean value of s_c^G over all conditions $c \in C$. The collection of conditions with significant mean is then extracted as $S = \{c \in U || s_c^G - s^G| \geq \theta\sigma\}$ where $\sigma = \frac{1}{\sqrt{|G|}}$ and θ is a parameter.

The mean score is finally defined as:

$$MeanScore(G) = \sum_{c \in S} (|s_c^G - s^G|) \tag{5.3}$$

For each PWM we ordered all yeast genes by the matching likelihood of their promoter to the PWM. We selected an optimal threshold for maximal activity score (Formula (5.2)). To determine significance, we used the maximum score obtained over 5000 instances in which the genes' promoters were randomly shuffled. The same process (selecting an optimal threshold and sampling) was repeated with MeanScore. As can be seen in Figure 5.9, both methods correctly identified the GAL4 and RAP1 sites as active, but only the activity score succeeded in identifying additional transcription factors which are known to be functionally associated with carbohydrate metabolism (MIG1, ADR1 and more).

5.5.2 Effect of motif localization on activity

We tested the utility of incorporating motif location distribution into the PWM by re-analyzing TRANSFAC's known PWMs on the same dataset using different positioning windows. We tested the activity of each PWM restricted to windows of 100bp starting at -500 with steps of 50bp and ending at 0. Some of the motifs (but not all) show higher activity in specific windows (Figure 5.8). Using LPWM can improve the specificity of TF-gene association and refine our understanding of the relation between DNA and TFs.

5.5.3 Discovery of active motifs in the galactose system

To infer a transcription program for the galactose system we first applied the active motif discovery algorithm to the dataset of [68]. This is a subset of 23 conditions out of the 61 analyzed above. We first repeated the screening with 32 TRANSFAC motifs as in Section 5.5.1. On this dataset, only GAL4, GCN4, and MIG1 were detectable as significantly active, in accordance with biological literature on the galactose pathway. We then screened all 6-mers with one gap of size 0-12 bases and optimized all hits with scores above the noise level. The active motifs are listed in Table 5.1. We accurately rediscovered de-novo the known motifs GAL4, GCN4 and ADR1/MIG1. We also discovered three additional motifs (CANCCCC, TT(9N)CCCC, CCG(5N)CCG) which achieve scores above or equal to ADR1/MIG1.

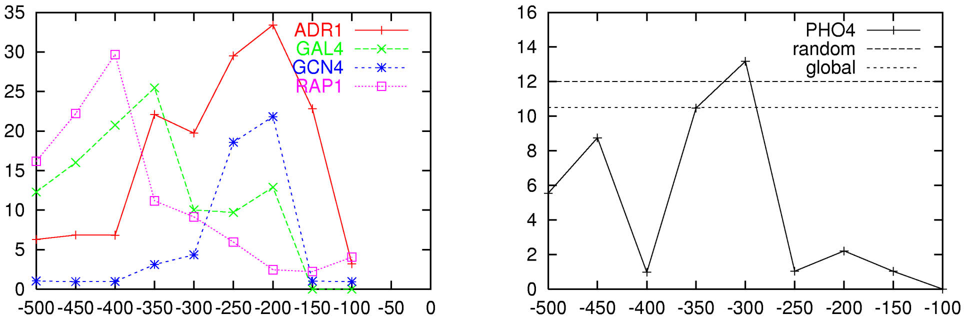


Figure 5.8: **Effect of motif position on activity.** Plots of activity as a function of the range window. X axis: distance of the left endpoint of the 100-bases window from promoter TSS. Y axis: AS score. The left figure shows the behavior of several known motifs: RAP1 is strongly biased to -400 bases from the TSS, GAL4 to -350, GCN4 to -200. In contrast, ADR1 is active in a broader range. The right figure shows the global and localized score for PHO4. The peak of the local score at -300 bases exceeds the detection level, while the global score is below it.

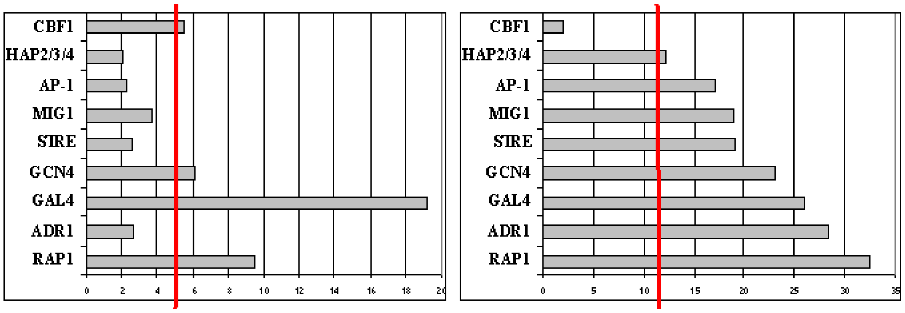
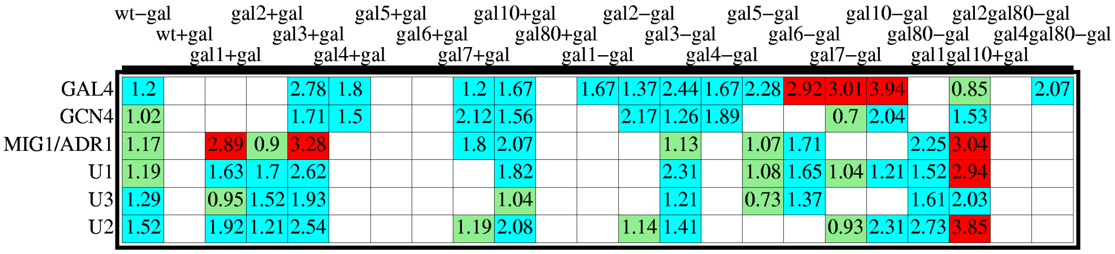


Figure 5.9: **Comparing the activity score to mean expression.** 32 yeast PWMs from Transfac were scored against a combined carbohydrate data set. Left: Mean scores. X axis: Mean score. Only RAP1, GAL4 and possibly GCN4 and CBF1 are detectable above the noise level (vertical bold line). Right: Activity scores. X axis: Activity score. Additional relevant active motifs (ADR1, MIG1, STRE, AP-1) are detected.

Figure 5.10 shows the activity profiles of all inferred motifs. We observe four transcriptional sub-programs: a) GAL4 regulated genes are repressed in many of the tested conditions, most dominantly when galactose is absent from the media or when upstream regulators (GAL3) are manipulated. b) GCN4 repression is limited to GAL4 active conditions, but seems to be independent of the level of galactose (active in GAL3,4,7,10 knockouts, both when galactose is present or absent). This may be explained by the fact that GCN4 regulates amino acid biosynthesis pathways that in part depend on the yield of the TCA cycle, which in turn may be dependent on the galactose pathway. c) MIG1/ADR1,U1 and U3 are active together and may represent a more general cellular response to stress. U1 and U3 have a common sub-motif (CCCC) which also appears in MIG1 complement. When comparing the gene sets associated with each of the three motifs, only the sets of U1 and U3 have statistically significant intersection (30% of the genes). All three motifs may be related to the STRE site (CCCCT-AGGGG), but the STRE motif is not active itself. A particularly high activity of these motifs is observed in the gal1-gal10-double mutant, in which GAL4 response is weak. Interestingly, this activity seems to be a combination of the activity in the single mutants gal1- and gal10-. d) The U2 motif contains a fragment of the GAL4 sequence, but its gene set does not significantly intersect that of GAL4. Its activity does not manifest either galactose dependent or galactose independent pattern.



	wt−gal	wt+gal	gal1+gal	gal2+gal	gal3+gal	gal4+gal	gal5+gal	gal6+gal	gal7+gal	gal10+gal	gal80+gal	gal1−gal	gal2−gal	gal3−gal	gal4−gal	gal5−gal	gal6−gal	gal7−gal	gal10−gal	gal80−gal	gal1gal10+gal	gal2gal80−gal	gal4gal80−gal
GAL4	1.2				2.78	1.8			1.2	1.67		1.67	1.37	2.44	1.67	2.28	2.92	3.01	3.94		0.85		2.07
GCN4	1.02				1.71	1.5			2.12	1.56			2.17	1.26	1.89			0.7	2.04		1.53		
MIG1/ADR1	1.17		2.89	0.9	3.28				1.8	2.07				1.13		1.07	1.71			2.25	3.04		
U1	1.19		1.63	1.7	2.62					1.82				2.31		1.08	1.65	1.04	1.21	1.52	2.94		
U3	1.29		0.95	1.52	1.93					1.04				1.21		0.73	1.37			1.61	2.03		
U2	1.52		1.92	1.21	2.54				1.19	2.08			1.14	1.41				0.93	2.31	2.73	3.85		

Figure 5.10: **Activity profiles of galactose-related motifs.** Rows represent the six binding site profiles that we detected. Columns represent different experimental conditions from [68] and are labeled using the naming convention of that study.

5.5.4 A dose-activity response model for the galactose system

We have used the six active motifs discovered above to build a model explaining the gene expression of all the variable genes in the 23 conditions in the data set of [68]. We selected 310 genes with significant expression change in the data set (at least 3 conditions with over 2-fold change in expression). We assigned affinity levels based on TF binding location data from [87] for GAL4 and GCN4. For the four other motifs, including MIG1, we used the LPWM matching likelihood. (The location data on MIG1 in [68] could not be used, since that experiment took place in normal growth, and in such condition MIG1 has low activity). We calculated initial TF doses using the activity scores of the motifs in each condition. We sorted affinities and doses of each TF and used the ranking as inputs for the inference algorithm. For model optimization, site activity and gene expression were assumed to have three possible values. We optimized DARs using consistency [130] for the topology score. Figure 5.11 shows the results of the optimization with consistency scores.

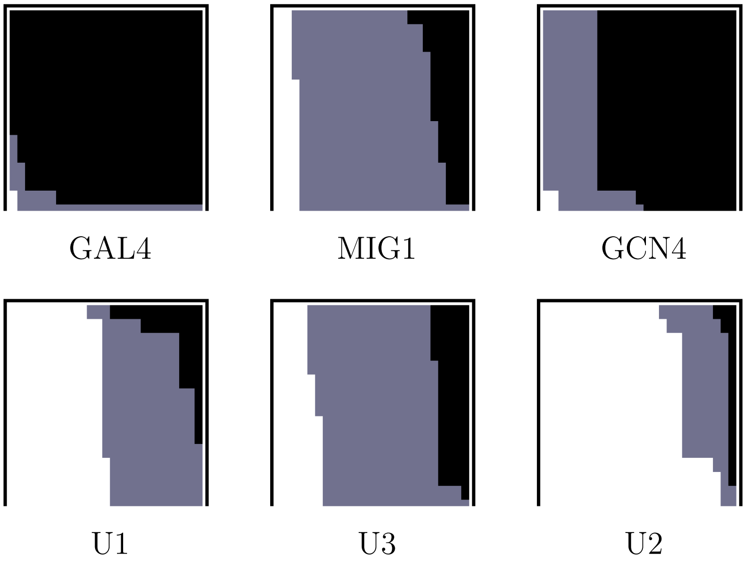


Figure 5.11: **Transcription program model DAR's for the galactose TFs.** X axis: ranked gene affinity. Y axis: ranked TF dose. The colors represent the response (or derived TF activity). Darker color indicates stronger response.

The optimized transcription model also constructs, of course, for each site, a concrete affinity level and concrete activity levels under each condition, as well as a specific regulation function for each gene. A fragment of the model is shown in Figure 5.12. We selected for easy visualization only those genes that are regulated by GAL4 according to the model. A gene is considered regulated by a TF if its affinity level is above the threshold selected to maximize the TF's activity5.2. The figure shows TF-gene interactions, where genes with an identical set of regulators are grouped together. The graph represents the predicted combinatorial regulation schemes involving GAL4. For example, the important regulatory gene GAL3 is combinatorially regulated by GAL4, MIG1 and GCN4, PUT4, a proline permease, is controlled by GAL4, U1 and U2 . Interestingly, there are 19 genes regulated by

GAL4 alone, 34 genes regulated by two TFs, 23 genes regulated by 3 TFs and only 3 regulated by four TFs. No gene is regulated by five or all six. This indicates both the power of combinatorial regulation to generate complex control using a few regulators, and its efficiency in terms of the number of regulators used.

To test the utility of our integrative approach for TF binding location and gene expression data we compared the results of the DAR optimization process when given the experimental TF location profiles for GAL4 and GCN4 and when using the LPWM matching likelihood instead. Overall we obtain improved score for the complete model (using all six TFs) when using the experimental location profiles (model score without location data: score=5657, with GAL4: score=5687, with GCN4: score=5687, with both: score=5717).

To further test the robustness of the framework, we repeatedly selected one location profile and one TF, and used that profile as the genes affinities for that TF. The rest of the model was initialized with LPWM matching likelihood and we recorded the score after optimizing the complete model. This was repeated for all 660 possible TF-profile combinations (for 6 TFs and 110 profiles). We found that the score reached for the only experimental profiles that were available (GAL4 and GCN4) was not matched by any of the other experiments (p=0.001). This shows the superiority of location profiles over computed ones, and also demonstrates the robustness of our methodology. Further experimentation based on this model may lead to improved understanding of the way in which the transcription program of galactose related genes respond to different stimulations.

Shortly after the initial publication of our results on the galactose system in yeast, a comprehensive comparative study was published by Kellis et al. [79]. In this study conservation across four yeast species was used to detect functional signals in promoters. Significantly, one of the new binding motifs detected by this study was identical to the motif U2 that we had reported.

LPWM Consensus	**AS Score**	**Remark**
CGG(11N)CCG	30.8	GAL4 consensus
TGACTCAWT	22.57	GCN4 consensus
TGGGGTA	22.03	ADR1/MIG1 consensus
CANCCCC	26.92	Unknown, denoted U1
CCG(5N)CCG	26.6	Unknown, denoted U2
TT(9N)CCCC	22.03	Unknown, denoted U3

Table 5.1: De-novo identification of active motifs in the galactose system. Using our framework for active motif discovery, we identified three known and three putative binding site motifs with statistically significant activity. The results match the literature in predicting activity of GAL4, GCN4 and MIG1. Putative sites represent predictions that extend the model of galactose related transcription program.

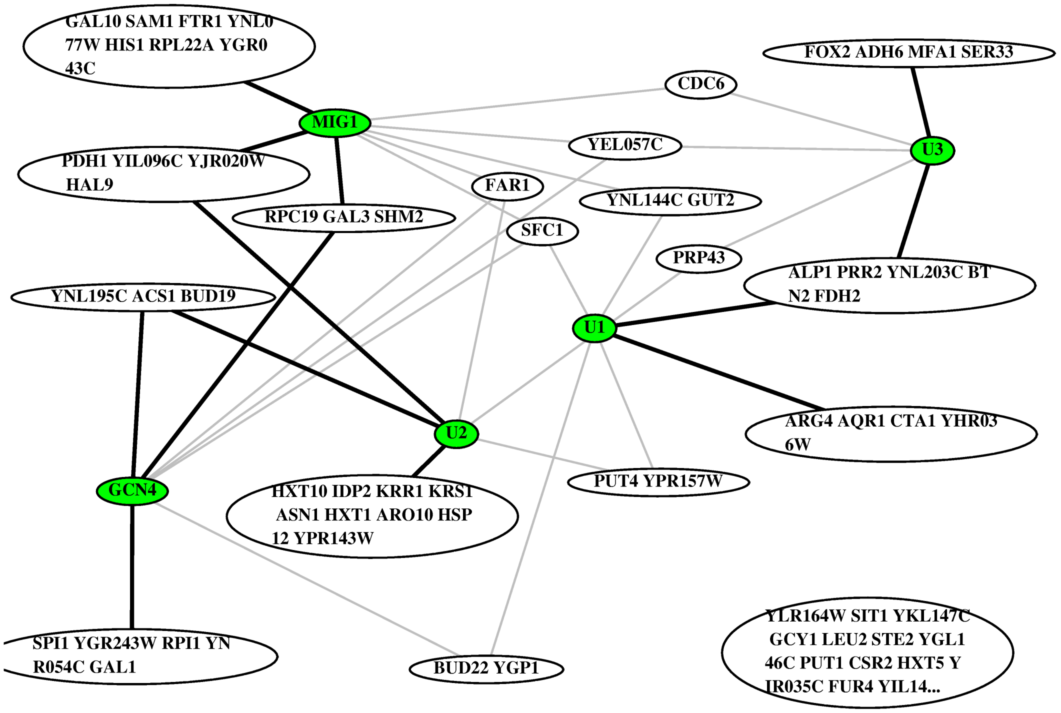

GAL10 SAM1 FTR1 YNL0 77W HIS1 RPL22A YGR0 43C
FOX2 ADH6 MFA1 SER33
CDC6
MIG1
YEL057C
U3
PDH1 YIL096C YJR020W HAL9
FAR1
YNL144C GUT2
RPC19 GAL3 SHM2
SFC1
PRP43
ALP1 PRR2 YNL203C BT N2 FDH2
YNL195C ACS1 BUD19
U1
U2
ARG4 AQR1 CTA1 YHR03 6W
PUT4 YPR157W
GCN4
HXT10 IDP2 KRR1 KRS1 ASN1 HXT1 ARO10 HSP 12 YPR143W
YLR164W SIT1 YKL147C GCY1 LEU2 STE2 YGL1 46C PUT1 CSR2 HXT5 Y IR035C FUR4 YIL14...
SPI1 YGR243W RPI1 YN R054C GAL1
BUD22 YGP1

Figure 5.12: **The Galactose system**. A part of the transcription program model reconstructed by the DAR methodology for the galactose system. Having optimized the DAR model, only genes that have affinity above the activity threshold for a particular TF are considered to be regulated by it. Regulation is indicated by an edge. Genes with identical set of regulators are grouped together and shown within the same oval. Thicker edges underline larger groups of genes. The figure presents only those genes that are regulated by the TF GAL4. Filled ovals: TFs. White ovals: genes. The large set of genes in the oval at the right bottom contains some of the genes that are regulated only by GAL4, according to the model. Numbers in the cells are the activity scores. Darker cell shades indicate higher scores, and blank cells represent lack of significant activity. Note that high activity may be due to either induction or repression with respect to the wild type state. The direction of activity is not shown in the figure (but can be easily checked using the original expression values of the genes regulated by each motif).

Chapter 6

The evolution of TF binding sites

As we saw in the previous chapters, the interaction between transcription factors and their DNA binding sites is a central element of gene regulation. Recall that transcription regulation is mediated by TFs that bind short specific sequences (denoted in this thesis as TFBSs) upstream of the regulated genes. Sequence specific TFs can physically recognize a limited repertoire of sequences variants, and each variant may have a different binding affinity. The functional effect of small variations in the binding sequence are known to play a role in transcription control [27, 75], and as we have shown, differences in binding affinities can be exploited when constructing a mathematical model for transcriptional programs. So far, our approach to the analysis of transcriptional regulation was based on experiments that tested the behavior of a single model system (*S. cerevisiae*) in a set of conditions. By analyzing the similarities (e.g., expression patterns, regulatory sequences) and differences in the response of many genes to similar environments, we inferred models for transcriptional regulation. In this chapter, and in the next one, we shall take a different approach, using evolutionary analysis to compare the function of transcriptional programs in several species. We shall focus here on the evolutionary dynamics of TFBSs and on the relations between TFBSs' function and the selective pressures affecting their evolution.

Results from this chapter were published in [128] in collaboration with Irit Gat-Viks.

6.1 Motivation

The natural diversity of living organisms suggests itself as a rich source for information on the function of regulatory networks in general and transcriptional programs in particular. Instead of probing the behavior of a single system in different conditions and perturbations, it is possible to study several different systems (species) and examine their evolutionary similarities and differences as a way to infer their function. The evolutionary paradigm provides us with structures (e.g., phylogenetic trees) and principles (e.g., parsimony) by which we can interpret diverse data on the phenotypes and genotypes of many species. When applied to transcriptional networks, we have the advantage of using well defined genomic loci (TFBSs) as genotypes that we can relatively easily correlate with the gene expression phenotypes.

The common approach to comparative genomics of regulatory regions [55, 124] highlights the identification of *conserved sequences* as a way to distinguish between functional and non-functional parts of the genome. According to this methodology, we summarize all our evolutionary insights into binary tagging of the sequence (conserved, unconserved). While this approach offers simplicity and was proved to be highly effective in revealing novel functions in genomes (e.g., small RNA genes [8], ultra conserved elements [10]), it is expected that much more can be inferred from the evolutionary relations among species. As we shall see below, the evolutionary dynamic of TFBS sequences reveals a rich structure of selective pressures that can be correlated with function in higher resolution than possible using conservation-based analysis.

6.2 The selection network

6.2.1 Standard models for neutral evolution

In this section we develop a mathematical representation for the evolutionary relations between short DNA sequences (including TFBSs sequences and non-functional short sequences). We start with a brief background on molecular evolution and the neutral model for sequence evolution (for the standard textbook introduction see [89]). We shall assume that in the course of evolution, one DNA sequence is transformed into another by a series of independent substitutions of single nucleotides. We will assume that other events causing insertion or loss of DNA segments are much less frequent and will neglect them throughout. The neutral theory of evolution models the evolutionary process as a combination of two processes: mutation and fixation. Mutations are continuously being introduced into the population due to DNA damage and inaccuracies in the transmission of genetic materials from generation to generation. The neutral hypothesis states that most of the mutations have small or no effect on the organism's fitness (these are *neutral* mutations). Nevertheless, in a population of a given size, a considerable fraction of the neutral mutations would become fixated (i.e., present in all or most of the population), due to pure chance. On the other hand, mutations that have negative effect on the organism's fitness have very low fixation probability. In these cases, we will say that the mutation was under *negative selection.* The rate of substitution at each neutral position is modeled using a *substitution matrix* defining the probability of substitution from each nucleotide to each nucleotide over a unit time interval. In its simplest form (the one parameter model) the substitution matrix describes the rate of mutation without distinguishing which nucleotides are being substituted. In a more realistic version we use different rates for transitions (A↔G, C↔T) and transversions (A,G↔C,T, the *Kimura two-parameters model* [81]) or even set individual rates for each substitution (a sixteen parameter model). To sum up, it is widely assumed that neutral sequences (sequences that do not contribute to the organism's fitness) in a specific lineage evolve independently at each nucleotide according to a roughly fixed substitution matrix.

Given the DNA sequences of two species we can estimate a 16 parameters substitution matrix matching the evolutionary period between them by aligning the sequences and counting aligned nucleotide pairs (assuming the vast majority of sequence evolve neutrally). In probabilistic terms, this procedure can be interpreted as finding the maximum likelihood substitution model for the aligned sequences. When we have more than two species, we will use a phylogenetic tree to describe the evolutionary relations among them. According to the common maximum likelihood approach, one can infer a substitution matrix and phylogenetic branch lengths so that the likelihood of the aligned sequence is optimal (using, e.g., an EM algorithm [36]). We will take an alternative approach and estimate a substitution matrix separately for each branch in the phylogenetic tree. This approach is reasonable since we will analyze relatively few species (less than ten) using whole genomes (millions of loci for each species). Using the maximum parsimony assumption we shall infer the sequences of ancestral species and estimate a substitution matrix for each branch, as is done in the case of two species. The mutation independence assumption, together with estimated substitution matrices over a phylogenetic tree, provide us with a simple background model for the evolution of neutral sequences. The applications we will develop below test how well the neutral model can predict the observed sequences, and identify regions in which it fails to do so as targets for closer functional analysis.

6.2.2 Selective pressures on TFBSs

Sequences that are functional are evolving differently than neutral ones, being subject to *selection*. The most easily detectable and widespread type of selection is *negative selection*: a process that reduces the fixation probability of mutations that decrease the organism's fitness. Negative selection can be detected by comparing aligned sequences in several species and estimating the conservation rate along the alignment. Nucleotides that are more conserved than expected by the neutral model can be hypothesized to be under negative selection and therefore functional. Comparing observed sequences to a neutral model is the essence of today's comparative genomics [79, 55, 124].

Assume that we are observing a set of aligned regulatory regions and that these regions are mostly neutral, except for TFBSs that are relatively sparse. Also assume that in the phylogeny of species we are analyzing, there exists a set of TFs that are completely conserved, and that each TF is binding specifically a distinct and disjoint set of short DNA sequences (the same set in all species). Finally assume that binding of a TF to a regulatory region occurs at any locus that contains one of the sequences the TF recognizes. We can thus summarize all we have to know on TFs and their targets in a simple code, grouping short DNA sequences according to the TF that binds them. Under these simplified assumptions (which are clearly weak approximation of the reality, but will serve us in the following discussion), we would like to use comparative genomics to discover the code from aligned sequences. If this can be done, we will be extracting more information than mere tagging of sequence as functional or non functional.

Now consider the selective pressures acting on a nucleotide in the regulatory region, under our simplified assumptions. If the nucleotide is neutral (i.e., does not participate in any TFBS) then no selection should be observed. If the nucleotide is a part of a TFBS, the selective pressure on mutations involving it would be dependent on the family of sequences that the TF can bind. Mutations that change the nucleotide but keep the TFBS functional and identifiable by the same TF would behave neutrally. Mutations that compromise the function of the TFBS (by changing its identity or completely eliminating it) should be selected against. If we analyze short DNA sequences (or more specifically, k-mers), instead of single nucleotides, we can predict that in general, substitutions between two k-mers that belong to the same family should be neutral and substitutions that crosses family boundries should be selected against. By estimating the selection acting on each of the possible substitutions between k-mers we can group k-mers to neutrally interchangeable families and reconstruct the regulatory code.

In practice, our simplified functional model for TFs will only be approximately correct. First of all, many of the loci we shall analyze will not be functional, due to chromatin structure or other factors that affect the TF-DNA interactions and are not directly coded to the TFBS sequence. Second, the actual regulatory code is much more complex and less well defined than the cluster model we introduced above. Third, TFBSs may have different lengths which can be shorter or longer than k. The total selective pressure on each k-mer would therefore be a combination of possible effects on parts, suffixes or prefixes of the k-mer. As we shall see below, grouping k-mers into families will still be possible, even though our assumptions are gross simplifications of the biological reality.

6.2.3 Estimating selection on k-mers

We focus on changes in words of length k, and use the term *substitution* here to mean a change of a single base in a word. We estimate the selection on substitutions between k-mers by comparing the observed number of substitutions to the number of substitutions predicted by the neutral model. Assume first we have two species and that their sequences are aligned. We denote the multiplicity of each DNA k-mer m in the first species as $n_1(m)$. The expected number of substitution from m to m' (denoted as the *predicted count*) would equal $n^{pred}_{m,m'} = Pr(m \rightarrow m')n_1(m)$, where $Pr(m \rightarrow m') = \prod_i Pr(m[i] \rightarrow m'[i])$ according to the neutral model. On the other hand, the observed number of substitutions equals the number of aligned k-mer pairs m, m' which we denote as $n_{m,m'}$. We next define:

Definition 6.2.1 *The* selection ratio *of a substitution $m \rightarrow m'$ is defined as $\rho_{m,m'} = log(n_{m,m'}/n^{pred}_{m,m'})$. The* conservation ratio *for a k-mer m is defined as $\rho_{m,m}$.*

ρ values that are significantly smaller than 0 suggest negative selection, but we note that these values are very sensitive to the sample size $n_1(m)$ and cannot be used directly. Given our background neutral model we can model $n_{m,m'}$ as a binomial random variable that equals the sum of $n_1(m)$ Bernoulli variables with probability $Pr(m \rightarrow m')$. We can therefore compute a p-value for rejecting the neutral hypothesis on the substitution $m \rightarrow m'$ using the binomial distribution: $Pr(b(n_1(m), Pr(m \rightarrow m')) < n_{m,m'})$ (b being the binomial distribution).

Now assume that we are given aligned regulatory sequences $s_1, \dots, s_n$ for more than two species, and that the phylogenetic relations among the species are defined by the tree $T = (S, V, par)$ where $S = \{s_1, \dots, s_n\}$ are the extant species, V are the ancestral species and $par(v)$ defines parent-child relations in the phylogenetic tree. We again wish to estimate the selection on substitutions $m \rightarrow m'$. To that end we will reconstruct ancestral sequences and use the formula we developed above for the case of two species over each of the phylogenetic branches. To reconstruct ancestral sequences we may use any of the standard methods (parsimony or maximum likelihood). In the results reported below we used the maximum parsimony approach [36] to determine for each ancestral node and each position on the alignment the set of maximum parsimony nucleotides $i \in V, s_i[h] \subseteq \{A, C, G, T\}$. Note the $s_i[h]$ may contain more than one character if the parsimony solution is not unique. Given the most parsimonious sets of characters, we used a simplistic uniform probability model and define the probability of observing a k-mer m at the ancestral species i in position h as the product $Pr(s_i[h \dots h+k] = m) = \prod_j \frac{1}{|s_i[h+j]|}$ if $m[j] \in s_i[h+j]$ for all $j < k$ and 0 otherwise.

To estimate the predicted and the observed number of substitutions between each pair of k-mers we combine information from all of the branches. The contribution of a phylogenetic branch (i, i') to the number of observed substitutions between $m \rightarrow m'$ equals $n^{i'}_{m,m'} = \sum_h Pr(s_i[h \dots h+k] = m)Pr(s_{i'}[h \dots h+k] = m')$ where h is running over all aligned positions. The number of predicted $m \rightarrow m'$ substitutions is computed by using a background model for the branch $i \rightarrow i'$ and applying it to the expected number of m appearances in the specie i: $n^{i',pred}_{m,m'} = (\sum_h Pr(s_i[h \dots h+k] =$

$m)) * Pr(m \rightarrow m'|i \rightarrow i')$ where $Pr(m \rightarrow m'|i \rightarrow i')$ is the mutation probability of $m \rightarrow m'$ on the branch $i \rightarrow i'$. By adding up the observed and predicted number of substitutions across all phylogenetic branches we compute the $n_{m,m'}$ and $n^{pred}_{m,m'}$ statistics for the entire phylogeny. We can compute their ratio $\rho_{m,m'}$ as in the case of two species. We can also generate binomial p-values as before, although in this case we will have to treat a sum of several binomial variables with different parameters (each branch have different probability for $m \rightarrow m'$).

We can now define the selection network, for a fixed value of k:

Definition 6.2.2 *The* **selection network** *is a weighted directed graph* (S, E, ρ)*, containing a node for each DNA sequence* k*-mer and an arc from each node to each other node at hamming distance one. The arc weights* $\rho_{m,m'}$ *equal the substitution selection ratios defined above.* ρ *is also extended to include the conservation ratios.*

6.3 Estimating the *Saccharomyces* selection network

To compute the selection network for the *Saccharomyces* clade, we used promoter alignments of the four sensu stricto species *S. cerevisiae, S. mikatae, S. kudriavzevii* and *S. bayanus* [23], and the established phylogenetic relations among them. For each promoter we reconstructed the ancestral sequences using maximum parsimony as described above. We estimated the background model by counting single nucleotide substitutions at each phylogenetic branch. We used $k = 8$ throughout. We counted the number of octamer substitutions and built the selection network by computing for each edge the ratio $\rho_{m,m'}$. Our analysis was genome-wide, including all aligned octamers in each of the aligned promoters (up to 1000bp upstream the gene, about 2 million loci in total). We ignored octamers that were aligned with gaps while estimating the background model and while counting substitutions. Our analysis was therefore implicitly focused on generally conserved regions. We ensured that divergent promoters were treated only once (choosing the strand arbitrarily).

The global distributions of estimated substitutions ratios $\rho_{m,m'}$ and conservations ratios $\rho_{m,m}$ are shown in Figure 6.1. For conservation there is a clear bias toward increased ratios, reflecting k-mers that are functional and are more conserved than expected by the neutral model. For substitutions, we see the complementary effect on substitutions that have negative ρ values, suggesting negative selection on mutations that change functional sites.

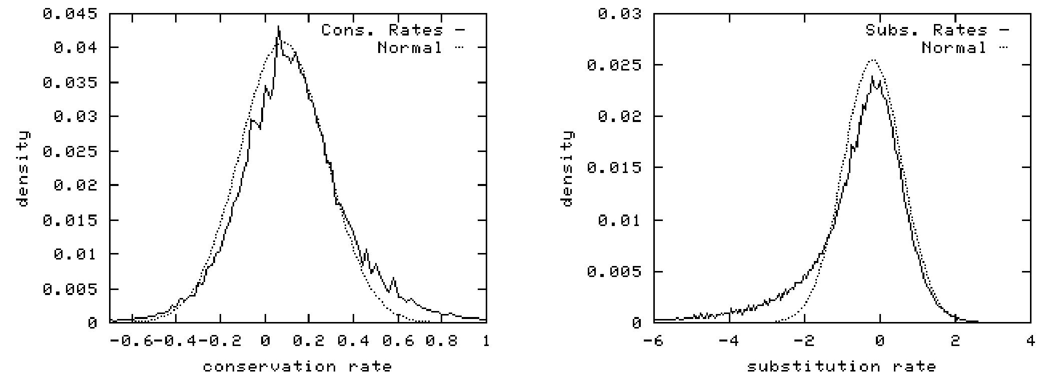


Figure 6.1: **Global distributions of substitution and conservation ratios.** The distribution of conservation ratios for the 65536 motifs (A) shows a normal-like form that is disrupted by a significant peak of motifs ($P << 10^{-10}$, G-test for goodness of fit) that are conserved more than expected. The distribution of substitution ratios for more than 1.5 million neighbor motif pairs (B) is enriched ($P << 10^{-10}$, G-test for goodness of fit) in substitutions that appear less than expected.

6.4 Analyzing reverse complementing substitutions

Recall that we have motivated the construction of the selection network by simplified assumptions on the relations between k-mers and TFBSs. These assumptions may or may not be an adequate formalization of the structure and function of TFBSs. Moreover, we constructed the network by compiling information for numerous loci, most of which are not likely to encode for a function or may encode for a function that does not match our TFBS model. The reliability of the estimated selection ratios on network edges should therefore be carefully assessed. One possible approach is to compute binomial p-values for the observed $n_{m,m'}$ as described above, but this approach cannot control for systematic errors in the background model or for inaccuracies in the estimation methodology. We shall next describe an alternative method using comparisons of reverse complementing substitutions.

It is known that transcription factors can bind both strands of the DNA and that in many cases TFBSs on both strands can be equally functional. Since we analyze data on the 5' strand only, and since each k-mer in the 5' appears in reverse complement on the coupled 3', the estimated ratios of reverse complementing substitutions can be argued to serve as two independent observations on the same physical quantity (under the assumption that motifs in both strands are functional). Given a k-mer m we denote its reverse complement as m^c. The reverse complementing substitution of $m_1 \to m_2$ is $m_1^c \to m_2^c$. The correlation between the ρ_{m_1,m_2} and $\rho_{m_1^c,m_2^c}$ is shown in Figure 6.2. The high correlation supports the hypothesis that motifs in both strand have similar functions in many cases. As expected, when the sample size (which we quantify using $n^{pred}_{m_1,m_2}$) increases, the estimated ρ values are more accurate and the correlation between reverse complementing substitutions increases. A similar effect can be seen by comparison of the conservation ratios $\rho_{m,m}$ and ρ_{m^c,m^c} (Figure 6.3).

We can thus use the differences between ratios of reverse complementing substitutions to assess the estimation errors for each $\rho_{m,m'}$. We partition all k-mer pairs into bins of similar sample size $b_{low} < n^{pred}_{m_1,m_2} \leq b_{high}$. Assume that for all motif pairs m_1, m_2 in a particular bin, we have $\rho_{m_1,m_2} = \rho'_{m_1,m_2} + \epsilon_{b_{low},b_{high}}$ where $\rho'_{m_1,m_2} = \rho'_{m_1^c,m_2^c}$ is the real selection ratio for the substitution and $\epsilon_{b_{low},b_{high}}$ is a normally distributed error term which is the same for all k-mer pairs in the bin. Given these assumptions we can estimate $\sigma(\epsilon_{b_{low},b_{high}})$ by computing the variance of the distribution of $\rho_{m_1,m_2} - \rho_{m_1^c,m_2^c}$ for pairs m_1, m_2 in the bin (Figure 6.4). The error term standard deviation is estimated as $\sqrt{2}\sigma$ where σ is the standard deviation of the difference distribution. We define the standard deviation for the substitution ratio of m_1, m_2 as the standard deviation of the error term of the appropriate bin. We denote it by σ_{m_1,m_2}.

6.5 Functional families in the selection network

Having constructed the selection network, given our estimated selection ratios and their confidence levels, we shall next examine the topology of the network and associate it with the function of TFBSs. The most prominent structure that we shall seek in the selection network is the partition into clusters, in accordance with our very basic model for TFBSs evolution. We define *functional families* as sets of conserved k-mers that are neutrally substituting with each other.

Definition 6.5.1 *Given a selection network with selection ratios and their error rates (S, E, ρ, σ), we define a* k-mer family *as a connected component in the subnet-*

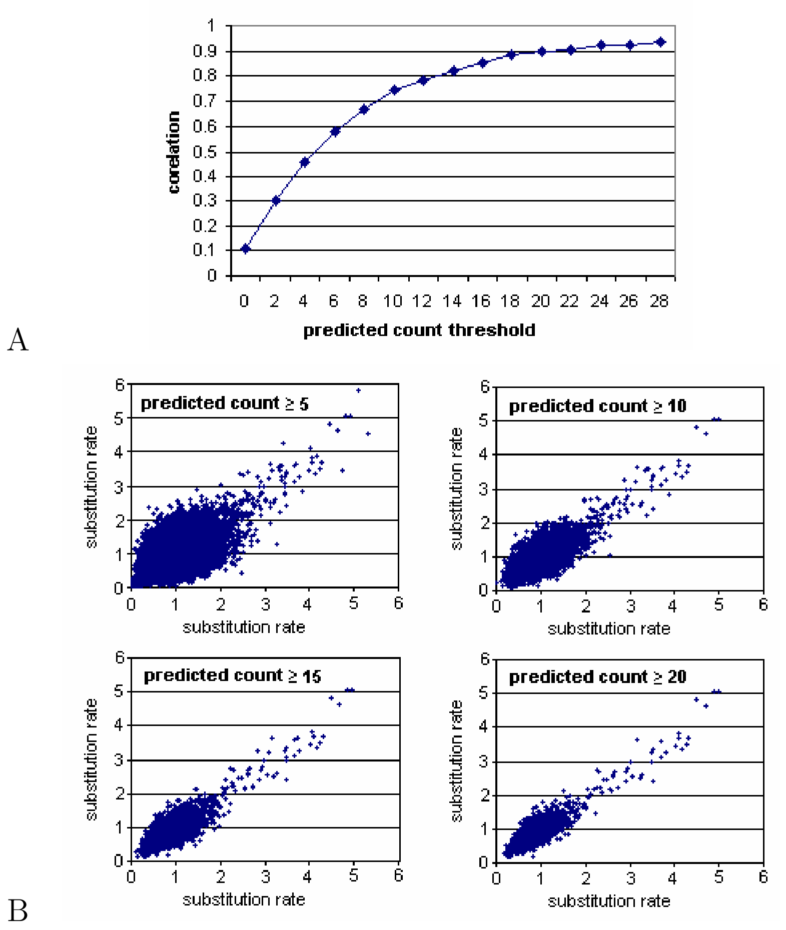


Figure 6.2: **Supporting substitution ratio estimations by analysis of reverse complementing substitutions.** The substitutions that change k-mer A to B and the reverse complement (RC) of A to that of B represent similar alteration of regulatory function. The ratios of such RC substitutions are estimated independently and can be used to assess the reliability of the statistics and the specificity of selective pressures. A) Pearson correlation between RC substitution ratios, as a function of $n^{pred}_{m,m'}$ threshold. For each threshold value, the correlation is computed only for substitutions with predicted count exceeding that threshold. B) Scatter plots for RC substitution rates (exponentiated *rho* values) for different thresholds. The increase in correlation as a function of the threshold reflects reduced noise levels when sample sizes for individual RC substitutions are larger. For large $n^{pred}_{m,m'}$ the estimated ratios are very specific.

work including only k-mers m for which $\rho_{m,m} - z_c\sigma_{m,m} > C_1$ and edges m, m' for which $\rho_{m,m'} - z_s\sigma_{m,m'} > C_2$, where z_c, C_1 are the conservation parameters and z_s, C_2 are the neutrality parameters.

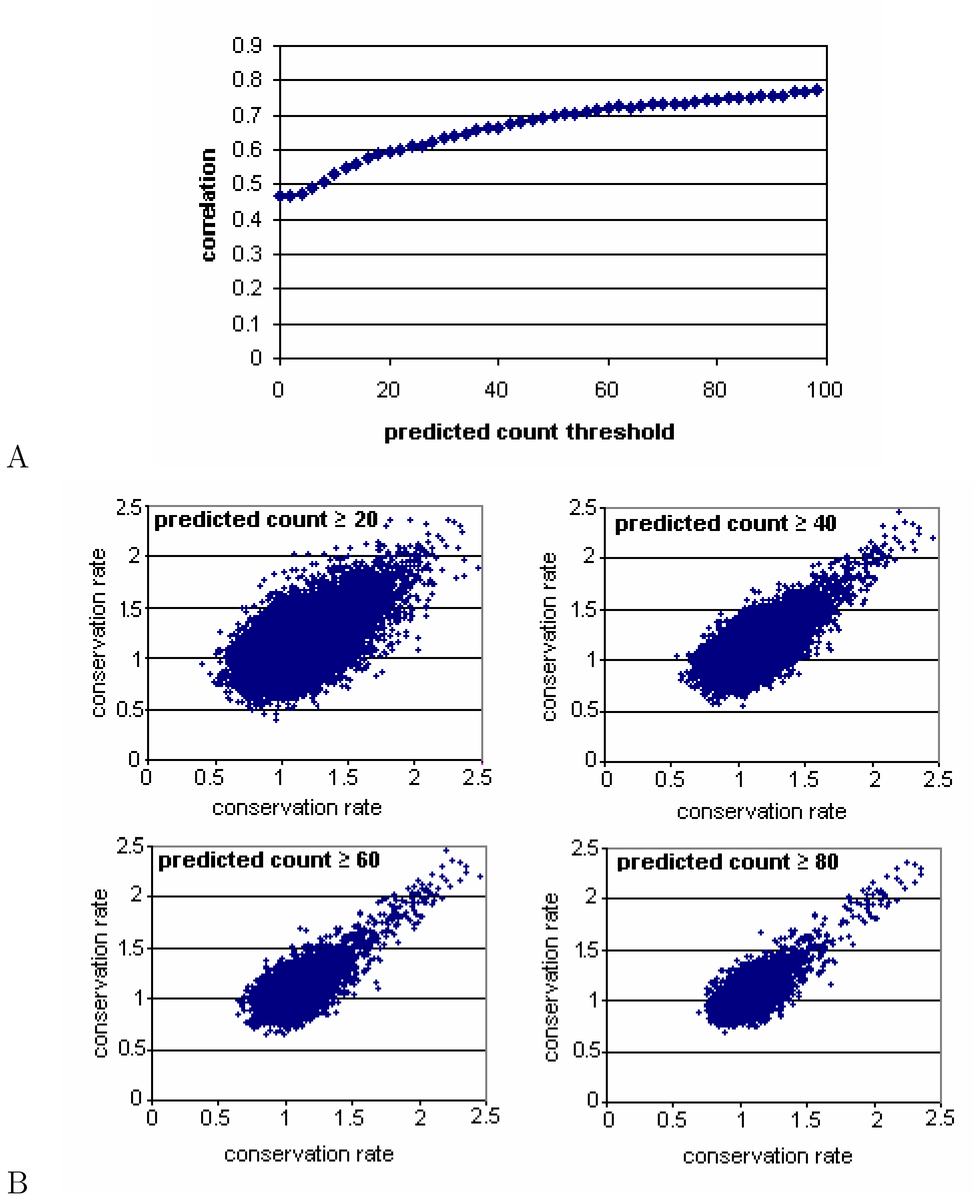


Figure 6.3: **Conservation ratios of reverse complementing k-mers.** A) Pearson correlation of conservation ratios, as a function of the $n^{pred}_{m,m}$ threshold. For each threshold value, the correlation is computed only for conservations with predicted counts exceeding that threshold. B) Scatter plots for conservation rates (exponentiated *rho* values) for different thresholds.

Under these definitions, searching for families is an easy computational problem (identifying connected components in a graph).

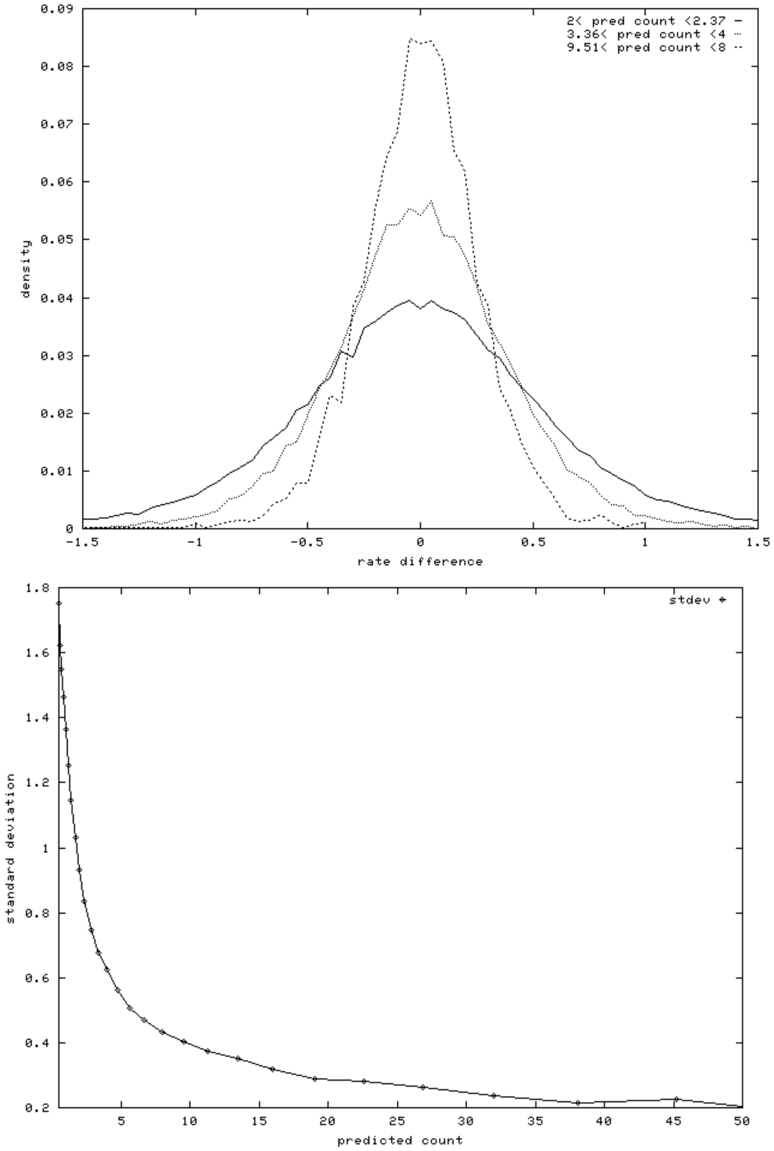


Figure 6.4: **Distribution of differences between reverse complementing substitutions.** (A) The distribution of differences between ratio estimations of reverse complementing substitutions with predicted counts in three different intervals. The variance in these normal-like distributions is approximately twice the estimation variance of individual ratios (see text). (B) Variance of estimation for increasing predicted count ranges. The estimation variance is decreasing with the increase in predicted count.

We searched for k-mer families in the yeast selection network (setting $z_c = z_s = 2, C_1 = 0.2, C_2 = -0.5$) and disregarded all single-vertex components. The results are detailed in Table 6.1 and in Figure 6.5. In principle, families could be artifacts

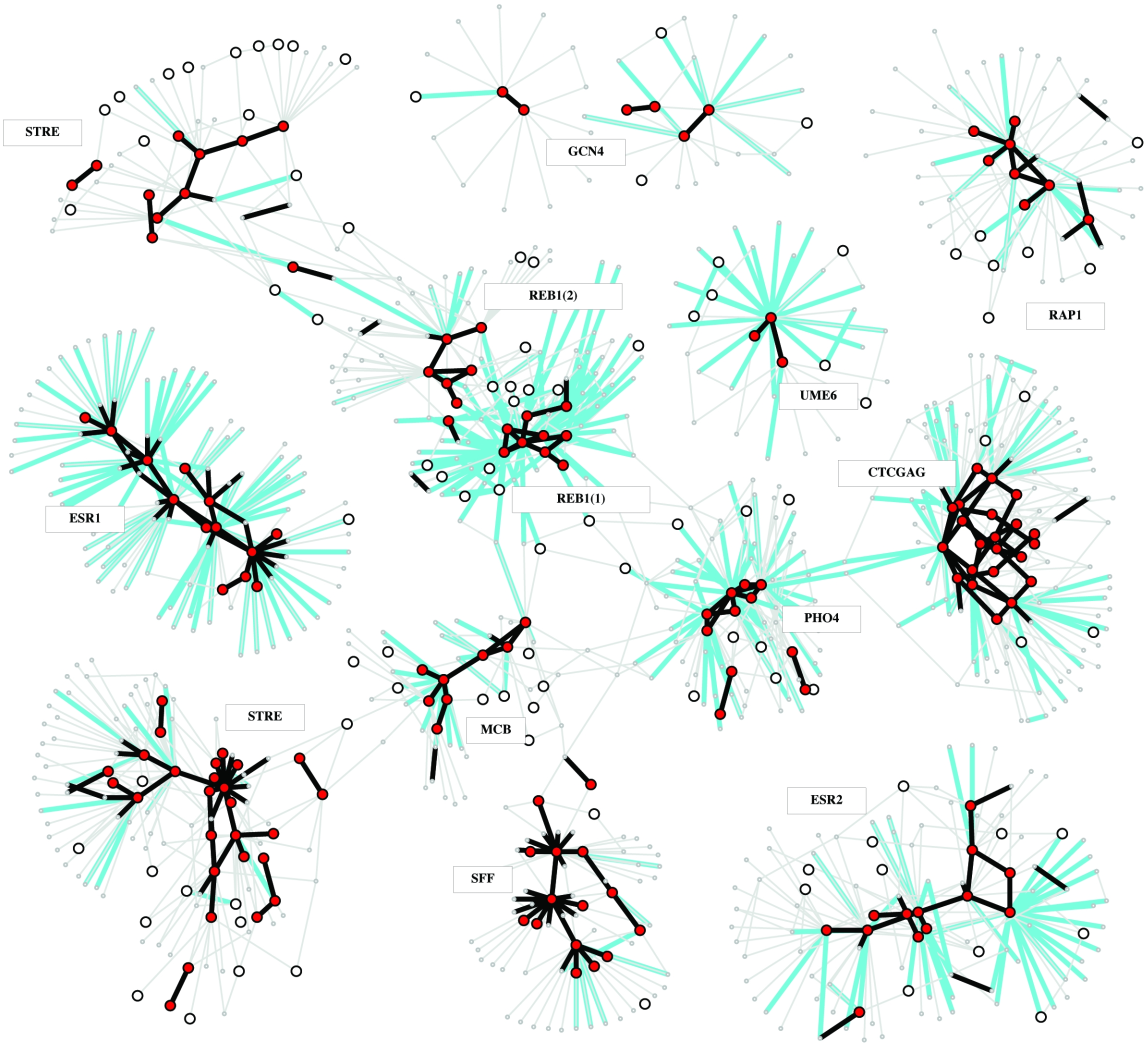


Figure 6.5: **Partial view of the selection network.** k-mers are represented as circles; k-mers that have $\rho_{m,m} - 2\sigma_{m,m} > 0.2$ are drawn as large circles, k-mers with conservation ratio $\rho_{m,m'} - 3\sigma_{m,m} > 0.2$ are shown as large filled circles. Substitutions are shown as color coded arcs. Black: non-negative substitution ratio ($\rho_{m,m'} > 0$ and $\rho_{m,m'} - 2\sigma_{m,m'} > -0.5$); cyan: negative ratios ($\rho_{m,m'} + 2\sigma_{m,m'} < -0.5$). A family in the selection network is a cluster of k-mers that are interconnected via high or neutral substitution arcs. Low ratio substitution arcs separate families. Arc directions are not shown for readability, but 91% of the negative ratio arcs point from a family k-mer to a k-mer outside the family. We annotated each family by a consensus motif, and by the names of the transcription factors that match that consensus (if such are known). Many known transcription factors are identified as families of k-mers, some of which are shown in this figure. Families can be further analyzed for intricate intra-family relations. Certain transcription factors (e.g., Reb1) correspond to more than one family suggesting multiple functionalities as discussed below. Transcription factors with binding sites that are less frequent in the genomes (e.g. Gcn4) have larger variance on ratio estimations, so are harder to cluster robustly.

resulting from errors in the estimation of selection ratios or from more basic problems with the neutral background model. To control for these, and to assign putative function for families, we tested if the association among k-mers from the same family could be supported by information on function of yeast genes. We define for each k-mer m a set of genes G_m including all genes bearing the k-mer in their promoter at least once. For a k-mer family $F = \{m_i\}$ we form the union of all gene sets $G_F = \cup_i G_{m_i}$. For each k-mer family we test for functional support using the following sources of information:

- Enrichment of G_F sets in ChIP targets: We tested directly if G_F sets were enriched in ChIP targets of specific TFs (using data from [87]). This was done by testing G_F independence (using hyper geometric p-value) from the sets of conservative ($p < 0.001$) and permissive ($p < 0.01$) TF targets. p-values were corrected for multiple testing using the conservative Bonferroni factor.

- Co-expression of G_F sets: We used a collection of 980 gene expression profiles (see Chapter 4) to test for each cluster F if genes in G_F are over-expressed or under-expressed in each of the gene expression profiles. We identified, for each of the expression conditions, the sets of up (expression over 1SD above the average) and down (expression under 1SD below the average) regulated genes, and computed their hyper-geometric p-values of independence from G_F (again corrected for multiple testing).

- G_F GO functional enrichment: We tested if G_F genes are enriched with genes known to be involved in specific processes by computing hyper-geometric p-values for independence of G_F and sets of genes with a particular GO annotation (see appendix A).

Using these three sources of information we can show that over 90% of the k-mer families can be associated with some function or TF (Figure 6.6), confirming that even given our assumptions and thresholds, we are still able to identify groups of motifs that are functionally related. Importantly, using the above data we can also suggest the functions that may be regulated by k-mers from different families, and the TFs that bind them (in cases where ChIP profiles are associated with a family).

The way we define and create motif families does not follow the common approach of identifying groups of binding sites by conservation and consensus. Here, instead,

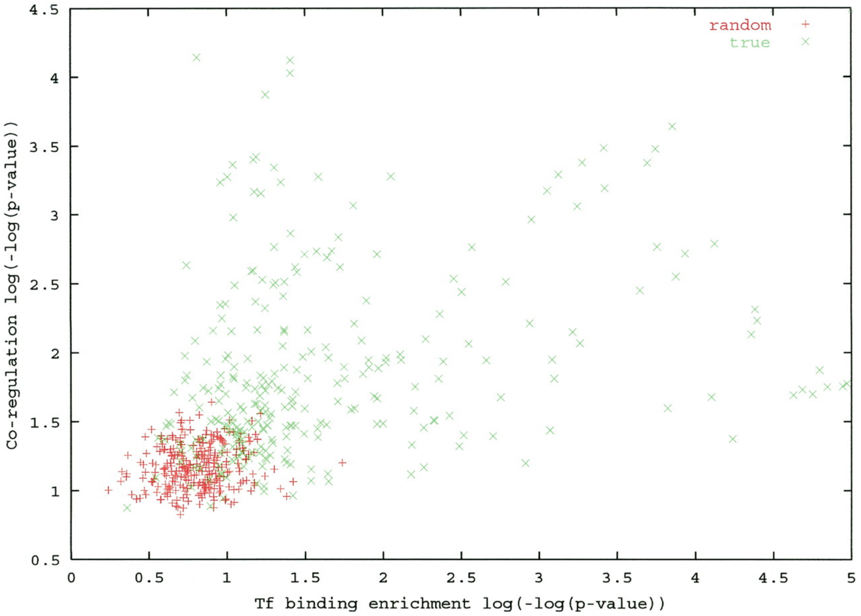


Figure 6.6: **Functional enrichment of genes associated with motifs families**. Each point represents a motif family. For each family, the set of genes that contain a motif from the family in their promoter was identified. The p-value of the best match to a ChIP experimental profile (x coordinate) and the highest co-regulation achieved in a collection of gene expression experiments (y coordinate), were computed. The vast majority of true motif clusters (green) are strongly supported by functional data, and are well separated from the respective p-values computed with the same motif clusters but randomly shuffled promoters-genes association (red).

it is the substitution rates between motifs that determine the families. Using this approach, we can identify motifs that were previously hidden by nearby stronger consensus sites. For example, the motifs cluster TCTCGAGA (Figure 6.5 consists of 11 motif variants that resemble the known PHO4/CBF1 cluster CACGTG but are well separated from it by negatively selected substitutions. The sets of genes with motifs from each of those families have totally different expression profiles (t-test, $p < 0.0002$). Interestingly, the TCTCGAGA motif set manifests strong down co-regulation ($p < 10^{-11}$) in a *gal80* and *gal1* knockout strains grown without galactose [68], supporting its possible role in transcription regulation (recall our analysis of these strains in Chapter 4). An additional example of a putative family is GGTRATGR with a possible role in the regulation of ribosome biogenesis ($p < 10^{-23}$). See Table 6.1 for more details on functional families and their annotations.

Cluster Consensus Name	Comments	Size	ChIP	P	GO Annotation	P	Expression	P	U/D
RTTACCCG	REB1 related motifs	3	REB1	-144.1	proteolysis and peptidolysis	-8.5	Natarajan01 GCN4C/ GCN4 (R4760/R6257)	-5.9	up
TTACCCT*		4	REB1	-69.3	metabolism	-6.6			
**CCGGGT		10	REB1	-45.7	protein catabolism	-5.4			
PHO4 CACGTG**	Pho4 motifs	9	CBF1	-81.1	sulfur metabolism	-15.6	Ogawa00 PHO4c vs wild type	-9.3	up
AATCACGT		5	CBF1	-60.5	sulfur amino acid metabolism	-7.4	Gasch00 Amino acid adenine starvation 2 h	-5.3	up
ESR1 A*CTCATC	ESR (PAC) and related motifs	11			ribosome biogenesis	-53.8	Gasch00 1 5 mM diamide 20 min	-63.0	down
ESR1 CTCATCGC		2	ABF1	-4.1	ribosome biogenesis	-37.1	Gasch00 1 5 mM diamide 20 min	-56.1	down
RAP1 GYATGGGT	RAP1/FHL1 motifs	6	RAP1	-61.8	macromolecule biosynthesis	-17.8	Natarajan01 WT +/- 100mM 3AT (Set C) (KNY164)	-16.2	up
CCGTGCAT		2	FHL1	-48.3	macromolecule biosynthesis	-17.5	Gasch00 Heat Shock 000 minutes series 2	-12.8	up
BAS1 GTGACTCA	GCN4/BAS1 motifs	2	GCN4	-47.2	amino acid biosynthesis	-23.8	Hughes00 top3 haploid	-38.1	up
BAS1 GTGAGTCA		2	GCN4	-30.5	amino acid biosynthesis	-26.1	Hughes00 top3 haploid	-32.6	up
SWI6 WRACGCGT	SWI4.SWI6 and IME5 motifs	6	MBP1	-42.3	DNA replication and chromosome cycle	-21.7	Spellman98 alpha factor release sample016	-32.3	up
ATTTCGCG		2	SWI4	-30.6	DNA replication and chromosome cycle	-6.3	Spellman98 CLN3 induction 30 minutes	-24.3	up
*RAACGCG		9	MBP1	-26.5	DNA metabolism	-13.3	Spellman98 CLN3 induction 30 minutes	-29.3	up
MSN2 MSN4 CMCCCCTT	STRE moitifs	7	HSF1	-7.8	carbohydrate metabolism	-5.2	Gasch00 heat shock 17 to 37 20 minutes	-26.5	up
MSN2 MSN4 TAAGGGGT		2	HSF1	-4.1			Gasch00 heat shock 21 to 37 20 minutes	-17.5	up

GTGGCGAA	RPN4 and proteolysis related motifs	2	REB1	-6.1	proteolysis and peptidolysis	-27.3	Gasch00 1 5 mM diamide 60 min	-21.4	up
TCGCCACC		2			protein catabolism	-23.8	Gasch00 1 5 mM diamide 60 min	-25.4	up
YTGTTTAT	fkh1/2 motifs	6	FKH2	-24.9	cell cycle	-7.9	”Robertson00 bni1D 50 nM aF, 120 min log10(ratio)”	-8.6	dow
FKH1 FKH2 TGTTTAC*		5	FKH1	-22.1	M phase of mitotic cell cycle	-4.4	”Robertson00 bni1D 50 nM aF, 120 min log10(ratio)”	-6.1	dow
RACAATRG	ROX1	6					Gasch01 pRS ROX1 GAL media	-19.7	down
TCGTTTMA	ERG11 URS1 but not exactly	3			steroid metabolism	-14.8	Hughes00 Itraconazole	-12.3	up
TTCYRGAA		3	HSF1	-18.9	protein folding	-12.5	Gasch00 Heat Shock 015 minutes	-9.1	up
ACCAATCA		2	HAP4	-16.2	ATP synthesis coupled proton transport	-9.3	Ferea99 parent vs evolved 2	-12.3	up
*TTATATA		5	HIR2	-5.3	main pathways of carbohydrate metabolism	-8.8	Gasch00 YPD 10 h 30C	-15.4	up
TCCRCGGA		3	YAP6	-7.1	response to drug	-6.8	Hughes00 aep2	-15.0	up
TGTGGCGT		2	MET4	-10.8	sulfur metabolism	-6.5	Gasch00 Amino acid adenine starvation 0 5 h	-6.9	up
CTCCGCGG	PDR1	4	CBF1	-6.6	response to drug	-6.2	Carrol02 pho85D 10 mM 1NaPP1	-10.7	up
ACACACAC	Meiosis related motifs	5	IME4	-10.5	pyridoxine metabolism	-4.5	Nautiyal02 tlc1 Expt.2 Passage 5	-6.1	up
YGTCACAR	”(Ume6,Sum1, Ime4)”	3	SUM1	-10.2					
MSE ACACAAAA		4	SUM1	-7.4	spore wall assembly (sensu Saccharomyces)	-4.9			
PDR1 UME6 TCRGCGGC		3	CBF1	-7.5	meiosis	-8.4	Hughes00 sgs1	-7.1	up
SWI5 GTGGCTGG		2	SWI5	-8.8					

MIG1 ACC-CCGCA		2			monosaccharide transport	-8.3	Hughes00 sir4	-7.1	up
CCAATCAA		2	HAP4	-8.2	”energy coupled proton transport down the electrochemical gradient”	-5.6	Ferea99 parent vs evolved 2	-7.3	up
GCGAAAAA		4	ASH1	-6.7	regulation of transcription from Pol III promoter	-4.2	Spellman98 CLN3 induction 30 minutes	-7.0	up
CGCCGTAC		2	DAL81	-7.0			Ideker01 gal1gal10+gal	-5.4	up
GGTACGGC		2	DAL81	-5.5	amino acid transport	-6.2	Ideker01 gal1+gal	-5.2	up
TGTGCCTT		2	PHD1	-6.0					
TCTCGAGA	Putative	11	SWI4	-5.2			Ideker01 gal80-gal	-14.7	down
GGTRATGR	Putative (ESR1? RAP1?)	15			ribosome biogenesis	-22.7	Gasch00 1 5 mM diamide 10 min	-23.7	down
CCCTTAAA	Putative (related to msn2/4? (CCCCT))	2					Gasch00 YPD 2 d 30C	-13.3	up
CCCTTGRA	Putative	3					Gasch00 heat shock 17 to 37 20 minutes	-12.5	up
GGGTGCAG	Putative (REB1 and RAP1 with 1 mis)	2			siderochrome transport	-10.8	Hughes00 kre1	-11.1	up

Table 6.1: Octamer families and their functional annotation. For each source of information P indicates the logarithm (base 10) of the p-value obtained. U/D indicates the direction of the co-regulation in gene containing the motifs in the condition that appears under ’Expression’.

6.6 Selection on Reb1 TFBSs

The representation of binding sites as families of motifs is a powerful and informative tool for analyzing their regulatory function. By studying intra- and inter-family substitution ratios in the selection network, one can sometimes reveal high resolution evolutionary structure inside families. Such structure may suggest that the targets of a certain TF are subdivided into two or more groups with different functions. To illustrate this, we examined the region of the selection network corresponding to Reb1 motifs (Figure 6.7).

We selected the relevant Reb1 motifs by taking 80 octamers with highest matching probability to the Transfac Reb1 position weight matrix M00307 [149]. To gain maximal amount of information, we used all selection network arcs, even where confidence intervals were large. This process generates larger clusters than those detected in the global analysis. Most Reb1 motifs form a large family of densely connected variants of a consensus. Interestingly, in addition to the large family, a well-separated smaller Reb1 family is also observed. Although many motifs in one family differ by one nucleotide from some motifs in the other family, there are very low substitution rates (and consequently negative selection ratios) on all these inter-family arcs. We tested the probability of detecting two separated clusters of the observed size in a random graph with the same number of non-negative-rate arcs as in the original one. Indeed we found that such pattern is unlikely to appear at random ($p < 10^{-4}$).

Both Reb1 families show strong association ($p < 10^{-60}$ for each) with the Reb1 ChIP profile [87] but expression of genes with binding sites from the two clusters differs (t-test, $p < 0.01$). The combination of evolutionary and functional genomics evidences lead to the hypothesis that each family represents a distinct mode of Reb1 operation. Reb1 is an auto-regulating TF and it was shown that several auto-regulatory binding sites are present in its promoter [147]. The two strongest of these sites contain the motifs TTACCCG (binding affinity k_d=25nM) and TTACCCT (k_d=70nM), which appear as major hubs in the two families. The direct binding affinity measurements of the two variants provide a possible mechanistic explanation for the functional diversity of the two families: The large family, containing the first auto-regulatory motif, is composed of sites with higher Reb1 affinity levels (lower k_d), and is capable of activation or repression in lower transcription factor concentrations. The smaller family, containing the second auto-regulatory motif, is composed of sites with lower Reb1 affinities, which respond only when Reb1 attains high concentrations. Reb1 may thus operate in two distinct modes, which are stabilized via autoregulation.

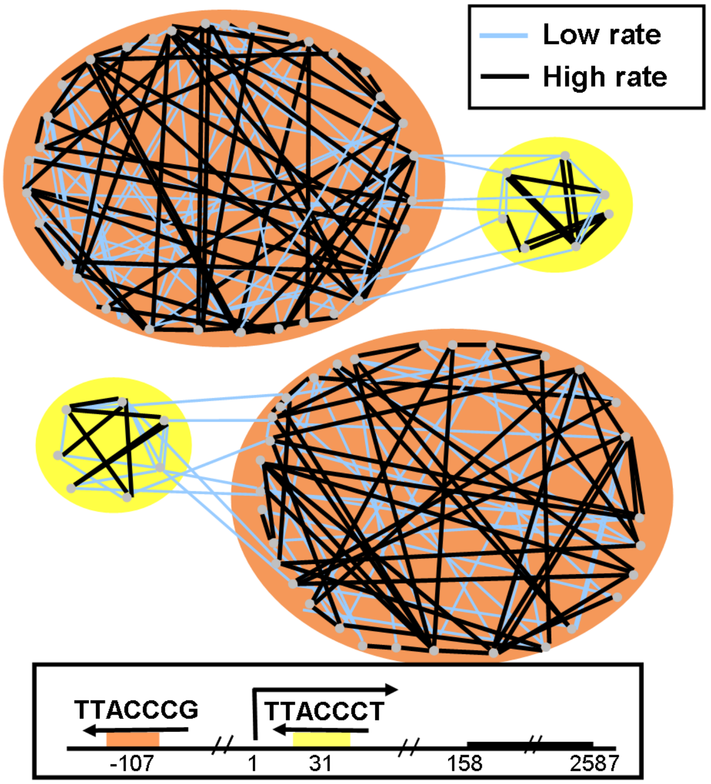


Figure 6.7: **Multi-modality of the Reb1 transcription factor**. A part of the selection network containing k-mers that are associated with Reb1 is shown in greater detail. All nodes represent variants of the Reb1 consensus. Note that in this figure, all selection network arcs, including those with low confidence, are plotted, and reverse complement motifs are not combined. Upper part: Reb1 motifs. Mid part: Reb1 reverse complementing motifs. Black arcs represent high substitution ratio (low selective pressure). Blue arcs represent low substitution ratio (high selective pressure). A clear two-family structure emerges, where low ratio arcs are separating a large Reb1 family from a smaller one. The structure is mirrored in the reverse complement motifs. The Reb1 promoter (lower part) contains two auto-regulatory sites, each located in a different family, with distinct binding site affinities. This raises the hypothesis of two distinct Reb1 modes of operation, each activating a different group of motifs in a specific concentration. Note that nodes in the network are octamers and the Reb1 consensus is a septamer.

6.7 Selection on Leu3 TFBSs

To further study the relation between binding affinity levels and transcription factor functional diversity, we analyzed in detail the selection on Leu3 binding sites. Leu3 binding affinity was measured for 50 different variants of the palindromic consensus decamer CCGGTACCGG [91]. For each of the 50 motifs with known affinity value, we also added to the list its reverse complement and assumed that they have equal affinities. This added 49 distinct motifs in total. We identified 455 weakly conserved loci, in which at least 3 out of the 5 sequenced sensu stricto species (the four mentioned above plus *S. paradoxus*) contained one of the 99 variants with known affinity, or a neighbor of such a variant. Additionally, we searched the promoters of *S. castellii* and *S. kluyveri* and added matching loci when found (about 15% success for each). Since Leu3 motifs are sparse and since we were interested in the selection inside the family, we estimated substitution rates directly, not computing selection ratios. Recall that we are using a phylogenetic tree (S, V, par) and assume that the length (duration) of the branch leading to a node v is given by t_v (t_v's were estimated from the data using a standard one-parameter model (as in [79])).The rate of substitution for a pair m_1, m_2 is computed as $n_{m_1,m2}/(\sum_v t_v n_{m_1}^{par(v)})$. In other words, we divide the number of inferred ancestral $m_1 \rightarrow m_2$ substitutions by the total time in which m_1 was under selection. Note that this approach is much more sensitive to artifacts than the selection ratios we introduced above, since we have to assume the t parameters and since we explicitly model the process as exponential.

Figure 6.8 shows a clear sub-family structure in the Leu3 selection network. Interestingly, the sub-family structure is matching the known Leu3 affinities. We observe one high affinity family (consisting of the palindromic consensus) and two reverse complementing, low affinity families. We estimated the average exponential rate of substitution between high affinity sites as 0.17, and the rate of high to low affinity substitutions (breaking the subfamily boundaries) as 0.01 (see Figure 6.8B). To test the significance of this rate difference we performed a generalized likelihood ratio test. The null hypothesis assumes all substitutions appear with equal rates. The alternative hypothesis allows different rates for the two types of substitutions (high-high and high-low). We estimated a p-value by re-sampling substitutions from an exponential distribution and computing the distribution of likelihood ratios under the null model. This procedure allowed us to reject the null hypothesis with a significant p-value ($p < 0.01$).

The significant rate difference supports the hypothesis that the two Leu3 subfamilies represent distinct functional modes of Leu3 regulation and that evolution conserved not only functionality (ability to bind Leu3) but also the more intricate level of activation. Motifs in both affinity domains are conserved at similar rates (Figure 6.9), indicating that both families are functional. Furthermore, motifs with k_d levels that fall in-between the high and low affinity families appear infrequently in promoters (Figure 6.8c), raising the hypothesis that sites with ambiguous affinity are selected against, and that evolution imposed a discrete bimodal structure on Leu3 sites, by selecting only sites that fall clearly in one of the two families.

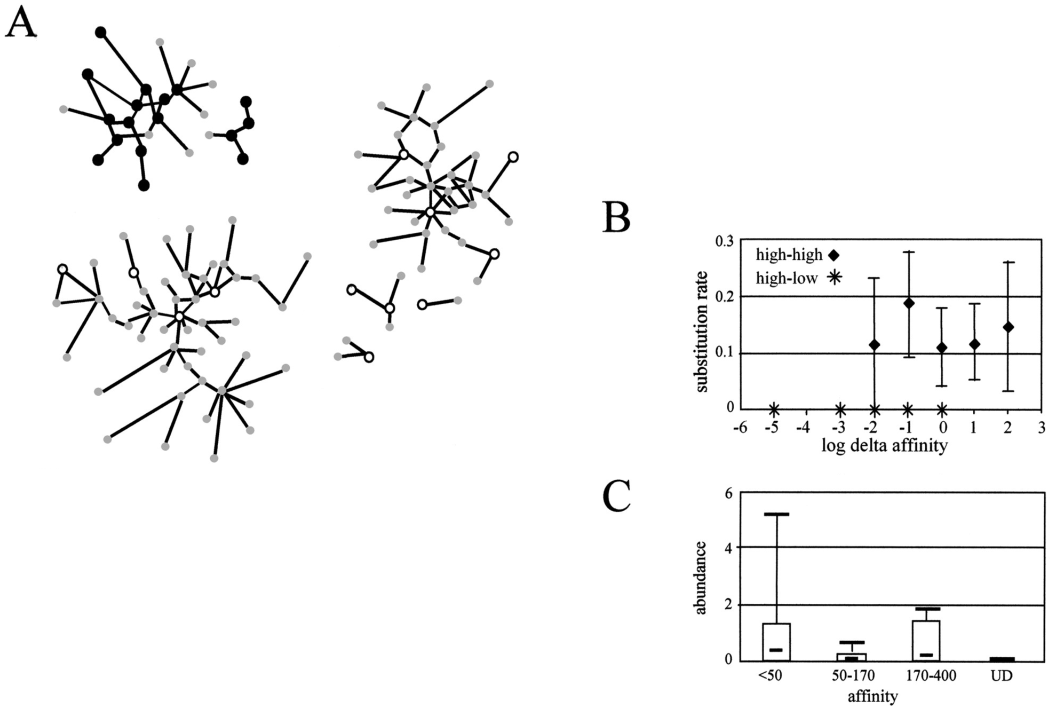


Figure 6.8: **The effect of binding site affinity on Leu3 multi-modality.** (A) Cluster structure in a fragment of the Leu3 selection network. Blue nodes: high affinity ($k_d < 50nM$) motifs; red: low affinity ($k_d > 170nM$); gray: unknown affinity. Arcs connect neighbors with high substitution rate. Arcs with low substitution ratio are not shown. As only motifs with measured affinities and their neighbors are presented, some real families may appear fragmented. Note that no component contains both high and low affinity nodes. (B) Rate of substitution as a function of affinity change within and between the families. Substitutions with similar effect on $\log(k_d)$ are grouped together and their joint rate is plotted. The rates of substitution between high affinity sites and from high to low affinity sites differ significantly ($p < 0.01$). (C) Motif abundance box-plots for different affinity intervals. Both non functional motifs (undetectable k_d-s) and medium affinity motifs ($50nM < k_d < 170nM$) have very low abundances compared to motifs in the high and low affinity intervals, which were also identified as families in the selection network. This may indicate that medium affinity motifs are selected against to avoid ambiguity of site modality and to increase transcriptional program robustness.

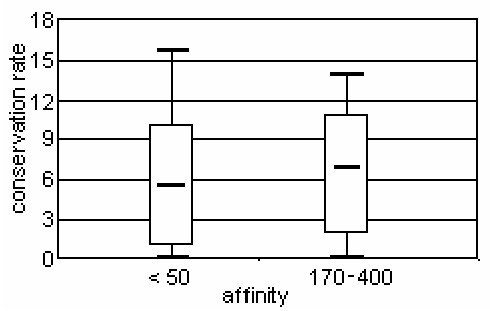


Figure 6.9: **Leu3 conservation rates in high and low affinity sites**. The boxplot represents the distribution of conservation rates for motifs in the affinity intervals $0nM < k_d < 50nM$ and $170nM < k_d \leq 400nM$. In spite of the difference in affinity level, the conservation rates behave similarly, supporting the claim that both constitute functional targets of Leu3.

Chapter 7

Evolution of transcriptional modules

In this chapter we combine some of the techniques developed above, in an attempt to characterize the evolution of cis-regulation in transcriptional modules. Transcriptional modules of co-regulated genes play a key role in regulatory networks. As was shown above, we can use functional genomics data to dissect large biological systems into such modules. Moreover, comparative studies show that modules of co-expressed genes are conserved across taxa [14, 125, 98]. A common tacit assumption is that conserved regulatory mechanisms underlie module conservation, as co-regulation imposes tight constraints on the evolution of a module's promoters. Indeed, recent studies showed that orthologous transcriptional modules are often associated with conserved cis-elements [44].

Here we explore the evolution of cis-regulatory programs associated with conserved modules by integrating expression profiles for two yeast species with sequence data of 15 other fungal genomes. Our goal is to match the conservation of the molecular phenotype (gene expression program) with the underlying genotype (cis-regulatory elements). We show that while the cis-elements accompanying certain conserved modules are strictly conserved, those of other conserved modules are remarkably diverged. In particular, we infer the evolutionary history of the regulatory program governing ribosomal modules. We show how a new cis-element emerged concurrently in dozens of promoters of ribosomal protein genes, followed by the loss of a more ancient cis-element. We suggest that this formation of an intermediate redundant regulatory program allows conserved transcriptional modules to gradually switch from one regulatory mechanism to another while maintaining their functionality. The methodology we develop in this chapter can serve as the basis for deeper explorations into the evolution of regulatory mechanisms.

Results in this chapter were derived in join work with Aviv Regev (Harvard) and were published in [129].

7.1 Phylogenetic cis-profiling

Our methodology for the analysis of cis-regulatory evolution of transcriptional modules consists of the following steps. First, we identify conserved transcriptional modules using expression data from two distant yeast species, *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe* (Figure 7.1a,b). Second, we use sequence information to derive orthologous modules in 15 additional fungal species and identify the cis- regulatory elements associated with each module in each species (Figure 7.11c-e). Third, we reconstruct the evolution of cis-elements associated with each module (Figure 7.1f).

7.1.1 Fungal sequences

We used the previously published genomic sequences and annotations of *Saccharomyces cerevisiae, Saccharomyces paradoxus, Saccharomyces mikatae, Saccharomyces kudriavzevii, Saccharomyces bayanus, Saccharomyces castellii, Saccharomyces kluyveri* [23, 79], *Kluyveromyces waltii* [78], *Ashbya gossypii* [32], *Candida albicans* [74], *Neurospora crassa* [43], *Aspergillus nidulans* (Aspergillus Sequencing Project, Center for Genome Research, http://www.broad.mit.edu/annotation/fungi/aspergillus/), *Candida glabrata, Kluyveromyces lactis, Debaryomyces hansenii, Yarrowia lipolytica* [35] and *S. pombe* [151].

7.1.2 Approximate promoter identification

For each species, we obtained a set of approximate promoter regions by extracting the sequence 600bp upstream of the transcription start site of each annotated gene.

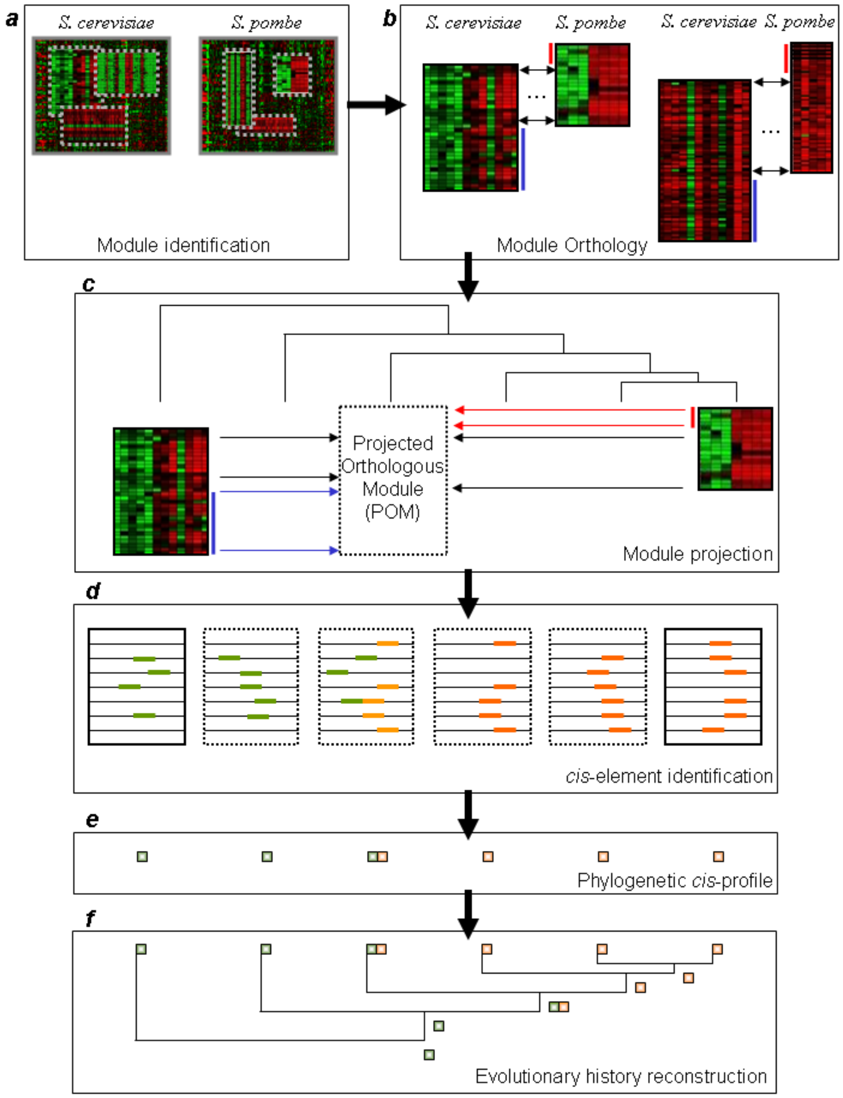


Figure 7.1: **A computational framework for evolutionary cis-profiling of transcriptional modules in fungi.** (a) Expression profiles (genes - rows, arrays - columns) are used to identify transcriptional modules - sets of genes with a shared expression patterns across a set of conditions (dashed grey rectangles). Independent sets of transcriptional modules are generated from *S. cerevisiae* and *S. pombe* gene expression profiles using the SAMBA algorithm. (b) Using orthology relations (see Methods), orthologous module pairs are identified and constitute the conserved transcriptional modules in subsequent analysis. Each pair of modules has a shared subset of matched orthologous genes (black arrows) together with species-specific genes (blue and red bars for *S. cerevisiae* and *S. pombe*, respectively). (c) Projected orthologous modules (POMs) (dashed rectangle) are generated in additional species by taking all genes in that species that are orthologous to genes from the conserved module pair - either just the orthologous core (black arrows) or all genes (black, red and blue arrows). POMs are generated for 15 additional fully sequenced fungal species. (d) Each of the original and projected transcriptional modules (solid and dashed rectangles, respectively) is searched for enriched cis-regulatory elements (green and orange bars) in its genes' promoters (black lines). (e) A phylogenetic cis-profile identifying the enriched elements (green and orange boxes) is generated for each set of orthologous modules. (f) The evolutionary history of the regulatory mechanism is reconstructed for each module from its phylogenetic cis-profile using the known phylogeny of the yeast species [84].

If another gene was annotated within that region, the sequence was pruned accordingly. For species of the Saccharomyces genus, we found that in many cases, the first exon was missed by the existing annotation, resulting in the loss of important regulatory signals in subsequent promoter analysis. To correct such annotation errors, we extracted the set of intronic *S. cerevisiae* genes and their orthologs in each of the Saccharomyces species, as defined by the Saccharomyces Genome Database [22] (SGD, http://www.yeastgenome.org). For each such potential intron-associated gene, we tested for the occurrence of the exact *S. cerevisiae* splicing branch site TACTAAC within 100bp upstream of its annotated start codon. If such a sequence appeared, we changed the annotation of that gene to reflect an intron of the same size as the *S. cerevisiae* intron, thereby moving the predicted promoter upstream. We manually reviewed all annotation changes.

7.1.3 Ortholog identification

To identify orthologous genes between *S. cerevisiae, S. pombe* and all other species we used a standard BLASTp [3] homology search of all open reading frames (ORFs) in one species against all those of another species. Two proteins in a pair of species were identified as orthologous if each was the other's best match according to the BLAST score, a standard approach in sequence analysis. For *S. cerevisiae, A. gossypii* and *K. waltii*, we also added orthologies based on the previously published whole genome duplication analysis [32, 78]. For the Saccharomyces species, we used previously published orthologies [23, 79]. Our subsequent analysis only used orthologies between pairs of species (e.g., for computing projected orthologous transcription modules), so pair-wise relations were sufficient and we did not employ higher order analysis (e.g., grouping orthologous proteins to clusters).

7.1.4 Orthologous transcriptional modules

To discover transcriptional modules we analyzed 1020 previously published expression profiles for *S. cerevisiae* and 87 available profiles for *S. pombe.* The expression data of each species were analyzed separately using the SAMBA algorithm and two sets of transcriptional modules were formed. Each transcriptional module is comprised of a set of genes with a significant shared expression pattern across a set of experiments. We measured the degree of orthology between modules based on the number of orthologous genes shared by them using the following approach. Given an *S. cerevisiae* module and an *S. pombe* module, we calculated the number of genes from the *S. cerevisiae* module that have an ortholog in the *S. pombe* module, and used the hypergeometric distribution to calculate a p-value for finding at least that many shared orthologs between these modules (given the total number of *S. pombe* genes with at least one *S. cerevisiae* ortholog). Two modules with lowest reciprocal orthology p-values were defined as orthologous and were used in subsequent analysis of conserved transcriptional modules. As the set of modules generated in each species may contain overlaps, this final step eliminated possible redundancies.

7.1.5 Phylogenetic cis-profiling

To discover cis-elements enriched in conserved modules in different species we employed the following procedure. Starting with a pair of orthologous transcriptional modules in *S. cerevisiae* and *S. pombe*, we first formed a projected orthologous module (POM) in each of the other species by taking all of the genes in this species that are orthologous to genes from the *S. cerevisiae* module. Several alternative definitions of the POMs, such as the union or intersection of the orthologous gene sets from both the *S. cerevisiae* and *S. pombe* modules, yielded similar results. Since *S. cerevisiae* is evolutionarily closer than *S. pombe* to the additional 15 species we analyzed, we report results based on the simplest procedure, in which we projected all POMs from the *S. cerevisiae* modules. Next, we applied our cis-element finding algorithm (Appendix B) to each of the POMs and construct a set of significant PWMs in each of them. We then used all the discovered PWMs (from all species and all modules) as seeds for the PWM optimization algorithm on all modules in all species. This final step ensures that the absence of a PWM in one species' POM is not an artifact of the motif finding procedure.

7.2 Conserved modules and their cis-elements

Using the methods outlined above, we have detected a set of conserved transcriptional modules and identified cis-elements that are enriched in their promoters (Figure 7.2). In several cases we found similar cis-elements enriched in the orthologous modules of both species, even when the orthologous genes constituted only a small fraction of the modules' genes. The conserved elements were often also known to correspond to binding sites of orthologous transcriptional complexes (Figure 7.2a-b). Surprisingly, we also found several cases where the "phenotypic" conservation of gene expression is not accompanied by a corresponding conservation of the enriched cis-elements. These cases include modules for key molecular functions, such as ribosomal protein synthesis (Figure 7.2d) and stress response (Fig. 1e), all of which were demonstrated to be conserved across a wide range of taxa [14, 125]. In other cases, such as the ribosome biogenesis module (Figure 7.2f) or the S phase module (Figure 7.2a), an *S. cerevisiae*-specific motif is found along with a second, conserved motif.

We wished to ensure that the differences in enriched cis-elements between species are not an artifact of the way in which we identified orthologous transcription modules or of our motif discovery approach. To verify that these differences are identifiable even when looking for cis- elements only in the promoters of conserved genes shared by the two modules, we repeated the cis-regulatory analysis using perfectly orthologous transcription modules in which each gene in one module is matched by at least one ortholog in the other module. To generate perfect orthologous modules from a pair of mutual (non-perfect) orthologous ones, we enhanced the existing SAMBA algorithm. The modified algorithm is initialized with a pair of modules and starts by removing all non-orthologous genes in the two modules. The algorithm then iteratively adds and removes pairs of orthologous genes to improve the total score of the module pair. In cases where a gene has more than one ortholog, the algorithm can add either a single gene pair or a larger orthologous group of genes, depending on which alternative scores higher. The algorithm outputs a pair of transcription modules, such that each gene in one module has at least one ortholog in the other module and such that additional gene pairs cannot be added or removed without decreasing the total score of the module pair. Importantly, the results of enriched cis-elements obtained on such perfect orthologous module pairs are consistent with those we reported for the (non-perfect) orthologous modules. They thus confirm that our findings on the evolutionary dynamic of cis-regulation were not biased by the imperfect orthology. To ensure that our motif discovery procedure does not lead to false negatives we initiated the motif search for a given species (in the second iteration) using the motifs discovered for the orthologous modules in all the other species.

We conclude that the divergence in cis-elements in conserved transcriptional

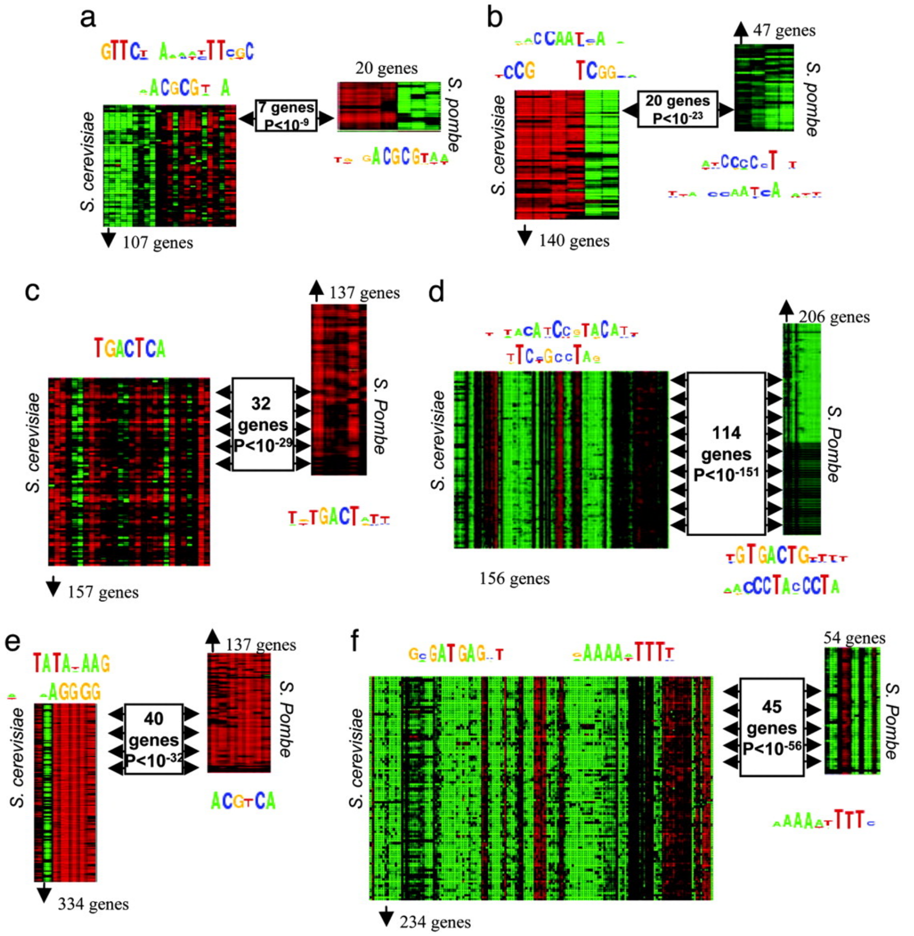


Figure 7.2: **Conserved transcriptional modules in** *S. pombe* **and** *S. cerevisiae* **and their associated cis-elements.** Shown are the *S. cerevisiae* and *S. pombe* modules for the six key conserved modules we identified, together with the cis-elements enriched in the promoters of these modules' genes. For each module, the profile shows the module genes (rows) induced (red) and repressed (green) across different experiments (columns). Rectangles indicate the orthologous genes, their number, and the p-value of their co-occurrence. The enriched cis- elements associated with each module are shown in sequence logo above or below it. (a) S-phase module, associated with the conserved MCB element (ACGCGT, bound by orthologous MBF complexes in both species), as well as an *S. cerevisiae*-specific element. (b) Respiration module, associated with the conserved HAP2345 site (CCAATCA, bound by the orthologous Hap2345 and Php2-5 complexes). (c) Amino acid metabolism module, associated with the conserved GCN4 site (TGACTCA). (d) Ribosomal proteins module, associated with RAP1 (TACATCCGTACAT) and IFHL sites (TCCGCCTAG) in *S. cerevisiae*, and with a Homol-D box (TGTGACTG) and a Homol-E site (ACCCTACCCTA) in *S. pombe*. (e) Stress module, associated with the STRE site (AGGGG) in *S. cerevisiae* and with the CRE site (ACGTCA) in *S. pombe*. (f) Ribosome biogenesis module, associated with the conserved element RRPE (AAAAATTTT) and the *S. cerevisiae*-specific PAC element (GCGATGAG).

modules does not stem from the non-orthologous members of these modules It was difficult to envision how such divergence in the mechanisms regulating the expression of highly essential and tightly coordinated modules could take place without deleterious effects.

7.3 Phylogenetic cis-profiles in 17 yeast species

To try and shed light on the apparent divergence of cis-regulatory elements in conserved modules, we analyzed the cis-elements enriched in conserved modules in 15 additional fully sequenced fungal species covering the evolutionary spectrum between *S. cerevisiae* and *S. pombe*. Since genome-wide expression data for these species are currently scarce, we inferred projected orthologous modules (POMs) in these species by taking all genes that have an ortholog in the *S. cerevisiae* conserved modules. We then searched for enriched cis-elements in the promoters of the projected module's genes and analyzed each motif identified in one species for its presence in all other species. As before, we verified that the motifs we detected were also enriched in modules that were generated by projecting only from the orthologous cores of the *S. cerevisiae* and *S. pombe* modules. This ensured that unique motifs are not simply contributed by the non-orthologous genes. The resulting phylogenetic cis-profile associates each module with a set of cis-elements in each species. By examining similarities across the profiles, we can identify conserved mechanisms. Indeed, for modules whose regulatory mechanisms were conserved between *S. cerevisiae* and *S. pombe*, the phylogenetic cis-profiles reveal perfect conservation in all other species (Figure 7.3), consistent with recently published results [44]. Importantly, by also considering differences between profiles and invoking classical evolutionary principles (e.g., maximum parsimony), we can reconstruct the evolutionary scenario that explains the divergence of the regulatory mechanisms associated with each conserved transcriptional module.

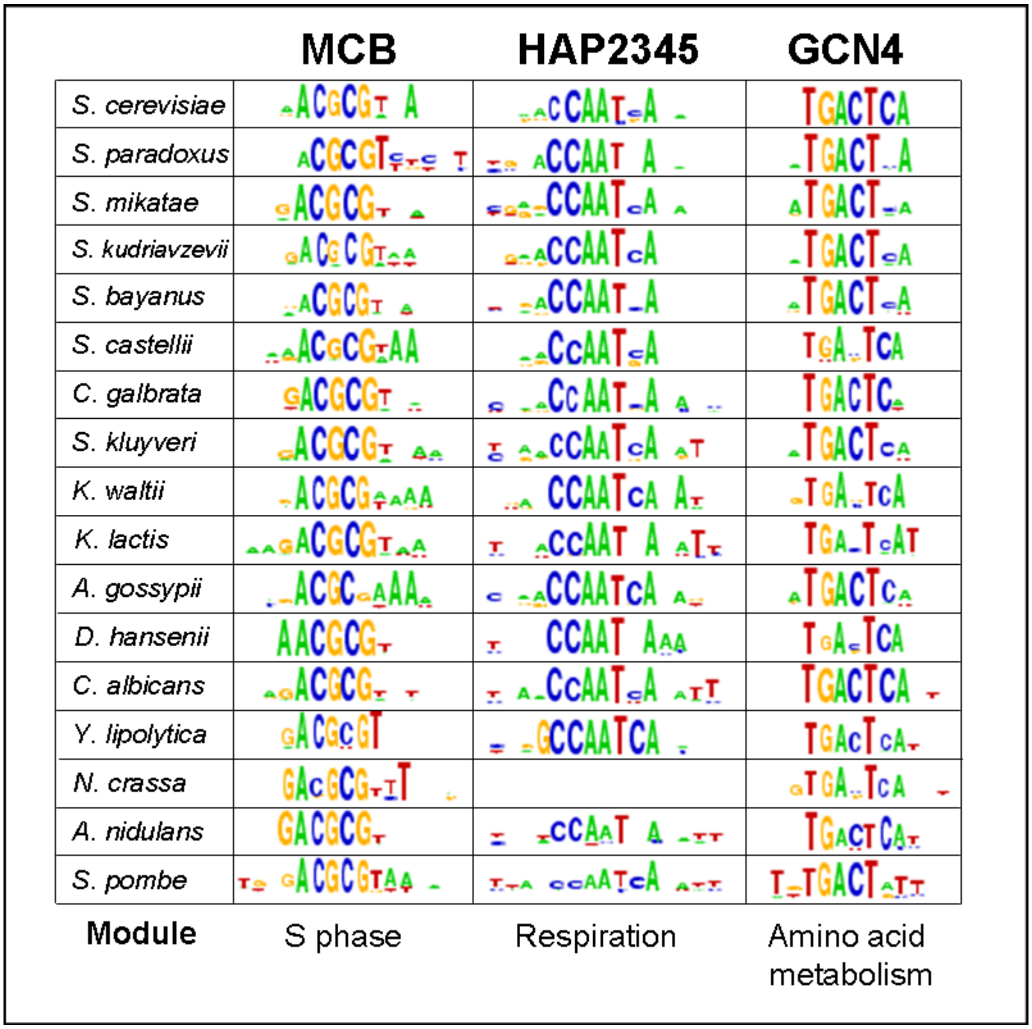


Figure 7.3: **Conserved cis-elements in orthologous modules from 17 yeast species.** For each of the 17 yeast species we analyzed, we show the sequence logo of the cis- elements discovered in the S phase, respiration and amino acid metabolism modules. For all species, we discover the MCB box in the S-phase module, the Hap2345 binding site in the respiration module and the GCN4 site in the amino acid metabolism module. The only exception is the absence of the HAP2345 element in *N. crassa*, which may be due to the relatively few genes in the orthologous module. We note that except for the four sensu stricto Saccharomyces species, all promoters were completely divergent and could not be aligned, consistent with previous reports [23].

7.4 The evolution of the ribosomal regulatory program

A remarkable example of regulatory divergence is the large, tightly regulated and highly conserved ribosomal proteins (RPs) module. Two elements are associated with the *S. cerevisiae* module: the well-known RAP1 binding site, and an IFHL site TC(C/T)GCCTA [97, 115, 145]. Two different elements are found in the *S. pombe* module: the Homol-D box (TGTGACTG) and the homol-E box (CCCTACCCTA), both of which have been shown to regulate the expression of RPs in this species [150]. Such disparity can result either from divergence in the DNA binding sequence of the same ancestral transcription factors or from the emergence of novel sites bound by distinct transcription factors. If the latter option is true, it also raises the questions of how a regulatory program can shift from one mechanism to another without affecting the module's function. The detailed phylogeny of cis-elements in RP promoters (Figure 7.4) allows us to infer the evolutionary scenario underlying this divergence and to address this question of evolvability.

As shown in Figure 7.4, the profiles of *S. castellii*, *C. glabrata*, *S. kluyveri*, K. waltii, and *A. gossypii* contain both the Homol-D box and the RAP1 site (in addition to a strong IFHL site, discussed below). This apparent redundancy of binding sites in these species has two important implications. First, the presence of intermediate species, in which both the Homol-D and the RAP1 sites appear in RP promoters, suggests that the regulatory mechanism associated with the RP module has "switched" from a Homol-D-based mechanism (in the Ascomycota ancestor and *S. pombe*) to a RAP1-based one (in *S. cerevisiae*) and was not modified due to a mere drift in cis-element sequences. Second, it points to a potential process by which such a dramatic change in the regulatory mechanism of an essential module could take place without destroying the coordinated regulation. According to the most parsimonious scenario supported by the data (Figure 7.5), an ancient Homol-D box played the key role in regulating RP transcription. Subsequently, before the divergence of *A. gossypii*, RAP1 emerged as an additional regulator of this module, while the module maintained the functionality of the Homol-D box. As the abundance of RAP1 sites increased, Homol-D lost its central role and was eventually eliminated, possibly following the divergence or loss of the corresponding unknown transcription factor. Thus, a process of infiltration (of RAP1) and loss (of Homol-D) swept through the promoters of the RPs.

To support this evolutionary scenario, we compared two possible scenarios for the ancestral regulation of RP genes by Homol-D and RAP1. In the most parsimonious scenario (Figure 7.5A), Homol-D was present at the ancestral Ascomycotes species, but RAP1 was not. This scenario requires six events (1) Loss of the Homol-D site in the *A. nidulans* - *N. crassa* lineage (2) Loss of the Rap1p TA domain in the *C. albicans* lineage (3) gain of RAP1 prior to *A. gossypii* speciation (4) loss of Homol-D in the *K. lactis* lineage (5) loss of Homol-D in the Saccharomyces lineage (6) Loss of the Homol-D site in the metazoan lineage (or alternatively gain of the site in the fungal kingdom). In the alternative scenario (Figure 7.5B) both elements were present at the ancestral species. This scenario requires (in the most parsimonious explanation) ten events (1) Loss of the Rap1p TA domain in *S. pombe* (2) Loss of the Rap1p TA domain in the *A. nidulans* - *N. crassa* lineage (3) Loss of the Homol-D site in the *A. nidulans N. crassa* lineage (4) Loss of the Rap1p TA domain in the *C. albicans* lineage (5) Loss of the Homol-D site in the *C. albicans* lineage (6) loss of Homol-D in the *K. lactis* lineage (7) Loss of the Homol-D site in the *S. cerevisiae* - *S. bayanus* lineage (8) Loss of the Rap1p TA domain in the *Y. lipolytica* lineage (9) Loss of the Rap1p TA domain in the metazoan lineage (or alternatively gain of the domain in the fungal kingdom) (10) Loss of the Homol-D site in the metazoan lineage (or alternatively gain of the site in the fungal kingdom).

7.5 The basis for regulator switching in the ribosomal regulatory program

Several additional lines of evidence support the ”TF switching” evolutionary scenario. First, we consider the Rap1p transcription factor. Rap1p’s binding specificity is associated with its ancient and conserved function in the regulation of telomere length [88]. The RAP1 binding site in RP promoters is a sub-motif of the telomeric repeat sequence bound by Rap1p in these species [25], including those that do not have a RAP1 site in their RP promoters. Thus, the sequence of the RAP1 motif that emerged in RP promoters matched Rap1p’s pre- existing DNA binding site. More importantly, analysis of Rap1p’s coding sequence in all 17 fungal species and in mammals suggests that the invasion of RAP1 sites into RP promoters is associated with the acquisition of a new trans-activation (TA) domain by Rap1p

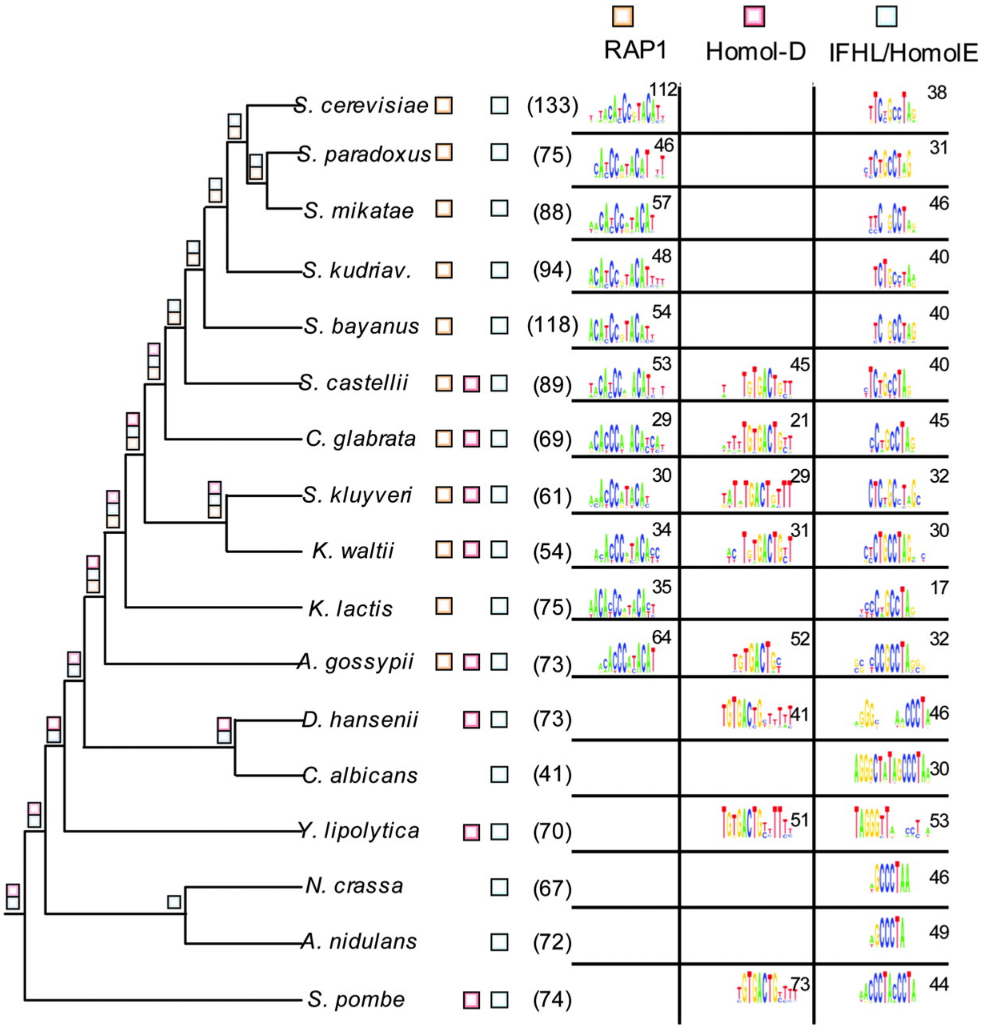


Figure 7.4: **Evolution of the regulatory mechanisms in the highly conserved module of ribosomal proteins.** Phylogenetic cis-profile of the RP module. A schematic phylogenetic tree (branches are not drawn to scale) representing the known phylogeny [84] of the 17 analyzed species is shown, together with the sequence logos of the main cis-elements enriched in each module's promoters, grouped into three distinct types (colored boxes): RAP1 (green), IFHL (blue) and Homol-D (magenta). The total number of genes in each POM is shown in parentheses, and the number of genes that contain each motif is indicated as well. Although the RP module is phenotypically extremely conserved, the phylogenetic cis-profile reveals a gradual switch from a Homol-D dominated mechanism to a RAP1-controlled one, beginning before the speciation of *A. gossypii*. Concomitantly, the IFHL site underwent gradual sequence divergence and possible dimerization or domain duplication of the corresponding transcription factor.

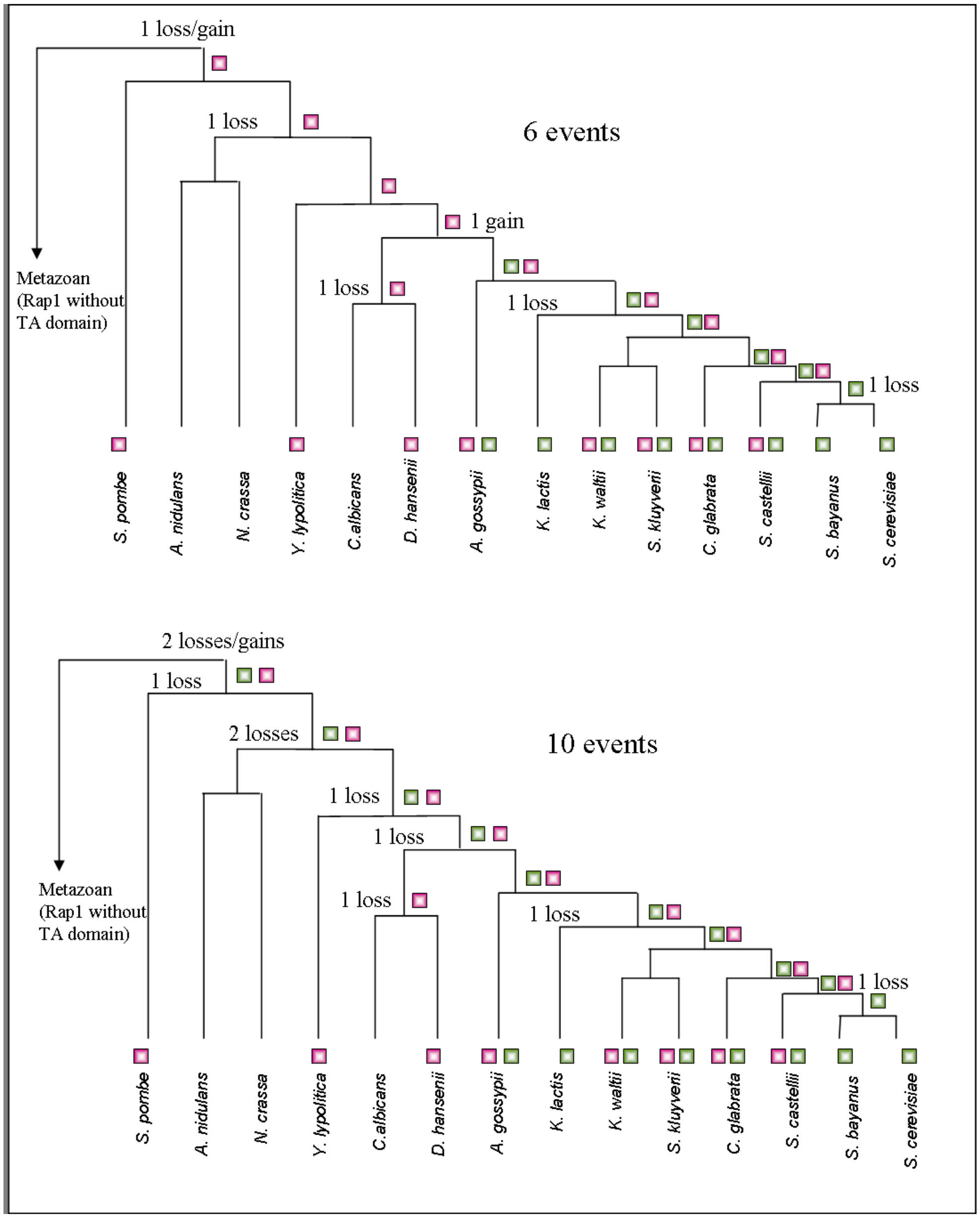


Figure 7.5: **Evolutionary scenarios for RP genes regulation by Homol-D and RAP1 cis-elements.**

after the *C. albicans* speciation and prior to the *A. gossypii* speciation event (Figure 7.6). Thus, while the DNA binding domains of Rap1p (Myb- domains) have been conserved in all species, the TA domain, which is responsible for Rap1p's role as an RP transcription factor [58], follows exactly the same evolutionary pattern as the RAP1 binding site in RP promoters, and is present only in the clade spanning from *A. gossypii* to *S. cerevisiae*. Moreover, Rap1p from species lacking the TA domain, such as *C. albicans* and *S. pombe*, cannot functionally complement for the *S. cerevisiae* Rap1p [76, 15, 141], while those with the TA domain (e.g., *S. castellii*) are adequate substitutes [146]. Thus, the newly acquired domain allowed Rap1p to assume a new role in transcriptional regulation, whereas its conserved DNA binding domain determined the sequence of its corresponding cis-element.

Analysis of the RP promoters loci that contain a Homol-D site in *A. gossypii* and *K. waltii*, two of the species in our collection to also exhibit a RAP1 site, suggests a possible mechanistic model for the process of switching of the transcription factor binding site. In these species, when both RAP1 and Homol-D sites appear in the same promoter, they are usually separated by no more than 2-6 base-pairs, in the conserved order 5'-HomolD-RAP1-3' (Figure 7.7,Figure 7.8), with Homol-D in a fixed orientation relative to the transcription start site (in contrast to *S. pombe*, where it has no strand preference). This strong association may indicate a corresponding cooperative association between the Homol-D binding protein and Rap1p, which may have facilitated Rap1p's infiltration into the RP regulatory program. Taken together, our results for both the transcription factors and their binding sites propose a coherent view of a process by which RP regulation gradually switched from one transcription factor to another without losing its essential functionality.

7.6 Gradual evolution in the IFHL box

Additional examination of the phylogenetic cis-profiles (Figure 7.4) suggests that the second cis-element in the *S. cerevisiae* RP module, the IFHL site (TCTGCCTA), has evolved primarily by a different mechanism, involving gradual divergence in DNA binding sequence. First, this element is clearly enriched in the entire Saccharomyces genus as well as *S. kluyveri*, *K. lactis*, *A. gossypii* and *K. waltii*. Furthermore, close inspection of motifs enriched in the remaining species, *C. albicans*, *D. hansenii*, *Y. lipolytica*, *N. crassa*, *A. nidulans* and *S. pombe*, suggests that they also carry related

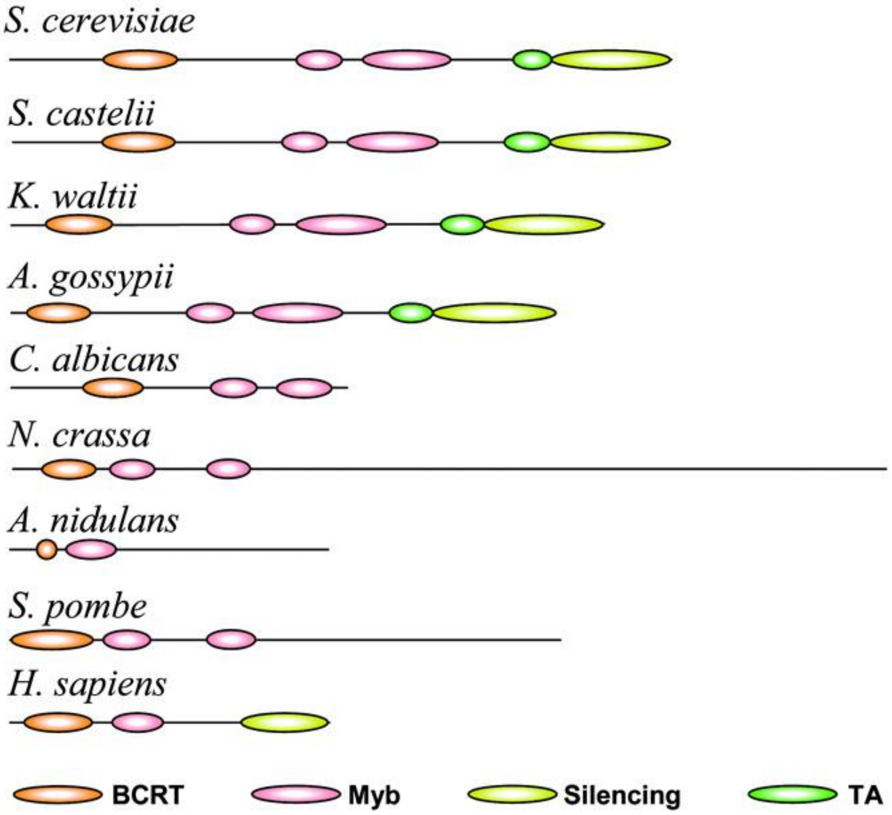


Figure 7.6: **Rap1p sequence evolution.** A scaled schematic representation of Rap1p sequences is shown for 8 of the species in panel A, along with the human protein. Colored ovals indicate the presence and position of BCRT (orange), DNA binding (Myb, pink), silencing (olive), and trans-activation (TA, dark green) domains. The DNA binding Myb-domain is present in all species, but the trans-activation domain is apparent only in those species that harbor the RAP1 motif in their RP module genes (*S. cerevisiae*, *S. castellii*, *K. waltii*, *A. gossypii* and all the intermediate species). The TA domain is absent from all species lacking the RAP1 element in RP promoters, including *C. albicans*, *N. crassa*, *A. nidulans*, and *S. pombe*. A Rap1p ortholog cannot be identified in *Y. lipolytica* and no significant homology was found to the TA domain for the *D. hansenii* Rap1p (not shown).

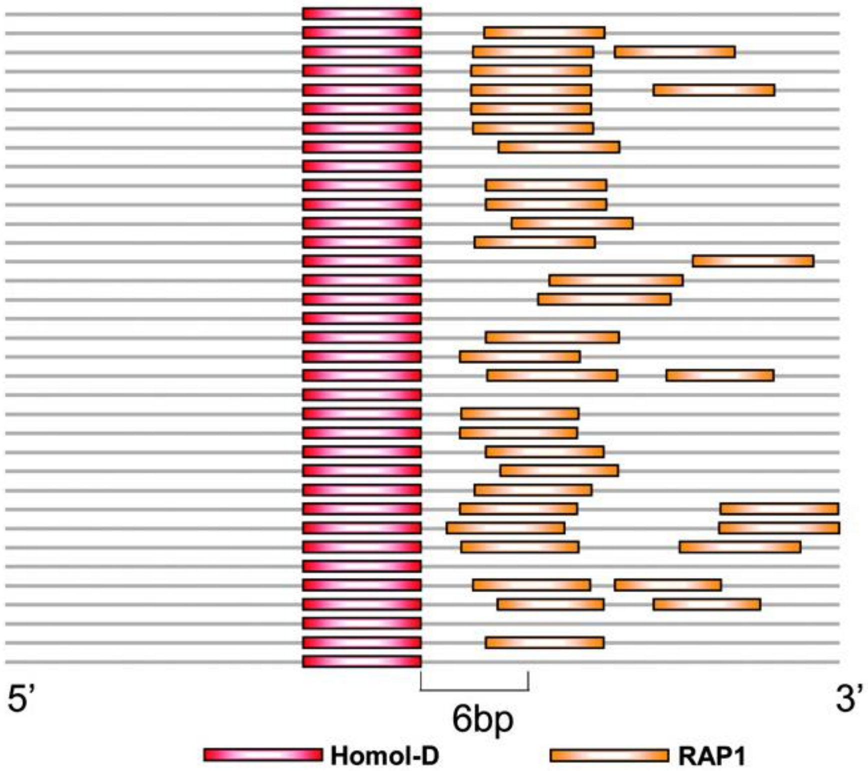


Figure 7.7: **The Homol-D-RAP1 cis-regulatory module.** Shown is a scaled schematic representation of the 35 promoters of the *A. gossypii* RP genes with the highest scoring Homol-D elements. Colored bars indicate the Homol-D (magenta) and RAP1 (green) sites. The two sites are extremely close, with the RAP1 trailing the Homol-D site by 2-6 bp, indicating a possible interaction between their corresponding transcription factors.

```
>282  ACTCTTGGCACTTGCTTATCACGTGTATATCATGTGACTGTATATCATGTGACTGTATATCATTTGAATTTCTT
>809  GCCATTGGAGTCCAGCAAGCGGGGAATGCTGTTGTGACTGTAACACCCATACATTGCAGCCCGTACATTTCAAC
>2153 CAGCCGCTTATATAAACACCTAGTTGTTTGTATGTGACTGTGCACCCATACATTTCAGCACCCATACATATAAC
>3378 CCCTGTTCCGGCGCTCCAGCCGGGCTCTGGACTGTGACTGTGCACCCACACATATGATATTTTATTTTTTTGTC
>4170 AACCTAAGGTCCAAATATCTTAGATATCCCCATGTGACTGTGAACCCATACATTTACATTTGCACCCATACATT
>4280 AGAATAGATTATGTAATTTGTAACCGTTATTTTGTGACTGTTCGCCCGTACATCTATATATATATTTTTGTTAT
>4588 AATCCAGGGTGTGTAGAGATTTAGCAGTCGTTTGTGACTGTAAACCCGTACATACTGTTACCATACTTGCCCGC
>5087 ATAGCCGCCCGCGGCTGCTTCAATATGTACTATGTGACTGTGTGCACCCATACATTCACACTGTTTGGATGGAT
>5904 CTAGTCGGTACCTTGGCTCAGCCATCATTCTGTGTGACTGTGGAACGCGGGGCGGTGCGATGTACGGGCGAGAG
>446  CCCAGACACCCTCAACAGGCGGCGGCAGACGGTGTGACTGCGGGACCCGTACACCACTTGGGACCCACGCACAG
>448  TAGCCTAACCGCCCGTGTCTGCGAGATCGTGTTGTGACTGCAGCACCCATACATCAGAAGTATGTTTTTTCTTG
>1995 CTGCGTGGAGTCGCTGCGCGGGATGCCTTCGTTGTGACTGCTGTGCTGCATGGGCTTCTGTGCGAGACCCATAC
>2256 AAATCGTCGCTTACGAAAGAGGGTACCCCTTCTGTGACTGCACATCCGTACATCCATGGACTCCGCCAAATCTG
>2861 --AGTATATAGCTGTGTACGCTGACTCAGTAATGTGACTGCTTCTCCAATGTACGGATGTTCTCGTATGTCTGG
>4963 CATTAGATAATGATTGGCGCAATTATGGGGTGTGTGACTGCGTGTTGTGGATGTGTGGGTGGTTGTGTGCGCCC
>5327 AATTCCGATGCTGAAAATAATAAACTATGAGATGTGACTGCTTGTTGGTATGTATGGGTGCTATTAGAGGTGCG
>5334 GGAATGCCAGATCCGCTTATCGGAGTTCACGGTGTGACTGCTTGTACGTATGTA--------------------
>5472 AAGCATCGCCGCCGCGGACTGCTATGGGAGCATGTGACTGCTGCGCCCATACATCCAAACAACGGAGTTGAAGC
>6103 GGGTTGCGGCACGACACAACGATCACGTGACGTGTGACTGCCACCCGTACATCCGAGACAGGGCGCGCGGCGGG
>931  TTCACCTGTAGGTCACGTGATTGTGAATTGTGTTTGACTGTTGCATCCGTACATTCGGAGGTGTGTGGGTTGTA
>6193 AATGCAGCTAAGTGCCTTATTGATTACGTGAGTTTGACTGTGTACATCCTAACATTTCCATCCGTAATTTTTTT
>3994 -----------TCACGTGTACATAACTTGGGATTTGACTGCCACCCATACATCTTGACTGCGTTTTTTGTTTTT
>229  GTAGACATAAAGTCCGGGTAAGGGCAATGGCGCGTGACTGTCACCCGTACATCTGCGCCCAGCACCCGGCGTGG
>2542 CTCCGTCGGGCCAACAGAAATCTCCGTCGGGCCGTGACTGTTCCATCCGTACATCCGCACCCACGCACCCATAT
>2941 ------------------------------ACCGTGACTGTTTGCATCCGTACATTTGCATCCGTACATTATTT
>208  CAATGACCAACTGTAGTTTCTTTTTTAGATACTGTGACTACACACCCACACACAGCACAAACCCAAACACCAGA
>223  ACGTTAATACCTGATTACATCAGCACACCGTCTGTGACTACCACCCATACATACGATACAGTAAACCCATACAT
>2727 CGTTAGCCGCCGAGGAGTGCCTGTGACAAACATGTGACTACACCCATACATCTATTTATTCAACACCCACACAT
>2790 TATTGTCACGTGAGAGGCAACATAATTGATCTTGTGACTACCACCCATACATTTTGCTACCACCCATACATACT
>5747 AAATAAGTAATAGTTTCTCCATGTCTGGTGCTTGTGACTACCTGTGAGTTCCATCCAGTATGACTACTTGATGA
>1056 -------------------TCACGTGAGTCTGCGTGACTGCACACCCACACCCACACCCATACATAACGACAGC
>438  ACCGAATGACCAGCGGTCCACATGCAAAATTCGGTGACTGTCAACACCCACACATTTCCACCCACACATTAATT
>6225 CTATGCCAGCGCACACGAGCACAGGGTTGCCTGGTGACTGCTCGGGACAATCGTGAGAAGTACGCCTCATCGGC
>3744 TGCCACTTCTCCGGCCGGGTAACCGTTAACCGTCTGACTGTTCCATCCGTACATCCGGTTTTTGCATCCGTACA
>4203 CGCTGTCTCTGAGTCGCCGCGTTTCTAGACGTTCTGACTGTCTGATCTGTCTACGAATATTAGTTTCTGTCACT
```

Figure 7.8: **The Homol-D-RAP1 cis-regulatory module in** ***A. gossypii*****.** Shown are the sequences flanking the highest scoring Homol-D sites in the promoters of RP genes in *A. gossypii*. Homol-D sites (magenta) and RAP1 sites (green) are highlighted. The two sites appear in very close proximity - within a 2-6 bp distance - indicating possible interaction between the corresponding transcription factors binding them.

variants of the same motif (albeit not identical ones). In *C. albicans*, the strongest cis-element in the RP module (AGGGCTATAGCCCT) is a palindrome containing two copies (TAGCCCT and its reverse complemented AGGGCTA) of a variant of the second part of the IFHL motif (GCCTA). A similar complex cis-element is present in *D. hansenii* and *Y. lipolytica*. The promoters of RP genes in the evolutionary distant *N. crassa* and *A. nidulans* contain an exact match to the second half of the *C. albicans* motif (GCCCTA) and the *S. pombe* Homol-E motif (CCCTACCCTA) is a duplicated variant of the same motif (CCCTA). Thus, an ancestral IFHL DNA binding protein may have been associated with the RP module throughout the evolutionary history of the Ascomycota clade. In addition to acquiring smaller scale mutations causing changes in its DNA recognition site, the IFHL binding protein may have either undergone convergent domain duplication in *C. albicans* and *S. pombe* or acquired a dimerization domain in these species. Note that these dimerization or domain duplication events have presumably occurred by different routes in the two species, accounting for the differences in the organization of the respective elements (direct repeats vs. palindromic ones). Additional species-specific motifs are also associated with the RP module, consistent with the evolutionary flexibility of the RP regulatory mechanisms. For example, RP module genes in *C. albicans*, *D. hansenii*, and *Y. lipolytica* are also enriched for the RRPE motif, which is usually involved in other stress related modules. Traces of this enrichment can also be found in other species, most notably *K. waltii*, *S. bayanus* and N. crassa.

7.7 Conservation of the spatial configuration in ribosomal promoters

To examine the interplay between the three main regulatory elements in different species, we analyzed their co-occurrence in the RP genes in each of the species and their relative spatial organization (Figure 7.9, Figure 7.10). In *A. gossypii* and K. waltii many promoters have ”redundant” regulatory mechanisms with three different cis-acting sites, whereas *S. cerevisiae* promoters are simpler and often contain only a (possibly duplicated) RAP1-binding element. Spatial analysis also reveals that certain features of global promoter organization are conserved across species. For example, we found that IFHL sites are typically found 100-200 bp 5’ to the Rap1 site, consistent with the functional constraint imposed by the interaction between

Ifh1p, Fhl1p and Rap1p in the combinatorial regulation of RP genes [97, 115, 145]. Finally, we asked whether some of the differences in the organization of the regulatory mechanisms may also match "phenotypic" differences in gene expression. The evidence from *S. cerevisiae* and *S. pombe* indicates that switching from a Homol-D to a RAP1 cis-regulatory mechanism does not entail such a change, as RP genes are strictly co-regulated in both species, and respond similarly to environmental stress. However, some of the organisms, for example *C. albicans*, employ a regulatory mechanism lacking both RAP1 and Homol-D elements (and using IFHL and RRPE elements). Indeed, a recent expression profiling study [39] indicates that the *C. albicans* RP module responds much more weakly to environmental stress than either *S. cerevisiae* or *S. pombe*.

7.8 Regulatory divergence in the ribosome biogenesis module

The phylogenetic cis-profile of the ribosome biogenesis (RB) module (Figure 7.11) further implies the rapid evolution of cis-regulatory mechanisms. We detected two elements that were previously associated with the transcription of ribosome biogenesis genes in *S. cerevisiae* - the Ribosomal RNA Processing Element (RRPE) [67] and the Polymerase A and C (PAC) element [30]. RRPE was detectable in each of the 17 species, whereas we found PAC as a possible innovation of the *C. albicans* - *S. cerevisiae* lineage (a GATA like box in *N. crassa* may suggest the origin of this innovation). The phylogenetic profile of a third element, TTTCTTTTT, indicates emergence prior to *A. gossypii* speciation and loss after the *S. kluyveri* - *K. waltii* speciation. Co-occurrence and spatial analysis indicate that, as in the emergence of the RAP1 site in the RP module, the transient TTTCTTTTT site is spatially clustered with the additional binding sites PAC and RRPE (Figure 7.12, Figure 7.13, Figure 7.14), possibly facilitating its emergence as a regulator of the module.

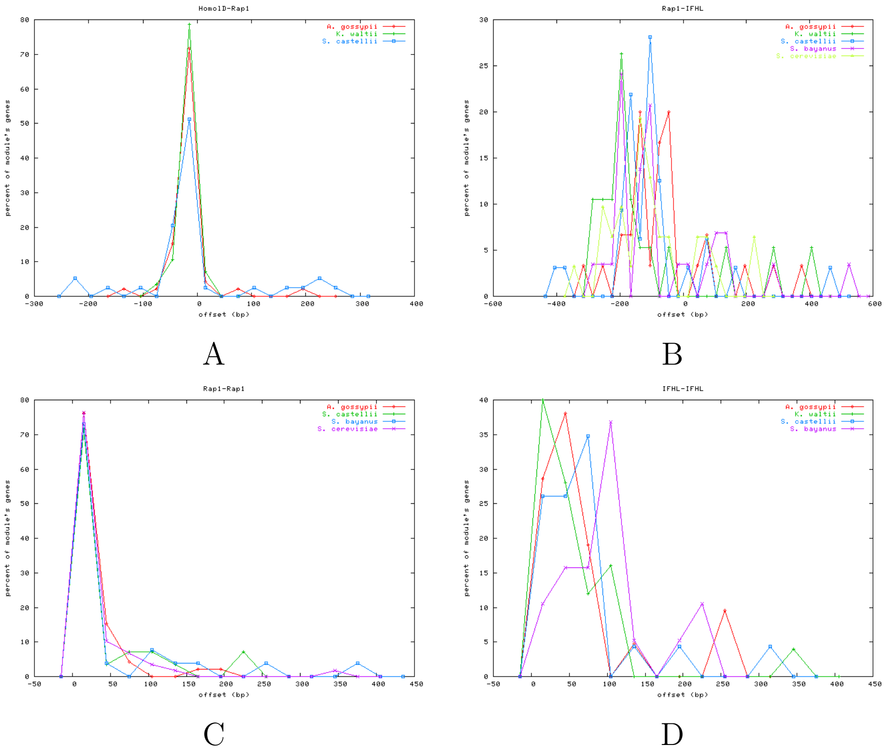


Figure 7.9: **Distributions of relative distances between pairs of cis-elements in promoters of RP genes.** Shown are distributions of the distances (in bp) between pairs of cis- elements in different fungal species. In case of several hits of the same element in the same promoter, the strongest match to the site was used. The coupling of sites with typical (often tight) distances in multiple species corroborates their functional interaction. (a) Distribution of distances between Rap1 and Homol-D sites. The two sites are tightly coupled when they co- occur in the same promoters. (b) Distribution of distances between Rap1 and IFHL sites. Both *A. gossypii* and *S. castellii* exhibit a similar and specific distance. A slightly increased distance is observed between the sites in *K. waltii*, probably as a consequence of IFHL infiltration in this species. The distances between the motifs in the sensu stricto Saccharomyces species, *S. cerevisiae* and *S. bayanus*, are more variable and the distribution is both less peaked and bimodal. (c) Distribution of distances between pairs of Rap1 sites. A very tight positional coupling of multiple RAP1 cis-elements is exhibited in all species, possibly serving to increase the binding affinity for the corresponding transcription factor and to enhance the evolutionary robustness of the transcription program. (d) Distribution of distances between pairs of IFHL sites. Positional coupling of multiple IFHL cis-elements is exhibited, in particular in *A. gossypii* and *K. waltii*.

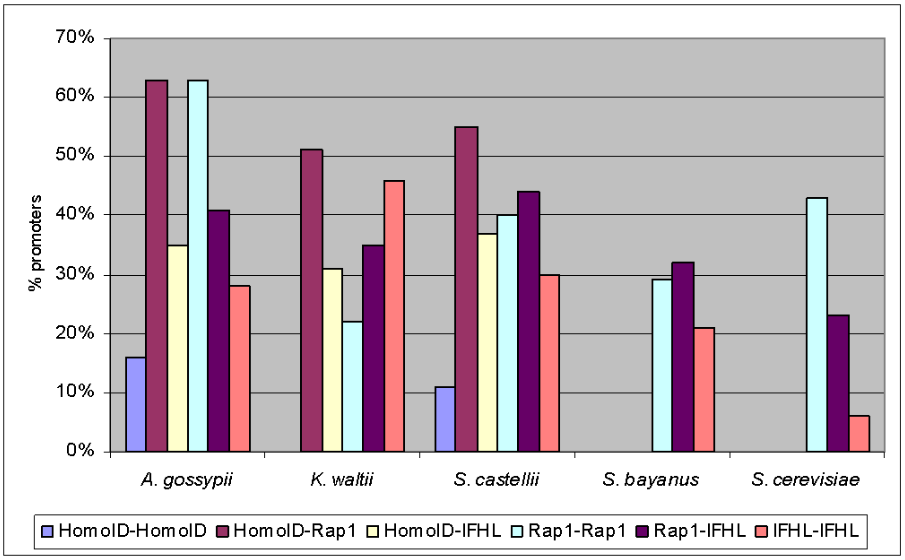


Figure 7.10: **Co-occurrence distribution of cis-regulatory elements in promoters of RP genes in specific species.** The percentage of promoters bearing each type of pair of cis- elements is shown for each of the species. We observe a transition from highly redundant promoters, with multiple types of cis-elements (e.g. in *A. gossypii* and *S. castellii*) to a RAP1- dominated regulatory mechanism (e.g., in *S. bayanus* and *S. cerevisiae*). Consistent with the spatial distributions, promoters with two IFHL cis-elements are more abundant in *K. waltii* compared to the other species, and Rap1-IFHL pairs are less abundant in the Saccharomyces sensu stricto species

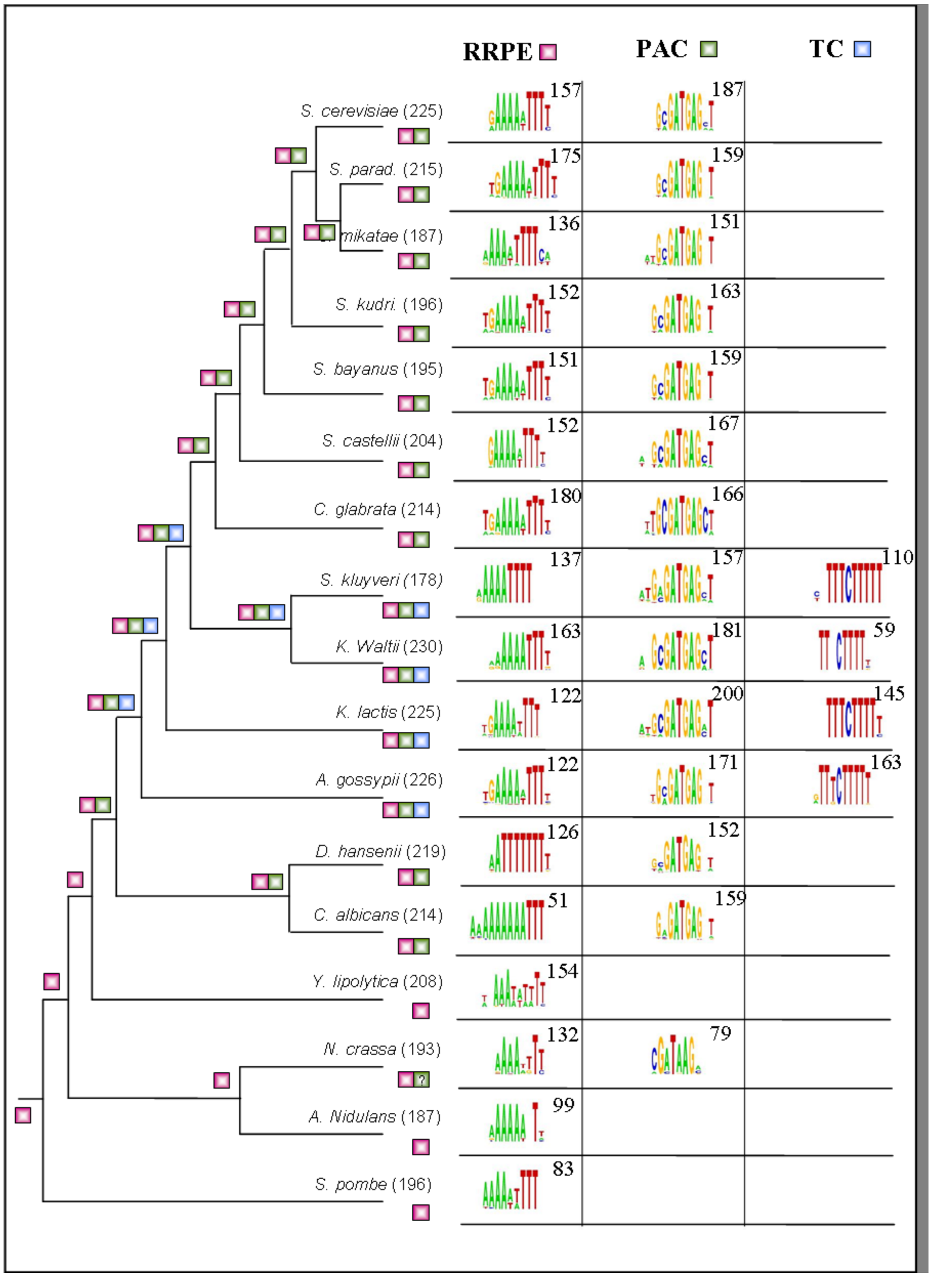
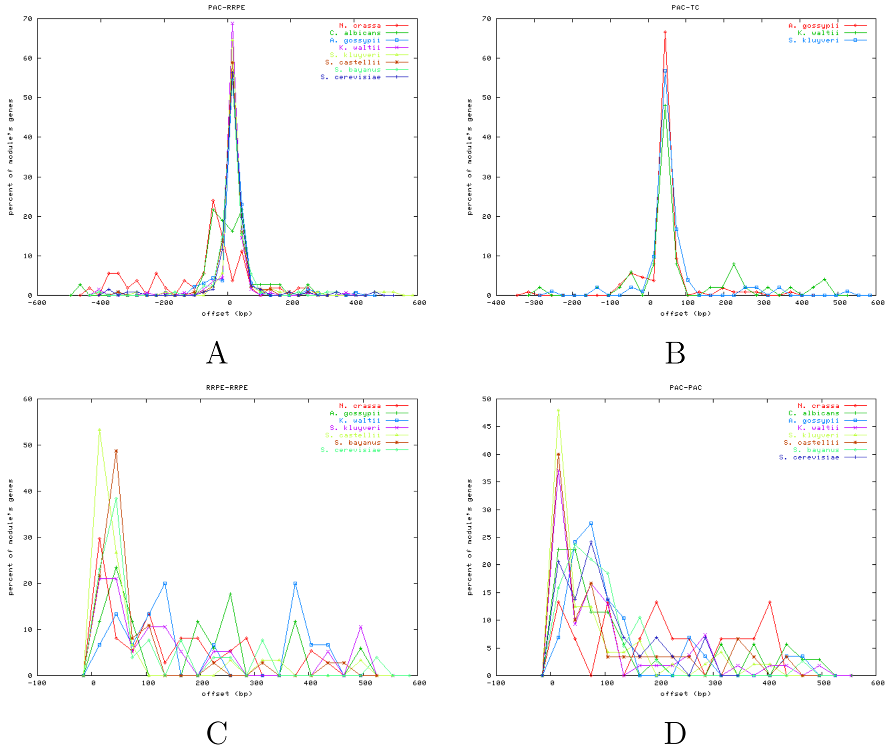


Figure 7.11: **Evolution of the regulatory program in the ribosome biogenesis module.** A schematic phylogenetic tree (branches are not drawn to scale) representing the known phylogeny [84] of the 17 analyzed species is shown, together with the sequence logos of the main cis-elements enriched in each module's promoters, grouped into three distinct types (colored boxes): RRPE (magenta), PAC (green) and a TC-rich sequence (blue). The total number of genes in each POM is shown in parentheses, and the number of genes that contain each motif is indicated as well. While the RRPE element is conserved in all 17 species, the PAC element emerged only prior to the *N. crassa* or *C. albicans* speciation, whereas the "TC element" (TTTCTTTTT) is specific to *A. gossypii*, *K. lactis*, *K. waltii* and *S. kluyveri*, possibly following rapid modification of the regulatory network before the speciation of *A. gossypii* and after the divergence of *S. kluyveri*. The GATA motif in *N. crassa* differs from the PAC element in one critical position.



A

B

C

D

Figure 7.12: **Distributions of the relative distance of cis-elements in promoters of RB genes.** Shown are distributions of the distances between pairs of cis-elements in different fungal species. Tight coupling is observed for all three factors (TC, RRPE and PAC).

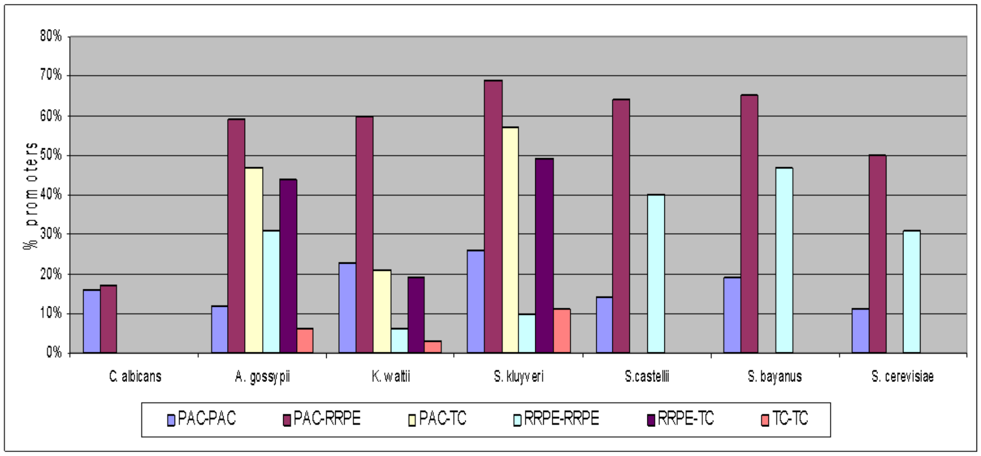


Figure 7.13: **Co-occurrence distribution of cis-regulatory elements in promoters of RB genes in specific species.** The percentage of promoters bearing each type of pair of cis- elements is shown for each of the species. We observe that co-occurrence of two RRPE sites in the same promoter is rare in *K. waltii* and *S. kluyveri*. This may be due to the presence of TC elements that serve as a substitute site in these species. *A. gossypii* genes, which also have the TC element, display more redundant promoters in general

7.9 Buffering and evolvability in transcriptional modules

We hypothesize that two major different trends shape the functional interaction among different cis-elements and influence their evolution. The first trend ("conservation") occurs wherever there is a specific regulatory role for each of the cis-elements, and selection conserves a particular combination of cis-elements present in each gene's promoter. The second trend ("buffering" [62]) occurs in cases where two cis-elements have a similar regulatory role, and increases the redundancy in the regulatory mechanism. We estimated the relative contribution of these two forces at different stages in the evolution of the RP regulatory mechanisms. For each element we produced a "cis-affinity profile" that indicates which genes it occurs in and at what strength. We then compared these profiles between species separated by various phylogenetic distances by computing their pair-wise correlations. When conservation is a dominant trend, we expect the same cis-regulatory element to appear in the same genes in two different species, as the element's function cannot be easily complemented by substitution with other elements active in the module. Hence, a high correlation between the corresponding profiles should be observed. On

```
>4266 CGTTTGAGCCTTGCAGCGATAGGAATGTCAACTGAAAAATTTGGCGATGCCCTAGCACCAGCTCGAGAGAAGACAA
>5919 GATTTGGGACTAAAAAAAGAAAGGTTATCTACTGAAAAATTTGAATTTATGCGATGAGCTGAAGAGTATAGTCGCT
>686  GCAGCTCGAGGATATCGTTTCTTTTGGTGGTACGAAAAATTTTAATTCGAGATGAGA-------------------
>1496 GTGATACTATGTGTCGTTTCTTTTTGTTGAATCGAAAAATTTTATGGCATCGATGCCTTGCAATTGTTTTATGAGA
>3597 GTATATAGAACAAATCGTTTCTTTTTTGGACTCGAAAAATTTTAGCCTACGCGATGAGATGAGCTACATTAAACTG
>4013 GCTGTGGCAGGGCGCTGGCGTCGTAGGGTTGGCGAAAAATTTTCAAGCTATATGCACTAGAGGCATCTCATCTTGC
>4104 AGTCGTTTCTTTTTTTCCATGCGACAGCCGCTCGAAAAATTTTCTCTCATCGCAGAATCTTTAGGCCTAAAGAGAC
>4575 GTCACGTGATTCCAGTGTTGTTTCTTTTGCCACGAAAAATTTTAAGTGTTGATGAGATGGGCCATTGCCCGCTATT
>5582 TTCTGAACGTTGCAAAAAGCAATGCCCGAAATCGAAAAATTTTCAGCATTTCATTATGTGAAAGCGGATAAGCTCA
>1634 TTTTGAACGAATTTATCTCGTTTCTTTTTTATTGAAAATTTTGTAGCGATGCGATGAGCTCGAGCCGGTATCGAAC
>2135 GTTATCCGGGTAACCACGTACAAAACGATATTTGAAAATTTTGTATTCACATTGCTTTTTGGTCATTAGCCTGCCC
>3764 ATCAAATATCGTTTTATACGTATTTCAGTAGGTGAAAATTTTGAACTTCCGATGCGATGAGGTTAGCGATGAGCAC
>5093 GCATTTTCCGGCATTGCATAACCGTAAAGGGCTGAAAATTTTGATAGCAGGTCATCGCATGTTGAAGAGGAATAAG
>6199 TTTGAAGGTTCGCGGAATCGTTTCTTTTTATATGAAAATTTTGAAGGTCCTTTGGAAGGCGATGAGAAGAAACGCC
>157 C GGGGTGGTCACGTGACTGGATGCTCTGCGCGCGAAAAATTTTTGCCGTGATAACTTCGATGGTCAAGAACTGTTG
>1094 TTTTTCAGAGGGGACAATGGGCTTCGAGATTTCGAAAATTTTTCGAGGGCCGGTGCGATGAGATGAGGTGAAAGAG
>2494 CGAATAACATATCATGAGTAATAATCATAAGGCGAAAATTTTTAGTGAAATATTAAAAAATTGAGCAGGTGACTGA
>2673 TGGATCGTTTCTTTTGTTGCTTTGGCGGAAGCCGAAAATTTTTTATCATCGTGCACAGCGGGCTCATCTCGTCGAT
>4960 TATAAATCACGTGACCTGAATTTTCGGTGATTCGAAAATTTTTGAGATGCGATGAGCTATGGTGTAACGCATAGTT
>5985 CTCGAGCTGGAGCTTTTCTTTTTCGTCCATCCCGAAAATTTTTCAACTAAAATAGCGTGCGATGAGCTCGTTACTG
>941  GCATGTGCTCATAGTTCCGTAAATCGTGATTTGGAAAATTTTCTGCTCATCGGCCTTGGAAGCCTGTATCTGAGCT
>2784 CTAAATTCGGGGAAGGTATAGTACGCAAAACTGGAAAATTTTCGCACACTCGAGAGCTCATCGCAGTTCATCGCTG
>3151 AAGACAGTGTGTGGGGCACGTTTCTTTTCGAGGGAAAATTTTCAGCTTAAGCTCATCGCATGACAAGGCCGATGCC
>3706 GTGCGACAGCATCTTGTTAGGATAGAGTACTGGGAAAATTTTCCAAGCTGCAAAATAGTAAACGGCACTCATCGGC
>4166 TTGACTAAAGCCCCCAAGCATGGAAGACCTTTGGAAAATTTTCAGTCTGAGATGAGCTATGATGGTACAAGTAGAA
>4742 ATCACGAGACAAAAAAGAAATGCTGACTGTATGGAAAATTTTCTGAGATGAGATGAGCTTAGGTACAAGTCGTTAG
>4786 TCGGAGATCGCGCAGCACGGAAAAAGAAAGATGGAAAATTTTCCAGCTGGCGGCCCTGAAAAGCGGCGATGAGCTG
>5165 TGCTTTTTGATTACGATGTGACTAGCGAGCTAGGAAAATTTTCCGCATTTTAATGTCATGAGCAGAGGCATCGAGA
>173  ACGCAACGACGTTACTTTTCTAACCGGACGGGCGAAAAATTTCACGGAATAGTATCAGCATCTCATCGCCTATATA
>1052 GCTACTCTACATGAGACACAAAAAGCAATTATCGAAAATTTTCACCGGTAAGCCTGCCGCGGGCTCGACGCCAGCC
>1449 TGATGCCTACTCAAAAAAAAAAGAAACGATATCGAAAATTTTCAGCCTAATTATGGGAGCTCATCTTGAGCTTAAT
>5378 ATTTTCTCCAGCCGTATGCTTCTGGACGGAACCGAAAATTTTCCCGTTCAGTAGGTCTAACTTTTGGGTCTACATA
>285  GAATTCGCTTGCTGTCTAAAAACCAATCTACTAGAAAAATTTTCCGAATGTTATAATTATACCTTCGCGATGGCGT
>764  GTAAGCACGTGACCGTGATATTTCTTTTTTTTAGAAAAATTTTAGCGGTCTTGATTAGAGTGGCTCGATGAGATGA
>2904 AAGCTTATATTTCCTGAACATCTCTCGTACTCAGAAAATTTTTTGGGTCAGCATCCGGCATGTCACTTGTCCCTCA
>3899 TGGGTGCTAGTCCCGACATGGACCACGAGATGTGAAAAATTTAGCCGTAGCTCATCTC------------------
>6173 TAACGGCTGGCATTTCTTTTTGTATCGAGCATTGAAAAATTTATTCCGGGCCTGCAACTCATCGCCATCAAGCGCA
>698  GCTTAGATATTAATTCGTTTTTAATGATACATTGAAAATTTTAGTCCACTGGTTATTCTCCTACAACAGAAAGGTG
>1076 TTAATCGCGTGTAGTGTGGTGTTAAGGTGTTTTGAAAATTTTAGGCTCTGCATAAAACATTATCGA----------
>3100 TAATGGTGATAGTTGATAAAAAGAAACGATACTGAAAATTTTAATGGTTACCAATCTCATCTCATCGCCATACTGA
>4535 CAGTACCGGATAACGATCACGCGAAACGAATATCAAAAATTTTTCAGTGGCTTACTGCTCGCGATGAGCTGAAGCA
>4967 TATATACGTGCTTACCCGGACAAGTGTCAGCTTCAAAAATTTTCCTCAGATCAGGCTTGATGAGCTTCCCGCATTA
>1113 TCACTATCTCTCGAGCTCATCGCGATGGCCCTTCAAAATTTTTTCAACTATAAAAAAGCAATTTAAAGCACGTGCT
>3826 ATAGTCCAATAAAAAGCAATGTACCTTAATGTGGAAAATTTTGCAACGCG--------------------------
>611  TCTTTTTCATGAAATTTTCACCATCTATATTCAGAAAATTTTCAAGCGATGAGATGAGCTGTGGGTATAGCGGAAG
>3441 GAATCACTCACTTTCTTTTTTTCTGACTGTACAGAAAATTTTCCTGCGATGAGCTTTGCACAGTATCGACTGTAAT
>4766 CAGTCAGACATTAATCGTTCTGCACGTTGTGGAGAAAATTTTCCGAGATTGCATCACCGAGCCTGAAATCGCATGC
>5718 GTTTGTATCATTTTTTACGTTACAATACGTACAGAAAATTTTCACATTAGACGCCAGTCCTCATGCGATGAGATGA
>1294 ACTTTTAGCACTGCTGGTCTACCGCACGAATCTGAAAACTTTTATGGATAACGTCTTTAGACTATCGTTGTGGCGC
>2757 GGCGTTTCTTTTTTTCAGGTGTAGAGGATCCATGAAAAGTTTTCATACTCATCACACTTCAACTTATACAGGTCAT
>4049 TAAGAATCGACGATATAACGTTCCTTTTTTGGTGAAAAGTTTTTGATCCTATCTCATCGCATACCAGAGTATAATC
>5956 TTGCTAAGGGTTAGAACTACGCGGATGGAACCTGAAATTTTTTTCTTGCTGGAACTGGACCATCCACCAATTTTAA
>5214 CACGGGAGTGCGGGTAACAGGCGACCTTAGCATCAAAATTTTCACCAGTAGCACCGCCATGAGATGAGATGAGCAG
>6000 GTCAGCTAGAAAAGCAATACCGAGGTCTGCGGTCAAAATTTTCACTGTATAGGCGGGCTCGAGGATATGAAGAGGT
>1158 CGAGAAGGTATGGTAATTATGTAAGCGCGCCATAAAAATTTTTCAGAGGCGATGAGATGAAGCACCTCTGCTCACT
>5033 TGACCATTTCTTTTTCATCTTCACTGTAAACGTAAAAATTTTTCAATTCATCAGCATACGCTACCTGCCTACCTCA
>3296 TCAATATGTACCACACCAGTATTCTTTTTACATGGAAAATTTTAGTACTACGCCATGCGATGAGTTCCAGAACACA
>5246 CATGTGATCACTATCTGTATTTCTTTTTGAGTTGGAAAATTTTATCTGGATAGCTACCTCATCGCGAGCTCGAAGT
>161  GTATGGGTGTGCAGTCACGGTCATCAAGGCGGTGAAAACTTTCAGGCCATCTCGCGGTCTCTCTAGATACTCAATA
>1465 GACTTCATATTATACGCCTTCTCTTATGTGTCTGAAATTTTTCACCGGCTATGAGCACTGGCATCGAGCACCGATC
>5618 GGACGTAGACGTCAAAAGTCCTGCTTTAAGACTGAAATTTTTCAGTTGGTGAACTCATCTCATCGCATCTCGTCTT
>398  TTTTGTCATCACGATGCTTCGAGCTGAATCGTTGAAGAATTTTCGAGTGAAAATTACGATGAGATGAGATGCGTTT
>1904 TTTTAAATTATTCCAGCATTACTTTAATCACGTGAATAATTTTCACAAAGATTTTACAATACTATTATTTTTTGAA
>3673 TCTTGATTCAAACGGTTATTACTTTTCCGCAGTGAAAAAATTTCGACCATGCGATGAGATGAGCTTGAGCACACAC
>3898 ACCCTGAATCAGTGGGCGTCACTTGCCGTTGGTGAAAAAGTTTCACTGATGCTCGAGAATCTCGGTGCGATGAGAT
>1691 CAGAAAAGTCCGCACGAGACTCACGGATTCAGTGAAAAATCTTGCATCCGGTTGAATAGTGTGGATATCCGGAAAG
>3318 GCTGGCGCACGTGACCTCTCACGTGACCTTCGTGCAAATTTTTTTGCGCGCGCGCCGCGCACCGGCAGCTTATCTG
```

Figure 7.14: **The TC-RRPE-PAC cis-regulatory module in** ***A. gossypii*****.** Shown are the sequences flanking the highest scoring RRPE site (magenta) in the promoters of RB genes in *A. gossypii*. RRPE sites (magenta), PAC sites (green) and TC sites (blue) are highlighted. The three sites are tightly organized into a cis-regulatory module, centered on RRPE, with the TC site upstream and the PAC site downstream.

the other hand, when buffering is dominant, and two elements are interchangeable, each should exhibit a lower correlation when comparing related species.

Specifically, for each combination of motif m and species s, we generated cis-element affinity profile as a vector $L^{m,s}$, where $L^{m,}s(g)$ is the PWM likelihood score of the best hit of motif m in gene g's promoter. We then computed inter- and intra- species correlations of such cis-element profiles by calculating the Spearman rank correlation of each pair $L^{m,s}$ and $L^{m',s'}$. We derived a p-value for rejecting the independence assumption corrected for multiple testing using Bonferroni's factor. In principle, species at different evolutionary distances may exhibit different levels of promoter conservation, resulting in different background levels of affinity profile conservation. However our analysis shows that for evolutionary distances beyond the sensu stricto Saccharomyces clade, this background level is negligible.

We found several cases of significant correlation between cis-element profiles (Figure 7.15). For example, the profiles of IFHL elements are often highly correlated with each other across different species (e.g., *A. gossypii*, *K. waltii* and *S. castellii*), suggesting that these elements play a distinct role in the regulation of a specific subset of RP genes in some species (Panel a). In other cases, we find correlation between different cis- elements in different species, suggesting that one element assumed the functional role of the other. This is the case when correlating the profile of Homol-D box elements in the species that are in the process of losing this site (*S. kluyveri*, *S. castellii*) and the RAP1 profile in species that have completely lost Homol-D (*S. cerevisiae*, *S. bayanus*, panel b). Finally, we observe cases where the correlation between similar cis-elements in different species is low, a fact that may indicate the aforementioned buffering trend. For example, the RAP1 profiles exhibit low correlation within some of the species (e.g. between *A. gossypii*, *K. waltii*, and *S. castellii*) (panel c). A similar low correlation is found between the Homol-D profiles in the same set of species (panel d). This may indicate that the two motifs are functionally redundant, and thus their particular set of targets is not conserved. Note, that while the affinity profiles are statistically robust and provide interesting insights in some cases, other findings cannot be explained in a straightforward way by the two evolutionary processes we considered. These include the high correlation between IFHL profiles in *N. crassa* and *S. kluyveri* and the medium correlation between Homol-D profiles in *S. pombe* and *S. kluyveri.* Overall, it appears that both conservation and buffering play a role in shaping the cis-regulatory mechanisms associated with the RP module in the 17 analyzed species.

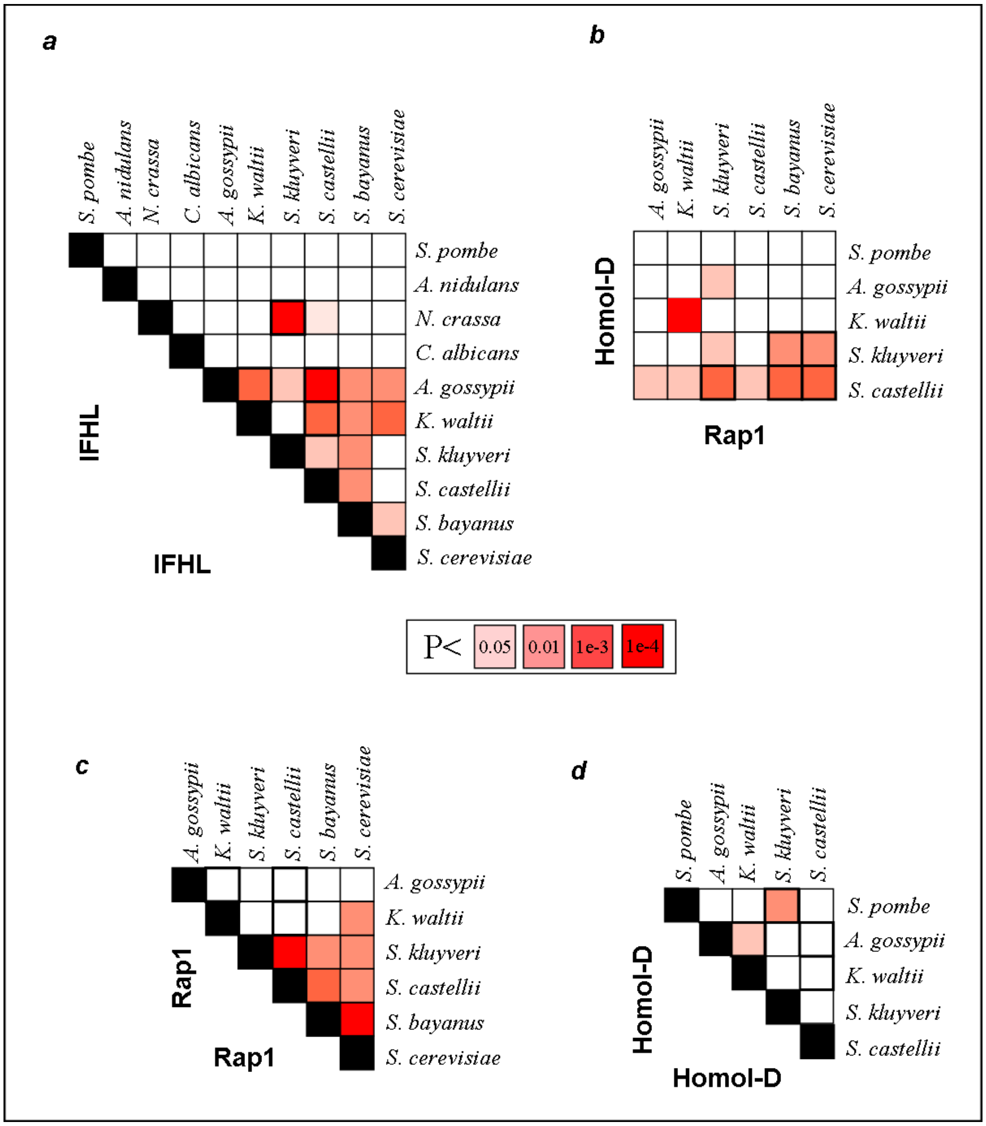


Figure 7.15: **Conservation and buffering in the evolution of RP regulation.** Shown are color-coded correlation diagrams between pairs of cis- element affinity profiles across different species. The color intensity (middle scale) of each entry indicates the significance of the correlation coefficient between the corresponding pair of cis-element affinity profiles in the two species (white entries are insignificant). Darker entries indicate that the pair of cis-elements may be functionally conserved in a gene specific manner between the corresponding species. (a) IFHL correlations. We observe a significant correlation between IFHL elements in *A. gossypii*, *K. waltii* and *S. castellii* and a weaker correlation in species closer to *S. cerevisiae*. (b) RAP1 and Homol-D correlations. We observe significant correlation between the Homol-D profile in *S. castellii* and *S. kluyveri* and the RAP1 profile in *S. cerevisiae* and *S. bayanus*. (c) RAP1 correlations. We observe lack of conservation in the gene specific affinity profiles of RAP1 in *A. gossypii* and *K. waltii*, two species in which both RAP1 and Homol-D sites are associated with RP promoters. This is in marked contrast to the significant correlation between RAP1 profiles in species that completely lost the Homol-D box (*S. cerevisiae* and *S. bayanus*). (d) Homol-D correlations. As with RAP1 sites, we observe lack of conservation between the gene specific affinity profiles of Homol-D in *A. gossypii*, *K. waltii*, and *S. castellii*.

7.10 Modes of evolution in transcriptional modules

The evolution of transcriptional modules is an important aspect in understanding regulatory networks. Previous studies have suggested that groups of genes that are orthologous to *S. cerevisiae* expression modules are frequently regulated by conserved cis-elements [44]. Our analysis demonstrates that the regulatory mechanisms associated with ancient and tightly conserved transcriptional modules can often be remarkably diverged. The results in this chapter suggest a general framework for the study of the evolution of module regulation, including full conservation of binding site and transcription factor, gradual changes in a single DNA binding site, simplification and elaboration of existing programs and even dramatic events of element infiltration and loss that result in transcription factor switching (Figure 7.16). In particular, we suggest that the formation of a redundant and coupled intermediate program may explain how a coordinated response may be conserved even though the underlying regulatory mechanisms are changing. This dynamic view of regulatory network evolution is consistent with previous studies on rapid promoter evolution [94, 93, 152], and with the known relative flexibility of cis- regulatory sequences compared to protein coding sequences. Our analysis implies that specific evolutionary processes exploit the dynamic nature of promoters to continuously modify the level of redundancy in regulatory mechanisms. Such redundancy may provide a buffering capacity [62] and may be important for the evolvability [82] of the regulatory program. Additional data and further studies are required to validate our hypotheses and fully elucidate such processes. For example, we still lack experimental evidence as to the redundancy of the various sites in the intermediate programs, and we do not know how a large number of novel binding sites are introduced in a coordinated fashion, whether the coupling of elements we observed within promoters facilitates or constrains the evolution of regulatory programs, and the exact rate of sequence changes necessary to introduce a novel motif.

The results discussed above have significant implications for the study of transcription regulation in an evolutionary context. We have shown that computational techniques, merging new data with well-established evolutionary concepts, facilitate improved integration of genomic (sequence) and phenotypic (expression) data and their synthesis into a coherent reconstruction of the evolution of regulatory net-

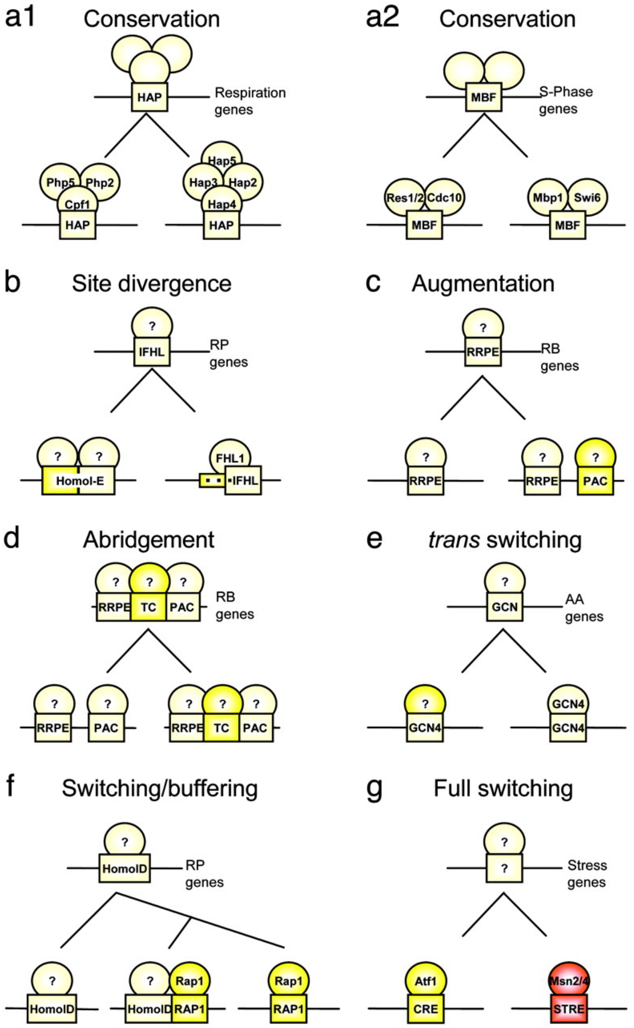


Figure 7.16: **Evolution of transcriptional modules regulation.** We summarize the results of our analysis for several transcriptional modules and their regulatory programs. The schematic figure represents the evolution of an archetypical module's gene promoter, but importantly, the evolutionary process is applied to dozens of promoters simultaneously. Conservation of the modules is sometime induced by conservation of both at both the cis- and trans- regulatory levels, but in other cases our results suggest gradual changes in the binding sites, elaboration of the transcriptional program by emergence of new cis-elements or even a process by which one cis-element is replaced by another through an intermediate phase of a redundant program. The regulation of modules thus evolve at both the cis- and trans- levels, and is flexible even when the module contains dozens of genes and is functionally conserved.

works. The evolutionary context is crucial for the exploitation of these data and greatly enhances the potential of comparative methods [128]. Whereas previous research in comparative genomics of regulatory networks focused on the identification of conserved cis-elements [23, 79, 108], our results emphasize the importance of accounting for changes - both gradual sequence divergence and dramatic innovation processes. Finally, the putative buffering effect of redundant regulatory elements that we report here may be instrumental in enabling rapid evolutionary change of regulatory networks, and may play a major role in metazoan eukaryotes. The typical animal promoter is organized into cis-regulatory-modules [96] that contain multiple, often redundant, binding sites. It is possible that this organization is a consequence of evolutionary processes similar to those we report here, that are essential for the emergence of the increased complexity and evolvability of animals' regulatory networks.

Appendix A

Functional annotation of gene sets using the GO hierarchy

A.1 Introduction

Many of the current methods in functional genomics, both experimental and computational, generate sets of genes as their output. For example, gene expression experiments are used to identify sets of differentially expressed genes or clusters of co-regulated genes. Successful analysis of these experiments relies on the association of gene sets with biological function. In many of the applications, known functions of genes in each set are used to predict the function of the whole set in a computational procedure that is called *functional enrichment testing*. Rougly speaking, we wish to check if a particular gene set contains an unusually high proportion of genes from any particular category, compared to the complete collection of genes. To facilitate such analysis, databases for gene functional annotations (for example, the important Gene Ontology database [50]) were formed and a considerable community effort is continuously put into their update.

A major concern with testing functional enrichment for a gene set vs. a modern gene annotation database is the extensive multiple testing that is performed when considering thousands of different functional categories in repeated statistical enrichment tests. The problem is particularly acute since the number of defined functional categories is rapidly increasing and, moreover, different categories interact in complex and sometime hidden ways so that naive schemes for multiple testing correction reduce sensitivity unjustifiably. To address this problem, we have developed a program called TANGO (Tool for ANalysis of GO enrichment). TANGO performs functional enrichment tests that fully account for multiple testing using a rapid implementation of a simple resampling algorithm. The program utilizes several methods for filtering similar annotations and also corrects for multiple testing in cases where many sets (e.g., all clusters) are screened at once. The TANGO algorithm thus allows functional testing using the entire GO hierarchy and provides reliable and parsimonious results for single or multiple gene sets.

A.2 TANGO main features

- Test functional enrichment of a collection of target gene sets given large databases of functional annotations (GO, MIPS)
- Stringent, yet sensitive correction for multiple testing providing more power than the standard correction schemes that assume independence between the tests (Bonferonni/FDR [12]).
- Test enrichment using the entire annotation vocabulary, finding the best level of granularity by testing all the hierarchy, eliminating the need for pre-processing and simplification of the annotations used (as in, e.g., GO Slim).
- Efficient implementation allows testing to be performed interactively with upstream analysis algorithms (e.g., clustering, biclustering).

A.3 Correcting for multiple testing

The basic statistical test that is commonly performed to test enrichment of a target gene set for genes annotated with a particular function, employs the hyper-geometric distribution. According to the test (also called the Fisher exact test for independence), we are observing a *background set* of n genes, m of which are annotated with a certain function (the set A). Given a target set T of m' objects, the probability that the intersection of T and A is of size k is:

$$Prob(|A \cap T| = k) = hg(n, m, m', k) = \frac{\binom{m}{k}\binom{n-m}{m'-k}}{\binom{n}{m'}} \tag{A.1}$$

The p-value for intersection of size k or larger between A and T is thus:

$$EnrichmentPV(T, A) = Prob(|A \cap T| \geq k) = \sum_{j \geq k} hg(n, |A|, |T|, j) \tag{A.2}$$

Given a database of functional annotations, including a collection of terms and their annotated sets of genes A_i, we use this formula to determine which of the annotation terms is enriched within the target set T. We define:

$$MinEnrichmentPV(T, \mathcal{A}) = max_i(EnrichmentPV(T, A_i)) \tag{A.3}$$

When the number of terms is large, the p-values we derive have to be corrected, since we test many hypotheses and can obtain low p-values even if the target set T is completely random. Assuming the sets A_i were independent, one could have applied a standard correction procedure (Bonferroni for controlling the maximal p-value, Benjamini- Hochberg for controlling the false discovery rate (FDR)). However, when the sets A_i are highly dependent, such correction may be too stringent. Consider for example two GO terms that have exactly (or almost exactly) the same set of annotated genes in a certain species. The p-values for enrichment in both of these sets are completely (or almost completely) dependent and we need not correct for multiple testing. The dependencies between annotation terms can be completely characterized using the sizes of their gene sets intersections, but as these intersections define an arbitrary dependency structure it is difficult to analytically describe the appropriate correction procedure for multiple testing. In other words, no closed form formula is known that determines, given arbitrary sets A_i and a target set T, the distribution of $MinEnrichmentPV(T, \mathcal{A})$.

To cope with this problem, TANGO takes a simple approach, and computes the empirical distribution of the maximal enrichment p-value by sampling a large number of random gene sets and computing their p-values vs. each of the annotation sets. Formally, TANGO estimates:

$$\begin{aligned} &CorrectedEnrichmentPV(T, \mathcal{A}) = \\ &\tfrac{1}{B}|\{j | 1 \leq j \leq B, EnrichmentPV(p_j(T), \mathcal{A}) \leq EnrichmentPV(T, \mathcal{A})\}| \end{aligned} \tag{A.4}$$

where p_i are random permutations of the genes background set (see comment below), and B is the sample size. When annotating a collection of gene sets T_i in a single analysis, we also have to correct for testing each of them separately:

$$\begin{aligned}&CorrectedEnrichmentPV(T_i, \mathcal{A}, \{T_1, \ldots, T_l\}) = \\ &\tfrac{1}{B}|\{j|1 \leq j \leq B, MinEnrichmentPV(\{p_j(T_l)\}_l, \mathcal{A})) \leq EnrichmentPV(\{T_i\}, \mathcal{A})\}|\end{aligned} \tag{A.5}$$

In practice, we compute corrected p-values for a collection of gene sets T_i and an annotation database A_j by estimating the distribution of enrichment p-values in permuted genes sets $p(T_i)$. The idea is that we keep all of the relations among annotation sets A_i and among target sets T_j, but we decouple any dependency between them by applying a random permutation on the gene ids used by the T_is. Finally, we note that if the analysis generating the target sets included only part of the genome (because of, e.g., the ensemble of genes printed on the chip), we should perform all the analysis with an appropriately chosen background set. To do this we intersect all annotation sets with the analysis gene subset, and assume the basic hyper-geometric tests to be performed with a universe including only these relevant genes. This precaution can prevent many artifacts, for example, discovering enrichment for cancer genes in clusters that were generated from data derived by a chip that contains only cancer related genes.

A.4 Filtering Redundancies

Gene annotation databases are designed for maximum flexibility and provide generalized biological vocabulary that can support diverse species and model systems. The organization of the vocabulary is hierarchical and often several level of the hierarchies are not relevant for a specific model system. As a result, when testing functional enrichment, one can detect several annotation terms with significant corrected p-values, such that all of the terms reflect essentially the same set of genes. To avoid reporting such redundant terms, TANGO performs a redundancy filtering procedure based on the conditional hyper-geometric test. The idea is to start with all annotation terms that got a significant enrichment p-values for a certain target set, and then test if given one of the enrichments, other enrichments are no longer significant. Formally, given a target set T that is enriched with genes from the set A', we test if T is enriched with genes from another set A, assuming we already know the size of intersection between A' and T and between A and A':

$$CondEnrichmentPV(T, A|A') = \sum_{k \geq |T \cap A \cap A'|} hg(|A'|, |A \cap A'|, |T \cap A'|, k) \times \sum_{l \geq |(T-A') \cap A|} hg(n - |A'|, |A - A'|, |T - A'|, l) \tag{A.6}$$

To filter redundancies, TANGO performs a greedy algorithm. First, all annotation sets are sorted by their p-values of enrichment in the set T, generating a ordered list $A_{i_1}, A_{i_2}, \ldots, A_{i_k}$. Second, the list is traversed, and only sets A_i for which $CondEnrichmetPV(T, A_i|A_j) < \tau$ for all $j < i$ are reported. The parameter τ can be modified to control the allowed level of redundancy. We note that three types of typical redundancies can be detected. The first scenario is where a general annotation terms is enriched in the set T and as a result, several of its specializations are also enriched (although their p-values will be typically larger). In this case TANGO selects the more general term and removes the specializations. The second scenario is where a specific annotation term is enriched in the set T and as a result several of its generalizations are also enriched. In this case TANGO selects the specialization and removes the generalizations. The third scenario is where two terms have almost identical gene sets associated with them. In this case TANGO arbitrarily chooses one of the terms.

A.5 Implementation notes

TANGO is implemented in C++ and is provided for Linux and Windows. The implementation makes heavy use of bit vectors operations that are optimized for finding the size of intersection between a target set and annotation set. TANGO can annotate hundreds of target sets using thousands of annotation sets with 1000 bootstraps iterations in minutes on a standard PC. The programs is accessible from the Expander system, but is also provided sperately with a standalone interface that is documented below.

A.6 Reference manual

```
Using the TANGO program
----------------------

TANGO is a program for functional annotation of gene sets. It uses pre-processed
tables of genes and their GO annotation, and performs hyper-geometric enrichment
tests for sets of genes with common annotation and sets of genes given as input.
Importantly, TANGO corrects for multiple testing at both the multiple GO classes
and multiple tested sets levels. It does so by bootstrapping and estimating the
empirical p-value distribution for the evaluated sets.

TANGO is a standalone program available for Linux and Windows. Precompiled
annotation tables are available through the Expander system. See
www.cs.tau.ac.il:/~amos for downloading the current standalone version.
and www.cs.tau.ac.il/~rshamir/expander for precompiled annotation files for
various species.

1.Running TANGO
------------

You should run tango like this:

tango parameter_file

TANGO will read files as specified in the parameter file and will write its
output to yet another file (that is specified in the param file). Below
you’ll find a description of the files and parameters.

2.Precompiled annotation files
----------------------------

TANGO use preprocessed GO annotation files to map genes and known annotations.
All files are tab delimited text files. A set of tables always describe
```

```
the annotation of a single species.

varob.txt - mapping variable internal ids with external gene identifiers
(ORF, Locuslink etc)

Field 0: Internal variable id (Number)
Field 1: Variable Name (String)
Field 2: Variable External Key (String)

Example line:
21043 TP53 7157

goclskey.txt - key file containing the names of all annotation categories.
Typically, each category reflects one GO attribute, and the gene associated
with it those annotated with this attribute, or with an attribute
that specialize it.

Field 0: Internal annotation category id
Field 1: GO id (or any external id for the annotation source)
Field 2: Category name
Field 3: Number of genes annotated with this category (not used by TANGO)

Example line:
0 GO:0008289 lipid binding 15

clsassoc.txt - this table associates categories with variable internal
ids (key to varob.txt)

Field 0: Category id (key to goclskey)
Field 1: Gene id (key to varob)


3.TANGO Input files
----------------
```

TANGO processes two input files. One defines the set of genes that should be considered as the background. Typically this set can include the entire genome, or only the genes that were printed on the chip that was used to generate the clusters/biclusters, or only the genes that survived the filtering that precede the analysis that generated the gene sets. The second file defines the actual sets (clusters/biclusters) to analyze.

chip.txt - defines the background set

Field 0: Gene external key (points to field 2 in the varob table). In other words - a list of locuslink ids (mammals), orf codes (yeast), flybase ids (fly) etc.

sets.txt - defines the sets to annotate

Field 0: Gene external key (points to field 2 in the varob table)
Field 1: Set Id (serial number for the sets to annotate)

Example:

```
YOR348C 0
YPL265W 0
YPL274W 0
YAL067C 1
YBL042C 1
YBR021W 1
```

This example defines two sets of yeast genes, each with 3 genes.

```
4.TANGO output file
------------------
```

TANGO generates a tab delimited text file including all significant annotations. The format is as follows:

```
Field 0: set id (key to sets.txt)
Field 1: annotation name (name from goclskey)
Field 2: uncorrected hyper-geometric p-value (log10)
Field 3: Corrected hyper-geometric p-value (log10)
Field 4: fraction of genes in the set annotated with the category
Field 5: number of genes in the set annotated with the category
Field 6: category external id (field 1 in goclskey)

5.TANGO Parameter file
-------------------

TANGO comes with a parameter file that controls the input files it
uses, as well as important algorithmic parameters. The file is formatted as
an INI file - including "scopes" (bracket delimited names in their own lines)
and "options" (assignments of values to parameter in the format
options=value). The ordering of options is not important as long as each
option is below its appropriate scope.

Here is an example of the parameter file, explanations are below:

#file starts here
[Random]
Seed=19
[Tables]
varob=/data/yeast/varob.txt
goclskey=/data/yeast/annots/go/goclskey.txt
clsassoc=/data/yeast/annots/go/clsassoc.txt

ChipOrfs=chip.txt
SetsOrfs=sets.txt

AnnotReport=annots.txt

[TANGO]
BootstrapNum = 1000
```

```
MinClsSize=5
MaxClsSize=1000
MinClsInter=4
MaxPvToRep=0.01
FilterRedPVThres = 0.05
#file ends here
```

Random::Seed - controls the pseudo-random sequence used for bootstrapping. Runnig tango twice with the same seed and same data will generate the SAME results.

Tables::varob - the full path + name of the varob file (see section 2)
Tables::goclskey - the full path + name of the goclskey file (see section 2)
Tables::ChipOrfs - the full path + name of the chip.txt file (see section 3)
Tables::SetsOrfs - the full path + name of the sets.txt file (see section 3)

Tables::AnnotReport - the tango output file (see section 4)

TANGO::BootstrapNum - number of bootstrap iteration to perform. The corrected p-value will always be larger or equal 1/BoostrapNum, but since the output report provides the uncorrected value as well as the corrected one, using 1000 should be generally enough. We recommend using 1000 bootstraps to determine which annotation is significant and the raw hypergoemetric p-value to further understand the strength of functional association.
Note that the number of bootstraps linearly affects the running time of the program (naturally), so use it carefully.

TANGO::MinClsSize - the minimal size of category to consider for annotation. Categories that have less annotated genes than this number will not be considered. Use this to save time and reduce the abundance of spurious results.

TANGO::MaxClsSize - the maximal size of category to consider for annotation. Categories that have more annotated genes than this number will not be considered. Use this to prevent very general annotations (e.g., metabolism).

TANGO::MinClsInter - the minimal number of genes that are annotated with the category and are part of the annotated set to be consider for annotation. Setting this to 0 will allow annotation using a single gene, which are prone to false positives. Although these will be corrected by the bootstrap procedure, we recommend to set this value to >3 to increase the statistical power.

TANGO::MaxPvToRep - the maximal p-value (uncorrected) to report on.

TANGO::FilteredPVThres - the maximal conditional p-value to consider when filtering annotations of the same set. TANGO filter results by performing conditional hyper-geometric tests for one category, assuming the observed enrichment in the other. Whenever this conditional p-value is higher than the threshold set by this parameter, TANGO will remove the weaker annotation of the two (see TANGO technical report for more details)

Appendix B

Finding cis-regulatory motifs in gene modules

In this appendix we describe a program for finding enriched cis-regulatory motifs in the promoters of gene modules. The approach is heuristic and combines combinatorial scoring with standard PWM models. We experimented with it extensively, see for example the results in Chapters 4 and 7.

Throughout this appendix we assume that we are given a set of putative regulatory regions (which we will call promoters), one region s_v for each gene $v \in V$. We are also given a gene module $B \subset V$. Our goal is to find sequence motifs that appear in $\{s_v | v \in B\}$ significantly more than expected by their frequency in the entire genome.

Scoring and searching for DNA *k*-mers. The first and most simple type of sequence motifs we consider are exact DNA words. We treat such words over an alphabet including the wildcard character "*" to allow gaps. Given a word m, we define V_m as the set of genes $v \in V$ for which m is a substring of s_v. Note that wildcard characters match any nucleotide. We test if an exact motif m is enriched in B by applying the hyper-geometric test to the intersection of B and V_m (see Appendix A). In other words, the score of m equals $hg(|V|, |V_m|, |B|, |V_m \cap B|)$. To search for motifs with significant scores we can exhaust all k-mers of specific lengths and gap structures. Algorithmically, this can be performed efficiently using a hash table h in which the keys are motifs and the values are bit-vectors over the gene set V. We iterate on each s_v and for all j mark $h[s_v[j \ldots j+k]][v] = 1$. After processing the promoters of all genes, we can efficiently intersect $h[m]$ with B and compute the score. To guarantee significant results, we must correct for multiple testing by the Bonferroni factor or using FDR. The good practice of motif finding, however, suggests that only motifs with very significant scores will be considered.

Scoring PWMs. PWMs (Position Weight Matrices, defined in Chapter 5) generalize simple combinatorial motifs by introducing a probability distribution over k-mers and defining it by means of independent contributions from each motif position. We define the PWM matching probability as $P(s_1 \dots s_k) = \prod_{0 \leq i < k} w_i[s_i]$ where the w_i are probability distributions over the four nucleotides and k is the PWM dimension. Note that a combinatorial motif can be expressed as a PWM in which each position specifies one of the nucleotides with high probability and the others with low probability. We define the affinity of a PWM to a regulatory region by integrating (in the max or sum norms) the matching probabilities across all possible sites in the sequence. $P(s_i) = \sum_h (P(s[h] \dots s[h+k])$ or $\max_h P(s[h] \dots s[h+k])$. Given an affinity threshold T we define the set of genes $V_P^T = \{v \in V | P(s_v) > T\}$. To score the enrichment of the PWM in the gene module B we optimize the threshold to maximize the combinatorial hypergeometric score $score(P) = max_T hg(|V|, |V_P^T|, |B|, |V_P^T \cap B|)$.

Optimizing PWMs. Given a initial PWM (built from a single k-mer or taken from the literature) we may optimize its enrichment score using the following iterative algorithm. The algorithm is analogous to the one described in Figure 5.6, so we shall only briefly outline it here. We are using two alternating phases (similar to an EM algorithm but with no guarantees for convergence). First, given the current PWM, the optimal affinity threshold is computed. This is done by sorting all genes according to their current PWM affinity and testing all possible affinity thresholds. This part is done (assuming hg is computed in $O(1)$) in $O(\sum_i |s_i|)$ for computing the affinities, $O(nlogn)$ time for sorting and $O(n)$ time for finding the optimal threshold by scanning the sorted list and updating the intersection size incrementally. Second, the PWM is re-estimated as the combination of all sequences with matching likelihood exceeding the threshold (see Figure 5.6 for a more formal description. The algorithm, while lacking formal performance guarantees, does well in practice. We terminate it after the first iteration in which the score does not increase. One major disadvantage of PWMs over combinatorial motifs is the potential over-fitting resulting from the additional degrees of freedom in the model. Our implementation tries to control for this by eliminating PWM positions with low entropy. After each re-estimation iteration, we compute for each position i the entropy

$\sum_{c\in\{A,C,G,T\}} w_i[c]log(w_i[c])$. If the entropy is higher than a threshold (we usually use 1.05), we change the weights to reflect the background nucleotide distribution. In other words - only positions with significant information content can be part of the PWM model. An additional parameter that the algorithm can optimize is the strand preference of the motif. In each iteration, the algorithm computes enrichment scores and affinity thresholds when matching only the 5' strand and when matching both strands. The alternative that scores higher is the one used when re-estimating the model for the next iteration and when reporting the final results.

A two-phase motif finding algorithm. To discover PWMs with optimal scores we are using a two phase approach. In the first phase, we search for k-mers with significant scores as described above. In the second phase we build PWMs by initializing the PWM optimization algorithm with seeds that are built from the highest scoring k-mers (with a weak prior). The algorithm tries optimizing each of the signifcantly scoring k-mers, ordered by their enrichment score. Whenever we find a statistically significant PWM, we mask all its hits (subsequences with matching probability above the affinity threshold) from further consideration by the next PWMs. In this way we avoid the identification of redundant motifs, as a typical PWM generalizes many high scoring k-mers.

Bibliography

[1] The chipping forecast II. Special supplement to Nature Genetics Vol 32, 2002.

[2] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson. *Molecular Biology of the Cell*. Garlend Publishing Inc., New York and London, 1994.

[3] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402, 1997.

[4] M. L. Angus-Hill, A. Schlichter, D. Roberts, H. Erdjument-Bromage, P. Tempst, and B. R. Cairns. A rsc3/rsc30 zinc cluster dimer reveals novel roles for the chromatin remodeler rsc in gene expression and cell cycle control. *Mol Cell*, 7(4):741–51, 2001.

[5] Z. Bar-Joseph, G. K. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon. Continuous representations of time-series gene expression data. *J Comput Biol*, 10(3-4):341–56, 2003.

[6] Z. Bar-Joseph, G.K. Gerber, T.I. Lee, N.J. Rinaldi, J.Y. Yoo, F. Robert, D.B. Gordon, E. Fraenkel, T.S. Jaakkola, R.A. Young, and D.K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21:1337–1342, 2003.

[7] Y Barash, G Elidan, T Kaplan, and N Friedman. Modeling dependencies in protein-dna binding sites. In *Seventh Annual Inter. Conf. on Computational Molecular Biology (RECOMB)*, pages 28–37, 2003.

[8] D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–97, 2004.

[9] M. A. Beer and S. Tavazoie. Predicting gene expression from sequence. *Cell*, 117(2):185–98, 2004.

[10] G. Bejerano, M. Pheasant, I. Makunin, S. Stephen, J.S. Kent, W.J.and Mattick, and D. Haussler. Ultraconserved elements in the human genome. *Science*, 304(5675):1321–5, 2004.

[11] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[12] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. B*, 57:289–300, 1995.

[13] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys Rev E Stat Nonlin Soft Matter Phys*, 67(3 Pt 1):031902, 2003.

[14] S. Bergmann, J. Ihmels, and N. Barkai. Similarities and differences in genome-wide expression data of six organisms. *PLoS Biol*, 2(1):E9, 2004.

[15] K. Biswas, K. J. Rieger, and J. Morschhauser. Functional analysis of carap1, encoding the repressor/activator protein 1 of candida albicans. *Gene*, 307:151–8, 2003.

[16] H. J. Bussemaker, H. Li, and E. D. Siggia. Regulatory element detection using correlation with expression. *Nature Genetics*, 27(2):167–71, 2001.

[17] M. J. Carrozza, S. John, A. K. Sil, J. E. Hopper, and J. L. Workman. Gal80 confers specificity on hat complex interactions with activators. *J Biol Chem*, 277(27):24648–52, 2002.

[18] H. C. Causton, B. Ren, S. S. Koh, C. T. Harbison, E. Kanin, E. G. Jennings, T. I. Lee, H. L. True, E. S. Lander, and R. A. Young. Remodeling of yeast genome expression in response to environmental changes. *Mol Biol Cell*, 12(2):323–37, 2001.

[19] J. Cheng, P. Kapranov, J. Drenkow, S. Dike, S. Brubaker, S. Patel, J. Long, D. Stern, H. Tammana, G. Helt, et al. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science*, 308(5725):1149–54, 2005.

[20] Y. Cheng and G.M. Church. Biclustering of expression data. In *Proc. ISMB'00*, pages 93–103. AAAI Press, 2000.

[21] D. Y. Chiang, P. O. Brown, and M. B. Eisen. Visualizing associations between genome sequences and gene expression data using genome-mean expression profiles. *Bioinformatics*, 17(Suppl 1):S49–55, 2001.

[22] K. R. Christie, S. Weng, R. Balakrishnan, M. C. Costanzo, K. Dolinski, S. S. Dwight, S. R. Engel, B. Feierbach, D. G. Fisk, J. E. Hirschman, et al. Saccharomyces genome database (sgd) provides tools to identify and analyze sequences from *saccharomyces cerevisiae* and related sequences from other organisms. *Nucleic Acids Res*, 32 Database issue:D311–4, 2004.

[23] P. Cliften, P. Sudarsanam, A. Desikan, L. Fulton, B. Fulton, J. Majors, R. Waterston, B. A. Cohen, and M. Johnston. Finding functional features in *saccharomyces* genomes by phylogenetic footprinting. *Science*, 301(5629):71–6, 2003.

[24] J. Cohen. Mathematics is biology's next microscope, only better; biology is mathematics' next physics, only better. *PloS biology*, 2(12):e439, 2004.

[25] M. Cohn, M. J. McEachern, and E. H. Blackburn. Telomeric sequence diversity within the genus saccharomyces. *Curr Genet*, 33(2):83–91, 1998.

[26] A. Colman-Lerner, T. E. Chin, and R. Brent. Yeast cbk1 and mob2 activate daughter-specific genetic programs to induce asymmetric cell fates. *Cell*, 107(6):739–50, 2001.

[27] J.N. Cowles, C.R.and Hirschhorn, D. Altshuler, and E.S. Lander. Detection of regulatory variation in mouse genes. *Nat. Genet.*, 32:432–437, 2002.

[28] J. Deckert and K. Struhl. Histone acetylation at promoters is differentially affected by specific activators and repressors. *Mol Cell Biol*, 21(8):2726–35, 2001.

[29] M.H. DeGroot. *Probability and Statistics*. Addison-Wesley, 1989.

[30] M. Dequard-Chablat, M. Riva, C. Carles, and A. Sentenac. Rpc19, the gene for a subunit common to yeast rna polymerases a (i) and c (iii). *J Biol Chem*, 266(23):15300–7, 1991.

[31] J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–6, 1997.

[32] F. S. Dietrich, S. Voegeli, S. Brachat, A. Lerch, K. Gates, S. Steiner, C. Mohr, R. Pohlmann, P. Luedi, S. Choi, et al. The ashbya gossypii genome as a tool for mapping the ancient *saccharomyces cerevisiae* genome. *Science*, 304(5668):304–7, 2004.

[33] M. T. Doolin, A. L. Johnson, L. H. Johnston, and G. Butler. Overlapping and distinct roles of the duplicated yeast transcription factors ace2p and swi5p. *Mol Microbiol*, 40(2):422–32, 2001.

[34] A. M. Dudley, D. . Janse, A. Tanay, R. Shamir, and G. M. Church. A global view of pleiotropy and phenotypically derived gene function in yeast. *Mol Syst Biol*, page doi:10.1038/msb4100004, 2005.

[35] B. Dujon, D. Sherman, G. Fischer, P. Durrens, S. Casaregola, I. Lafontaine, J. De Montigny, C. Marck, C. Neuveglise, E. Talla, and othres. Genome evolution in yeasts. *Nature*, 430(6995):35–44, 2004.

[36] R. Durbin, S. Eddy, A. Krough, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[37] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–8, 1998.

[38] R. Elkon, C. Linhart, R. Sharan, R. Shamir, and Y. Shiloh. Genome-wide in silico identification of transcriptional regulators controlling the cell cycle in human cells. *Genome Res*, 13(5):773–80, 2003.

[39] B. Enjalbert, A. Nantel, and M. Whiteway. Stress-induced gene expression in candida albicans: absence of a general stress response. *Mol Biol Cell*, 14(4):1460–7, 2003.

[40] S. Even. *Graph Algorithms*. Computer Science Press, Potomac, Maryland, 1979.

[41] M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34:596–615, 1987.

[42] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–20, 2000.

[43] J. E. Galagan, S. E. Calvo, K. A. Borkovich, E. U. Selker, N. D. Read, D. Jaffe, W. FitzHugh, L. J. Ma, S. Smirnov, S. Purcell, B. Rehman, et al. The genome sequence of the filamentous fungus neurospora crassa. *Nature*, 422(6934):859–68, 2003.

[44] A. P. Gasch, A. M. Moses, D. Y. Chiang, H. B. Fraser, M. Berardini, and M. B. Eisen. Conservation and evolution of cis-regulatory systems in ascomycete fungi. *PLoS Biol*, 2(12):e398, 2004.

[45] A. P. Gasch, P. T. Spellman, C. M. Kao, O. 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(12):4241–57, 2000.

[46] I. Gat-Viks and R. Shamir. Chain functions and scoring functions in genetic networks. *Bioinformatics*, 19(Suppl 1):i108–17, 2003.

[47] I. Gat-Viks, A. Tanay, D. Raijman, and R. Shamir. The factor graph network model for biological systems. In *Ninth Annual Inter. Conf. on Computational Molecular Biology (RECOMB), 2005*, 2005.

[48] I. Gat-Viks, A. Tanay, and R. Shamir. Modeling and analysis of heterogeneous regulation in biological networks. *J Comput Biol*, 11(6):1034–49, 2004.

[49] J. V. Geisberg, F. C. Holstege, R. A. Young, and K. Struhl. Yeast nc2 associates with the rna polymerase ii preinitiation complex and selectively affects transcription in vivo. *Mol Cell Biol*, 21(8):2736–42, 2001.

[50] Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[51] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc Natl Acad Sci U S A*, 97(22):12079–84, 2000.

[52] G. Getz, E. Levine, E. Domany, and M. Q. Zhang. Super-paramagnetic clustering of yeast gene expression profile. *Physica A*, 279:457–64, 2000.

[53] S. Ghaemmaghami, W. K. Huh, K. Bower, R. W. Howson, A. Belle, N. Dephoure, E. K. O'Shea, and J. S. Weissman. Global analysis of protein expression in yeast. *Nature*, 425(6959):737–41, 2003.

[54] G. Giaever, A.M. Chu, L. Ni, C. Connelly, L. Riles, S. Veronneau, S. Dow, A. Lucau-Danila, K. Anderson, B. Andre, A.P. Arkin, A Astromoff, et al. Functional profiling of the *saccharomyces cerevisiae* genome. *Nature*, 418:387–91, 2002.

[55] R.A. Gibbs, G.M. Weinstock, M.L. Metzker, D.M. Muzny, E.J. Sodergren, S. Scherer, G. Scott, D. Steffen, K.C. Worley, P.E. Burch, et al. Genome sequence of the brown norway rat yields insights into mammalian evolution. *Nature*, 428(6982):493–521, 2004.

[56] L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, et al. A protein interaction map of drosophila melanogaster. *Science*, 302(5651):1727–36, 2003.

[57] A. S. Goehring, D. A. Mitchell, A. H. Tong, M. E. Keniry, C. Boone, and Jr. Sprague, G. F. Synthetic lethal analysis implicates ste20p, a p21-activated potein kinase, in polarisome activation. *Mol Biol Cell*, 14(4):1501–16, 2003.

[58] I. R. Graham, R. A. Haw, K. G. Spink, K. A. Halden, and A. Chambers. In vivo analysis of functional regions within yeast rap1p. *Mol Cell Biol*, 19(11):7481–90, 1999.

[59] 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, et al. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.

[60] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Combining location and expression data for principled discovery of genetic regulatory networks. In *Pac Symp Biocomput*, pages 437–449, 2002.

[61] J.A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, 1975.

[62] J. L. Hartman, B. Garvik, and L. Hartwell. Principles for the buffering of genetic variation. *Science*, 291(5506):1001–4, 2001.

[63] J. Hastad. Clique is hard to approximate within n1-. In *FOCS '96: Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, page 627, Washington, DC, USA, 1996. IEEE Computer Society.

[64] Y. Ho, A. Gruhler, A. Heilbut, G.D. Bader, L. Moore, S.L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, et al. Systematic identification of protein complexes in *saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415:180–3, 2002.

[65] T. Hofken and E. Schiebel. A role for cell polarity proteins in mitotic exit. *Embo J*, 21(18):4851–62, 2002.

[66] S. Hohmann. Osmotic stress signaling and osmoadaptation in yeasts. *Microbiol Mol Biol Rev*, 66(2):300–72, 2002.

[67] 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*. *J Mol Biol*, 296(5):1205–14, 2000.

[68] T. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292(5518):929–34, 2001.

[69] T.E. Ideker, V. Thorsson, and R.M. Karp. Discovery of regulatory interaction through perturbation: inference and experimental design. In *Proceedings of the 2000 Pacific Symposioum in Biocomputing (PSB 00)*, pages 305–316, 2000.

[70] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nat Genet*, 31(4):370–7, 2002.

[71] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *J. Bioinform. Comput. Biol.*, 2:77–98, 2004.

[72] V. R. Iyer, C. E. Horak, C. S. Scafe, D. Botstein, M. Snyder, and P. O. Brown. Genomic binding sites of the yeast cell-cycle transcription factors sbf and mbf. *Nature*, 409(6819):533–8, 2001.

[73] F. Jacob and J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, 3:318–356, 1961.

[74] T. Jones, N. A. Federspiel, H. Chibana, J. Dungan, S. Kalman, B. B. Magee, G. Newport, Y. R. Thorstenson, N. Agabian, P. T. Magee, R. W. Davis, and S. Scherer. The diploid genome sequence of candida albicans. *Proc Natl Acad Sci U S A*, 101(19):7329–34, 2004.

[75] S. Kalir, J. McClure, K. Pabbaraju, C. Southward, M. Ronen, S. Leibler, M.G. Surette, and U. Alon. Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria. *Science*, 292(5524):2080–3, 2001.

[76] J. Kanoh and F. Ishikawa. sprap1 and sprif1, recruited to telomeres by taz1, are essential for telomere function in fission yeast. *Curr Biol*, 11(20):1624–30, 2001.

[77] S. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, 1993.

[78] M. Kellis, B. W. Birren, and E. S. Lander. Proof and evolutionary analysis of ancient genome duplication in the yeast *saccharomyces cerevisiae*. *Nature*, 428(6983):617–24, 2004.

[79] 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, 2003.

[80] M. Kimura. Evolutionary rate at the molecular level. *Nature*, 217:624–626, 1968.

[81] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16:111–120, 1980.

[82] M. Kirschner and J. Gerhart. Evolvability. *Proc Natl Acad Sci U S A*, 95(15):8420–7, 1998.

[83] Y. Kluger, R. Basri, JT. Cheng, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.*, 13(4):703–16, 2003.

[84] C. P. Kurtzman and C. J. Robnett. Phylogenetic relationships among yeasts of the Saccharomyces complex' determined from multigene sequence analyses. *FEMS Yeast Res*, 3(4):417–32, 2003.

[85] ES. Lander et al. Initial sequencing and analysis of the human genome. international human genome sequencing consortium. *Nature*, 409:860–921, 2001.

[86] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002.

[87] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in *saccharomyces cerevisiae*. *Science*, 298(5594):799–804, 2002.

[88] B. Li, S. Oestreich, and T. de Lange. Identification of human rap1: implications for telomere evolution. *Cell*, 101(5):471–83, 2000.

[89] W. H. Li and D. Graur. *Fundamentals of Molecular Evolution.* Sinauer, Sunderland, MA, 1991.

[90] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Proc. Pacific Symposium on Biocomputing*, pages 18–29. World Scientific, 1998.

[91] X. Liu and N.D. Clarke. Rationalization of gene regulation by a eukaryotic transcription factor: calculation of regulatory region occupancy from predicted binding affinities. *J Mol Biol.*, 323(1):1–8, 2002.

[92] M.C. Lopez and H.V. Baker. Understanding the growth phenotype of the yeast gcr1 mutant in terms of global genomic expression patterns. *J Bacteriol*, 182(17):4970–8, 2002.

[93] M. Z. Ludwig, C. Bergman, N. H. Patel, and M. Kreitman. Evidence for stabilizing selection in a eukaryotic enhancer element. *Nature*, 403(6769):564–7, 2000.

[94] M. Z. Ludwig, N. H. Patel, and M. Kreitman. Functional analysis of eve stripe 2 enhancer evolution in drosophila: rules governing conservation and change. *Development*, 125(5):949–58, 1998.

[95] R. M. Marion, A. Regev, E. Segal, Y. Barash, D. Koller, N. Friedman, and E. K. O'Shea. Inaugural article: Sfp1 is a stress- and nutrient-sensitive regulator of ribosomal protein gene expression. *Proc Natl Acad Sci U S A*, 101(40):14315–22, 2004.

[96] M. Markstein, P. Markstein, V. Markstein, and M. S. Levine. Genome-wide analysis of clustered dorsal binding sites identifies putative target genes in the drosophila embryo. *Proc Natl Acad Sci U S A*, 99(2):763–8, 2002.

[97] D. E. Martin, A. Soulard, and M. N. Hall. Tor regulates ribosomal protein gene expression via pka and the forkhead transcription factor fhl1. *Cell*, 119(7):969–79, 2004.

[98] S. A. McCarroll, C. T. Murphy, S. Zou, S. D. Pletcher, C. S. Chin, Y. N. Jan, C. Kenyon, C. I. Bargmann, and H. Li. Comparing genomic expression patterns across species identifies shared transcriptional profile in aging. *Nat Genet*, 36(2):197–204, 2004.

[99] I. Nachman, A. Regev, and N. Friedman. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20:248–256, 2004.

[100] K. Natarajan, M. R. Meyer, B. M. Jackson, D. Slade, C. Roberts, A. G. Hinnebusch, and M. J. Marton. Transcriptional profiling shows that gcn4p is a master regulator of gene expression during amino acid starvation in yeast. *Mol Cell Biol*, 21(13):4347–68, 2001.

[101] S. M. O'Rourke and I. Herskowitz. Unique and redundant roles for hog mapk pathway components as revealed by whole-genome expression analysis. *Mol Biol Cell*, 15(2):532–42, 2004.

[102] J. Pearl. *Probabilistic Reasoning in intelligent systems.* Morgan Kaufmann publishers, inc, 1988.

[103] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17 Suppl 1:S215–24, 2001.

[104] D. Pe'er, A. Regev, and A. Tanay. A fast and robust method to infer and characterize an active regulator set for molecular pathways. *Bioinformatics*, 18(Suppl 1 (Proc. ISMB 2002)):S258–67, 2002.

[105] Y. Pilpel, P. Sudarsanam, and GM. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, 29(2):153–9, 2001.

[106] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

[107] M. Primig, R.M. Williams, E.A. Winzeler, G.G. Tevzadze, A.R. Conway, S.Y. Hwang, R.W. Davis, and R.E. Esposito. The core meiotic transcriptome in budding yeasts. *Nature Genetics*, 26(4):415–23, 2000.

[108] M. Pritsker, Y. C. Liu, M. A. Beer, and S. Tavazoie. Whole-genome discovery of transcription factor binding sites by network-level conservation. *Genome Res*, 14(1):99–108, 2004.

[109] S. Ramaswamy et al. Multiclass cancer diagnosis using tumor gene expression signature. *Proc. Natl. Acad. Sci. USA*, 98(26):15149–15154, 2001.

[110] E. Ravasz, A.L. Somera, A.D. Mongru, Z.N. Oltvai, and A.L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–5, 2002.

[111] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young. Genome-wide location and function of dna binding proteins. *Science*, 290(5500):2306–9, 2000.

[112] M. Rep, M. Krantz, J.M. Thevelein, and S. Hohmann. The transcriptional response of *saccharomyces cerevisiae* to osmotic shock. hot1p and msn2p/msn4p are required for the induction of subsets of high osmolarity glycerol pathway-dependent genes. *J. Biol. Chem*, 275:8290–8300, 2000.

[113] M. Rep, M. Proft, F. Remize, M. Tamas, R. Serrano, J.M. Thevelein, and Hohmann. S. The *saccharomyces cerevisiae* sko1p transcription factor mediates hog pathway-dependent osmotic regulation of a set of genes encoding enzymes implicated in protection from oxidative damage. *Mol. Microbiol.*, 40:1067–1083, 2001.

[114] T. Rozovskaia, O. Ravid-Amir, S. Tillib, G. Getz, E. Feinstein, H. Agrawal, A. Nagler, E.F. Rappaport, I. Issaeva, Y. Matsuo, U.R. Kees, T. Lapidot, F. Lo Coco, R. Foa, A. Mazo, T. Nakamura, CM. Croce, G. Cimino, E. Domany, and E. Canaani. Expression profiles of acute lymphoblastic and myeloblastic leukemias with all-1 rearrangements. *Proc Natl Acad Sci U S A*, 100(13):7853–8, 2003.

[115] S. B. Schawalder, M. Kabani, I. Howald, U. Choudhury, M. Werner, and D. Shore. Growth-regulated recruitment of the essential yeast ribosomal protein gene activator ifh1. *Nature*, 432(7020):1058–61, 2004.

[116] M. Schena, D. Shalon, R. Heller, A. Chai, P. O. Brown, and R. W. Davis. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci. U.S.A.*, 93:10614–9, 1996.

[117] H. J. Schuller. Transcriptional control of nonfermentative metabolism in the yeast *saccharomyces cerevisiae*. *Curr Genet*, 43(3):139–60, 2003.

[118] 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. *Nat Genet*, 34(2):166–76, 2003.

[119] E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional modules from dna sequence and gene expression. *Bioinformatics*, 19 Suppl 1:i273–82, 2003.

[120] R. Sharan, A. Maron-Katz, and R. Shamir. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics*, 19(14):1787–99, 2003.

[121] R. Sharan and R. Shamir. CLICK: A clustering algorithm for gene expression analysis. In S. Miyano, R. Shamir, and T. Takagi, editors, *Currents in Compuational Molecular Biology*, pages 6–7. Universal Academy Press, 2000.

[122] I. Simon, J. Barnett, N. Hannett, C. T. Harbison, N. J. Rinaldi, T. L. Volkert, J. J. Wyrick, J. Zeitlinger, D. K. Gifford, T. S. Jaakkola, and R. A. Young. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106(6):697–708, 2001.

[123] P. T. Spellman, G. Sherlock, et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

[124] L. D. Stein, Z. Bao, D. Blasiar, T. Blumenthal, M.R. Brent, N. Chen, A. Chinwalla, L. Clarke, C. Clee, A. Coghlan, et al. The genome sequence of Caenorhabditis briggsae: a platform for comparative genomics. *PLoS Biol*, 1(2):E45, 2003.

[125] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–55, 2003.

[126] P. Sudarsanam, V. R. Iyer, P. O. Brown, and F. Winston. Whole-genome expression analysis of snf/swi mutants of *saccharomyces cerevisiae*. *Proc Natl Acad Sci U S A*, 97(7):3364–9, 2000.

[127] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T.R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *PNAS*, 96:2907–2912, 1999.

[128] A. Tanay, I. Gat-Viks, and R. Shamir. A global view of the selection forces in the evolution of yeast cis-regulation. *Genome Res*, 14(5):829–34, 2004.

[129] A. Tanay, A. Regev, and R. Shamir. Conservation and evolvability in regulatory networks: the evolution of ribosomal regulation in yeast. *Proc Natl Acad Sci U S A*, 102(20):7203–8, 2005.

[130] A. Tanay and R. Shamir. Computational expansion of genetic networks. *Bioinformatics*, 17 Suppl 1:S270–8, 2001.

[131] A. Tanay and R. Shamir. Modeling transcription programs: Inferring binding site activity and dose - response model optimization. *Seventh Annual Interna-*

tional Conference on Computational Molecular Biology (RECOMB 03), pages 301–310, 2003.

[132] A. Tanay and R. Shamir. Multilevel modeling and inference of transcription regulation. *J Comput Biol*, 11(2-3):357–75, 2004.

[133] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci U S A*, 101(9):2981–6, 2004.

[134] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1 (Proc. ISMB 2002):S136–44, 2002.

[135] A. Tanay, R. Sharan, and R. Shamir. Algorithms for biclustering gene expression data. In S. Aluru, editor, *Handbook in bioinformatics.* Kluwer, New York, 2005.

[136] A. Tanay, I. Steinfeld, M. Kupiec, and R. Shamir. Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium. *Mol Sys Biol*, page doi: 10.1038/msb4100005, 2005.

[137] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[138] M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol*, 23(1):137–44, 2005.

[139] A. H. Tong, G. Lesage, G. D. Bader, H. Ding, H. Xu, X. Xin, J. Young, G. F. Berriz, R. L. Brost, M. Chang, et al. Global mapping of the yeast genetic interaction network. *Science*, 303(5659):808–13, 2004.

[140] O.G. Troyanskaya, K. Dolinski, A.B. Owen, R.B. Altman, and D. Botstein. Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc. Natl. Acad. Sci. USA*, 100:8348–53, 2003.

[141] H. Uemura, M. Watanabe-Yoshida, N. Ishii, T. Shinzato, R. Haw, and Y. Aoki. Isolation and characterization of candida albicans homologue of rap1, a repressor and activator protein gene in *saccharomyces cerevisiae*. *Yeast*, 21(1):1–10, 2004.

[142] P. Uetz et al. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403:623–7, 2000.

[143] G. Velours, C. Boucheron, S. Manon, and N. Camougrand. Dual cell wall/mitochondria localization of the 'sun' family proteins. *FEMS Microbiol Lett*, 207(2):165–72, 2002.

[144] J.C. Venter et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001.

[145] J. T. Wade, D. B. Hall, and K. Struhl. The transcription factor ifh1 is a key regulator of yeast ribosomal protein genes. *Nature*, 432(7020):1054–8, 2004.

[146] J. Wahlin and M. Cohn. Analysis of the rap1 protein binding to homogeneous telomeric repeats in *saccharomyces castellii*. *Yeast*, 19(3):241–56, 2002.

[147] K.L. Wang and J.R. Warner. Positive and negative autoregulation of reb1 transcription in *saccharomyces cerevisiae*. *Mol Cell Biol.*, 18(7):4368–76, 1998.

[148] W. Wang, J. M. Cherry, D. Botstein, and H. Li. A systematic approach to reconstructing transcription networks in *saccharomyces cerevisiae*. *Proc Natl Acad Sci U S A*, 99(26):16893–8, 2002.

[149] E. Wingender, X. Chen, E. Fricke, R. Geffers, R. Hehl, I. Liebich, M. Krull, V. Matys, H. Michael, R. Ohnhauser, M. Pruss, F. Schacherer, S. Thiele, and S. Urbach. The TRANSFAC system on gene expression regulation. *Nucleic Acids Res.*, 29:281–3, 2001.

[150] I. Witt, M. Kwart, T. Gross, and N. F. Kaufer. The tandem repeat agggtagggt is, in the fission yeast, a proximal activation sequence and activates basal transcription mediated by the sequence tgtgactg. *Nucleic Acids Res*, 23(21):4296–302, 1995.

[151] V. Wood, R. Gwilliam, M. A. Rajandream, M. Lyne, R. Lyne, A. Stewart, J. Sgouros, N. Peat, J. Hayles, S. Baker, et al. The genome sequence of *schizosaccharomyces pombe*. *Nature*, 415(6874):871–80, 2002.

[152] G. A. Wray, M. W. Hahn, E. Abouheif, J. P. Balhoff, M. Pizer, M. V. Rockman, and L. A. Romano. The evolution of transcriptional regulation in eukaryotes. *Mol Biol Evol*, 20(9):1377–419, 2003.

[153] L.F. Wu, T.R. Hughes, A.P. Davierwala, M.D. Robinson, R Stoughton, and S.J. Altschuler. Large-scale prediction of *saccharomyces cerevisiae* gene function using overlapping transcriptional clusters. *Nature Genetics*, 31(3):255–65, 2002.

[154] Davidson EH. Yuh CH, Bolouri H. Cis-regulatory logic in the endo16 gene: switching from a specification to a differentiation mode of control. *Development*, 128(5):617–29, 2001.

[155] G. Zhu, P. T. Spellman, T. Volpe, P. O. Brown, D. Botstein, T. N. Davis, and B. Futcher. Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature*, 406(6791):90–4, 2000.