Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tomaž Curk

Računski pristopi k odkrivanju genskih mrež

DOKTORSKA DISERTACIJA

Ljubljana, 2007

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tomaž Curk

Računski pristopi k odkrivanju genskih mrež

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Blaž Zupan

Ljubljana, 2007

University of Ljubljana
Faculty of Computer and Information Science

Tomaž Curk

Computational approaches for gene network discovery

DOCTORAL DISSERTATION

Supervisor: Prof. Dr. Blaž Zupan

Ljubljana, 2007

Abstract

This dissertation proposes a set of computational methods for inference of gene networks from heterogeneous data sources. These methods address several important problems in gene network inference, including function prediction from different types of gene profile/phenotype data (*i.e.*, gene expression profiles, mutant transcriptional phenotypes and mutant sensitivity profiles), methods for the analysis of gene regulatory regions, and methods for decomposition of gene expression signature profiles.

The central part of the dissertation and its main contribution is a method that can combine sequence and phenotype information to find clusters of genes with similar phenotype and sequence. The method relies on a new machine learning approach called rule-based clustering. The approach is able to discover clusters – groups of data items, or genes – that are described using a symbolic assertion on the gene sequence (the conditional part), with genes in the group bearing similar phenotypes (the action part of the rule). While designed with bioinformatics problems in mind, the proposed rule-based clustering can be regarded as a general machine learning technique. It requires two sets of attributes: one is used to calculate distance among examples, while the other set is used for construction of symbolic descriptions of discovered clusters. Our inference algorithm uses beam-search heuristics and statistical tests for rule selection and stopping criteria. Rule-based clustering can discover overlapping clusters. Rule-based relations between data items (genes) can be regarded as networks. We propose a set of visualizations to aid in the presentation and interpretation of inferred rules.

We experimentally evaluated and successfully applied the proposed methods to infer patterns of gene regulation in slime mold *Dictyostelium discoideum* and budding yeast *Saccharomyces cerevisiae* from DNA microarray and regulatory region DNA sequence data. We show how rule-based clustering can help to answer some very important biological questions about the regulation of gene expression: what are the most informative features in the regulatory region and where in the DNA sequence do they reside. Experiments with different kinds of genomic phenotypes (DNA microarray, mutant transcriptional phenotype and mutant sensitivity profiles) show that each better predicts different aspects of gene annotation.

We also propose a computational method for the decomposition of gene

expression profile signature into a combination of known mutant expression profiles or profiles from different treatments or both. Again, the resulting model is a network of genes (mutants) and conditions (treatments) that complements the analysis done with rule-based clustering and can be used to infer pathways and functions in which genes are involved.

Keywords

machine learning
bioinformatics
visualization
rule-based clustering
gene networks
functional genomics
modeling regulation of gene expression
decomposition of gene expression profile signatures

Acknowledgements

My sincere gratitude goes to my mentor, Blaž Zupan, who introduced me to the field of machine learning and bioinformatics, and guided me through many interesting problems in the field. I thank Ivan Bratko, the head of Artificial Intelligence Laboratory, for all the advice and for providing a great working environment. All my research work in these past years was supported by a junior research grant awarded by the Slovenian Research Agency.

I appreciated and enjoyed collaborating with the laboratory of Gad Shaulsky, at Baylor College of Medicine in Houston, Texas. The enthusiasm about new scientific questions, the richness of data and knowledge they provided greatly motivated and shaped my work. I thank the members of Shaulsky's lab that I had the privilege to work with: Mariko Katoh, Nancy Van Driessche, Anup Parikh, Rocio Benabentos, Qikai Xu, Ezgi (Okyay) Booth, Eryong Huang, Chad Shaw, Anupama Khare, Edward Miranda, and Lorenzo Santorelli. Here I would also like to thank Adam Kuspa and Ricky Sucgang from Baylor for the help on *D. discoideum* sequence data, and Rex Chisholm and Eric Just from Northwestern University, for their help with dictyBase.

Uroš Petrovič represents the Slovene counterpart of Shaulsky's lab. He provided many good ideas, knowledge and data on *S. cerevisiae* that motivated our work. I thank Mojca Mattiazzi for her laborious experimental work that resulted in good quality data.

I thank all the members of the AI lab for providing a stimulative environment. Especially, I would like to mention Janez Demšar and Gregor Leban for their work on Orange and discussions on various research topics, Minca Mramor, Peter Juvan, and Lan Umek for their work and discussion on interesting bioinformatics problems. Aleks Jakulin directed me to a large number of interesting papers in the field of machine learning and bioinformatics. I thank Riccardo Bellazzi and Lucia Sacchi for collaborating on interesting problems, and for welcoming me during the work visits at the Università degli Studi di Pavia.

I am greatly thankful to Patricia and Bruce Davis for welcoming me in their home, first as an exchange student in high school, and then in all my graduate study years when visiting and working at Baylor College of Medicine in Houston. Above all, I am thankful to my family: Janja, Vojko, Nataša, Aleš, Vida, Iztok.

Tomaž Curk

Contents

Chapter 1

Introduction

Indeed, we believe computer science is poised to become as fundamental to biology as mathematics has become to physics.

The 2020 Science Group, July 2005

Computer science is already having a profound impact and it is playing a crucial role in sciences investigating complex systems such as biology, chemistry, physics, biotechnology and biomedicine. Recently it has been argued that the development of new conceptual and technological tools is required to further advance scientific computing and take the leap from the application of computing to support scientists to 'do' science (*i.e.*, computational science) to the integration of computer science concepts, tools and theorems into the very fabric of science [The 2020 Science Group, 2006; Foster, 2006]. Nonetheless, allowing and supporting the scientists to discover, 'look at' and easily interact with patterns in heterogeneous data remains a key enabler of research and discovery in data-rich scientific domains for many years to come.

This dissertation focuses on the development and application of computational methods, especially those from the field of machine learning and bioinformatics, for the analysis of heterogeneous genomic data. For over a decade, modern high throughput technology in genomics allows biological experimentalists an almost real-time acquisition of huge amounts of various genomic data. Sequencing the DNA of an organism and identifying all genes were the ini-

tial steps taken to build the "parts" list of an organism [International, 2004; McPherson et al., 2001]. To get an overview of the dynamic part of the cellular processes geneticists then devised various methods to measure the presence and activity of individual genes and their products – proteins [Luscombe et al., 2001]. Analyzing all these heterogeneous data sets represent a great challenge and opportunity to advance our understanding of the underlying cellular processes and as a consequence improve medical treatment procedures [Schlitt and Brazma, 2005, 2006].

With a fast-growing base of genome data sets, new computational methods for analysis and tools for visualization are needed to extract and represent discovered patterns in an intelligible form and thus enable the user to interpret and gain new knowledge from experimental data. A widely used formalism in this area is a gene network that can, in general, represent any inferred relations between genes, their products and other cell components in biological systems. Depending on the level of biological inquiry and available experimental data, there are many different types of gene network models, including transcription regulation networks, metabolic networks, signaling networks, protein interaction networks, *etc.* Although in nature all these networks are profoundly interlinked and understanding each on its own is a prerequisite to understanding the complete biological network [Schlitt and Brazma, 2005], this dissertation focuses on networks for which genome-wide data on sequence and expression is available. These include transcription regulation networks, for which we developed a new inference algorithm called rule-based clustering, gene networks representing functional similarity that are often modeled and assessed with gene co-expression networks, and networks of gene expression response for which we propose a method based on decomposition of gene signature profiles.

The most intuitive way to visualize networks is in form of graphs where nodes and edges bear problem domain-specific symbolic meaning. Methods that discover patterns from experimental data and encode them in transparent and intelligible symbolic models of low complexity (*e.g.*, set of IF-THEN rules) are suitable for human interpretation and can be, when combined with appropriate visualization and software implementations, used for explorative data analysis [Tukey, 1977]. We believe that the computational methods proposed and described here (especially rule-based clustering and decomposition of gene signature profiles) are of this kind.

State-of-the-art experimental techniques enable us to collect various types of genomic data for the same experimental condition. An important question is how a model's ability to correctly predict gene function depends on the choice of the type of genomic data used to infer the predictive model. For example, Winzeler et al. [1999] showed that there is little correlation between gene expression as measured with DNA microarrays and gene function. Consequently, one can expect that predictive models built using DNA microarray data will successfully predict function of only a small portion of genes. We take this consideration further and try to, using computational tools for *in silico* experimentation, systematically evaluate, compare, and determine which types of gene profile data (gene expression under various conditions, mutant transcription phenotypes, and mutant sensitivity profiles) can better predict specific functional annotations. Our particular advantage is that we stem from the machine learning framework, where model testing, evaluation, and statistics to report on the evaluation scores are well developed. Various types of gene profiles are then assessed through how well do modeling methods perform on these profile types in terms of predictive accuracy.

Another important question is how to make best use of huge and ever growing amounts of available genomic data that is being collected in genome-wide studies (*e.g.*, DNA microarray data, mutant sensitivity profiles, *etc.*). We believe that specialized tools, which allow the user to compare and relate data from his experiment to what has already been observed and reported in other experiments, are crucial. The decomposition of gene profile signatures that we propose is a step in this direction.

1.1 Contributions of the dissertation

The main and original contributions of this work are:

- the design and implementation of a rule-based clustering method – a new machine learning clustering method for handling complex and feature rich problem domains,

- a practical application of rule-based clustering in the area of bioinformatics, which also required us to formalize and evaluate the descriptive

(hypothesis) language used to model the structure of gene regulatory regions,

- the design and implementation of a computational method for decomposition of gene profile signatures,
- investigations in computational phenomics, where we have used the devised computational techniques to experimentally evaluate the predictive value of different types of genomic data.

We also propose a set of visualizations to support explorative analysis and interpretation of models inferred with rule-based clustering. The developed methods are applied for the analysis of data on two model organisms of interest: *Saccharomyces cerevisiae* and *Dictyostelium discoideum*.

Practical contributions of the dissertation include the implementation of the rule-based clustering method through Python scripts and implemented within the Orange data mining suite [Demsar et al., 2004a], and the implementation of the decomposition of gene profile signatures method as a publicly available, web-based application (available at http://bubble.fri.uni-lj.si/microCOMB).

1.2 Overview of the dissertation

This dissertation is organized in eight chapters. An overview of computational methods for inference of gene networks and methods for the analysis of regulatory regions is given in Chapter 2. In Chapter 3 we introduce the terms "computational phenotype" and "computational phenomics" and present a systematic, empirical study showing that different types of gene profile data are more suitable for reasoning and inference of predictive models for different aspects of gene functional annotation.

Our rule-based clustering method is described in Chapter 4. The heuristic rule search is presented first, followed by a description of the on-the-fly operator-based feature construction which is then incorporated into the search. A method for the final generalization and post processing of the discovered model (*i.e.*, set of IF-THEN rules) is described next. Space and time complexity of rule-based clustering is investigated and compared to the complexity of the standard CN2 [Clark and Nibblet, 1989] rule induction algorithm. Chapter 4 concludes

with the proposal of an evaluation method for models inferred by rule-based clustering.

In Chapter 5 we propose a set of intuitive visualizations that can be used to gain insight into the discovered clusters of objects (*i.e.*, genes in all our applications), relations between overlapping clusters, and also conditional terms of rules describing the discovered clusters. These visualizations allow the user to explore individual objects, observe how they cluster into higher-order structures, and observe terms in the conditional part of discovered rules (*i.e.*feature from data or newly constructed features) that are common to objects or clusters. We also describe the structure of web-pages that are automatically generated for presentation of rule-based clustering results. In this dissertation, applied to problems in bioinformatics, the pages are linked to already existent tools and databases, such as the Genome Browser [Durbin et al., 2000] and GO Term Finder (http://www.yeastgenome.org), and which foster further exploration of discovered patterns of gene regulation and corresponding gene clusters.

Results of experimental analyses using rule-based clustering and evaluation of inferred models on the data from two model organisms are presented in Chapter 6. In this Chapter, we also investigated the predictive value of various features from gene sequences, constructed during machine learning using constructive operators of different kinds and complexity. We also use rule-based clustering and standard model evaluation procedures from machine learning to computationally determine the most informative sub-interval in the gene regulatory region.

The idea of decomposition of gene expression profiles is introduced in Chapter 7. A heuristic algorithm for decomposition is presented and some operators suitable for decomposition are discussed and evaluated. Examples for the two model organisms are given. We also describe a web-based tool we have implemented for decomposition of gene profile signatures.

Chapter 8 concludes the dissertation with a discussion of the proposed methods and results of our experimental studies, and present several ideas for further work.

Chapter 2

Background in artificial intelligence and bioinformatics

The Thesis is related to research at the intersection of machine learning and bioinformatics. In this Chapter, we present the background on both research areas that are related to our original approach we have developed for machine-learning based inference of gene interaction networks.

2.1 On search and inference of predictive models

A substantial part of the research in this Thesis involves the construction of search algorithms. Search is a fundamental topic in artificial intelligence and in its subfields on machine learning [Mitchell, 1997], data mining, and knowledge discovery [Fayyad et al., 1996b,a]. Search is the main approach used for problem solving, automated reasoning, inference of models, and many other combinatorial processes whose goal is to find an optimal, if not the best solution. For example, in algorithmic problem solving, a general scheme called *state space* is normally applied for representing and solving problems. The state space is most intuitively represented as a graph, where nodes correspond to problem situations, and edges correspond to "legal moves" or transitions between situations. Solving a given problem is then translated to searching of a state space graph and exploring alternatives with the goal of finding the optimal path to the solution. In optimization, the state space is explored to find an optimal state

according to some criteria function. Basic strategies for exploring alternatives include depth-first search, breadth-first search and iterative deepening [Bratko, 2001].

Real–life problem solving may be extremely complex and implementation of automated reasoning and construction of search algorithm often require representation of a problem and data at various levels of abstraction. Even then, the number of alternatives, for example, the size of the problem space, can be extremely high and complexity of the search algorithm becomes most critical. For example, if we represent the search space with a tree, where each internal node has b successors, and all solutions are found by traversing at least d nodes form the root of the tree, then the total number of paths is in the order of b^d or $O(b^d)$. Even with relatively small values of b, the number of candidate paths grows exponentially with their length, and one is faced by the so called phenomenon of combinatorial explosion [Bratko, 2001]. This combinatorial complexity is typical and an unavoidable problem associated with search.

Basic, also called uninformed or greedy search strategies mentioned above are not sufficient for solving large-scale problems. They treat all alternatives in a state-space as equally promising and will explore every one of them. The approach becomes unfeasible when faced with complex combinatorial problems. Search for solutions of such problems must be guided by problem-specific information or heuristic. These kinds of algorithms are thus called heuristic search algorithms. A well-known example of one such classic algorithm uses best-first heuristic principle where the currently most promising node, according to a heuristic estimate, is explored and expanded first [Bratko, 2001].

The task of inferring symbolic models by learning rule-based relational descriptions from a set of examples, which we address in this Thesis, is subject to similar search and combinatorial complexity and thus also requires appropriate heuristics. The goal of learning is to find a kind of generalized theory that explains and is able to recognize all objects forming a concept. A concept is a subset of all possible examples that a learner may encounter when applying the inferred theory to recognize (new) examples, including examples which were not seen in the learning phase. Examples belonging to a concept are called positive examples, all other are negative examples. Most approaches to rule induction, including the well-known CN2 covering algorithm [Clark and Nibblet, 1989] for discovering rules of form IF *condition* THEN *consequence*, start with a simple

rule (normally the predicate "True") that covers all examples in a learning set given by the user. An example is said to be covered by a rule if it matches the description stated in the conditional part of the rule. The initial rule is subsequently refined by systematically changing the conditional part until the best rule is found. Various criteria are used to evaluate rules. The currently best rule is then added to the set of discovered rules forming the final model or hypothesis. Examples covered by the rule are either removed or their weights are decreased, and search for the next best rule is reiterated. This is repeated until all examples are covered or some other stopping criteria are met. Ideally, each rule in the final model has to be complete, *i.e.*, it must cover all positive examples, and it has to be sound, *i.e.*, it must not cover any negative example. In practice, relaxed variants are used that allow covering some negative examples and not covering all positive examples. Rule inference is an inherently combinatorial search problem. The normally vast hypotheses space can be significantly constrained by the hypothesis language (which introduces a linguistic or language bias) which allows the construction and testing of only certain hypotheses. Search is also constrained by the search method that determines what parts of the hypotheses space will be explored. This is called the search bias.

Using problem-specific information, *i.e.*, background knowledge and data, in the automated, computer-aided data analysis and discovery process is referred to as intelligent data analysis. This type of analysis is possible and most desirable in knowledge rich fields where knowledge is encoded in an electronic form and readily shared in open data and knowledge bases. Biology and functional genomics are such fields. Using additional knowledge sources in the analysis can prevent discovering the obvious, it can complement an inferred hypothesis with references to already proposed relations, it can prevent inferring overconfident models and it allows for a systematic comparison of findings to existent knowledge. In this Thesis we heavily rely on background knowledge and data stored in publicly available databases (*e.g.*, TRANSFAC for data on known transcription factors binding sites, www.geneontology.org for gene annotation on molecular function, biological process, and localization within the cell, *etc.*) for the formalization of the descriptive language used to model problems. For these reasons the proposed methods can be classified as methods for intelligent data analysis.

2.2 Related methods for gene network construction

A gene is defined as a stretch of DNA that can be transcribed into RNA by the transcriptional cell machinery. The gene itself is a passive entity. It needs to be transcribed into RNA and eventually translated into proteins to have an active role in cell processes. There are different types of RNAs. For example, microRNA (miRNA), small nuclear RNA (snRNA), small nucleolar RNA (snoRNA), and other noncoding RNAs have a regulatory role in transcription and translation, while the messenger RNA (mRNA) is used as a template for translation into a polypeptide chain which then folds into a protein encoded by the gene [Alberts et al., 1994; Kimball, 2006]. The ribosomal RNAs (rRNA) are used in the building of ribosomes, which are the machinery for synthesizing proteins by translating mRNA. There are different kinds of transfer RNAs (tRNA), each responsible for bringing the correct amino acid (one of the twenty amino acids) into the growing polypeptide chain [Kimball, 2006]. Transcription factors are special kinds of proteins, which can inhibit or excite the expression of other genes by binding to sequence-specific binding sites in their regulatory region [Latchman, 1998]. Usually, a transcription factor can directly influence the expression of only a small number of genes, but those genes subsequently can regulate the expression of other genes. A gene can thus, figuratively speaking, directly or indirectly influence the activity of many other downstream genes, and at the same time its activity can be driven by many genes upstream of it Auto regulation, when a gene regulates its own activity directly or indirectly through regulatory loops, is also found in gene networks. Regulation by binding of transcription factors is just one of the many mechanisms of gene expression regulation. Gene expression is also determined by changes in the cell's environment that can influence gene expression through both transcription factors or through epigenetic effects [Pedersen et al., 1999] (*e.g.*, chromatin structure). Other forms of regulation that affect DNA structure [Segal et al., 2006; Choi et al., 2004], mRNA and protein activity, stability, localization include post-transcriptional, translational, post-translational, and other regulation [Wasserman and Sandelin, 2004].

Genetic (regulatory) network is a widely used formalism to model the regulatory influences between genes. Different formalisms were developed to rep-

resent genetic networks at various levels of abstraction, each type providing answers to different kinds of biological questions. These network formalisms range from simple Boolean networks [Akutsu et al., 1999, 2000; Liang et al., 1998; Wuensche, 1998], generalized Boolean networks [Tanay and Shamir, 2001], directed graph models [Wagner, 2001], qualitative network models [Shrager et al., 2002; Thieffry et al., 1998], complex non-linear differential equations [Wessels et al., 2001; Wahde and Hertz, 2001; De Jong et al., 2004b,a] to probabilistic graphical models [Pournara and Wernisch, 2004; Friedman et al., 2000; Yoo and Cooper, 2002], [Yoo et al., 2002; Pe'er et al., 2001; Hartemink et al., 2001; Toh and Horimoto, 2002], and gene regulation networks and their reconstruction through abductive inference [Van Driessche et al., 2005; Zupan et al., 2003a,b]. For a detailed overview of listed methods see Chapter 2 in Juvan Most of the methods for gene network reengineering, that is, their reconstruction from experimental data, use data on either DNA microarray gene expression, possibly from experiments that measure expression at different time points or under different perturbations of environment or genes. Time series data is used to model the change of gene expression levels and also to model the relations of gene expression among genes. Perturbation data is used to model the relations between perturbed and all other genes.

Inference of Boolean networks, solving differential equations, probabilistic networks and other similar computational methods have been shown to have high predictive accuracy. However, it may be argued that some of these techniques generate and encode models in forms which can be quite challenging to interpret and understand (*e.g.*, set of differential equations). Intelligent data analysis favors machine learning methods which can present the inferred models in an intelligible form that allows the interpretation and study of discovered patterns in data. Methods like the induction of symbolic rules may be better in this respect. While, in general, the inferred models have a high predictive power, *e.g.*, in successfully predicting gene expression levels, they can be substantially different in terms of structure (gene relations) from the real, underlying model. This was shown on synthetic data and it suggests a low inferential power of some numeric methods mentioned above [Wessels et al., 2001].

The DNA hybridization array technology allows to measure simultaneously, in a single experiment, the levels of transcribed RNA for thousands of genes [Friend and Stoughton, 2002]. The raw measurement data has to be adjusted by

various within-array and across-array normalization steps [Quackenbush, 2002]. These measurements are finally reported as expression levels in log base two of ratios of the quantity of RNA in a test sample over the quantity of RNA in a reference cell sample.

Each DNA microarray measurement gives a "snapshot" of gene expression levels of sampled cells at a global, whole genome scale. As such, the snapshot can be said to report on a state of the organism. This concept was recently studied by Van Driessche et al. [2005], who explored the idea of treating a mutant's expression profile, also called a transcriptional profile, as a "universal" phenotype and use it for inference of gene networks by means of epistasis analysis and abductive reasoning. This approach stems from a method [Zupan et al., 2003a,b] which uses "classical" qualitative phenotypes of an observed biological process or biological entity (*e.g.*, organism grows normally/grows slowly/does not grow, cells aggregate/do not aggregate, *etc.*) for epistasis analysis and abductive reasoning. When experimental results cannot be explained within a given background theory, a reasoning process, called abduction, can be used to explore all possible explanations (set of relations) for the observed experimental results. The newly constructed explanations (hypotheses) for the observed phenomena have to be consistent with a given background theory, and preference criteria are used to select 'the best' among alternative explanations. Four abductive inference patterns, forming the background theory of genetic regulation, are formalized and used by Zupan et al. [2003a]: influence, no-influence, epistasis and parallelism. While influence and no-influence describe the relation between a gene and a biological entity (*i.e.*, qualitative phenotype of the observed biological process or some other gene), epistasis and parallelism patterns are used to indirectly relate two (mutated) genes with respect to a common biological entity, and thus play a major role in determining the final genetic network. The logic of epistasis analysis was described by Avery and Wasserman [1992]. Epistasis analysis serves as a genetic tool for determining the order of action of genes in a regulatory hierarchy, in which the phenotype of the double mutant is compared with that of single mutants. The epistatic gene is said to act after the other gene in a regulatory network. The abductive inference patterns are stated in form of rules 'IF certain genetic experiments exist, THEN a certain relation between genes and a biological process is hypothesized' [Zupan et al., 2003a]. The two most important inference patterns (rules) are:

- Epistasis: IF genes g_1 and g_2 act in a liner pathway and $p_1 \neq p_2$ and $p_2 = p_{12}$ (this implies that $p_1 \neq p_{12}$), THEN gene g_1 influences gene g_2.

- Parallelism: IF $p_1 \neq p_{12}$ and $p_2 \neq p_{12}$ (note that there is no condition on p_1 and p_2), THEN g_1 and g_2 act in parallel pathways.

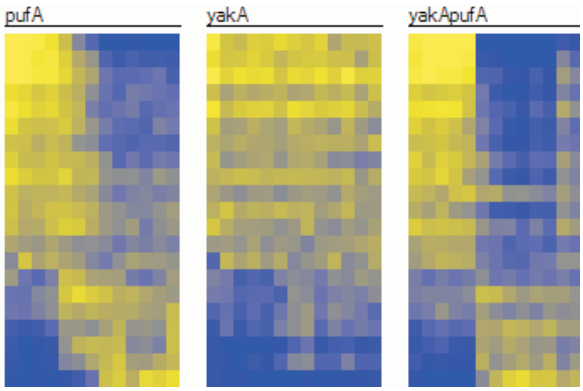
Variables p_1 and p_2 are qualitative phenotypes of single mutants in genes g_1 and g_2 respectively, and p_{12} is the qualitative phenotype of the double mutant where both genes g_1 and g_2 are mutated.

These same rules can also be applied when using global transcriptional phenotypes of wild-type, single and double mutants. Determining the exact value of a qualitative phenotype may be problematic, as qualitative states may not be so distinctive when observed experimentally, and mapping between an experimentally observed behavior and qualitative state is not trivial. However, their subsequent use in inference patterns is simple since it requires only the comparison ("equal" or "different") of symbolic values. On the contrary, measuring quantitative global-scale phenotypes with DNA microarray technology is in principle straightforward, while comparing and assessing the similarities of phenotypes proves to be more challenging. For example, how do we determine if one quantitative phenotype is different from another? Instead of using crisp degrees of similarity between quantitative phenotypes, as is the case with qualitative phenotypes where two phenotypes can only be "equal" or "different," a quantitative degree of similarity can be calculated and subsequently used. For example, applying statistical tests, *e.g.*, ANOVA, on repeated experimental measurements were shown to work for the task

Transcriptional phenotypes have been proven useful for reconstruction and reconfirmation of genetic networks initially inferred using classical phenotypes [Van Driessche et al., 2005]. For example, see Figure 2.1 for data and inferred epistatic relations between two genes in *D. discoideum*. The richness of information in transcriptional phenotypes allows further development of methods for investigation and refinement of genetic networks. In Chapter 7 we show how the proposed method for decomposition of mutant transcriptional profiles can be used to identify genes acting on parallel pathways and for inference of gene networks.

Mutant	Aggregation
pufA-	++
yakA-	−
yakA- *pufA*-	++

(a)



(b)

Figure 2.1: a) Qualitative and b) quantitative phenotypes used for epistasis analysis. Data for three *D. discoideum* deletion mutants is shown. a) The qualitative phenotype marks the degree of observed aggregation (all possible values are: $-, \pm, +, ++$) of mutant strains. b) The quantitative phenotype represents the global expression of strains in a time series (each row represent the average of a group of 250 most-similar genes, columns represent time points, from 0 to 24 hrs, in increments of two hours). In both cases the phenotype of *pufA*- mutant is similar to that of the double mutant (*yakA*- *pufA*-). Following the inference pattern for epistasis, *yakA*- is found to influence *pufA*- (*i.e.*, *pufA*- is epistatic to *yakA*-).

2.3 Related bioinformatics methods for the analysis of gene regulatory regions

Gene expression data alone does not provide enough information for the unequivocal identification of groups of genes that are regulated by same regulators [Tavazoie et al., 1999]. Other types of genomic data are also needed to further elucidate the underlying gene networks [Qiu, 2003]. In this Section, we focus mainly on methods for discovering relations between gene expression and DNA sequence of gene regulatory regions. We start with an overview of the biology of gene regulation and continue with an overview of published methods for the analysis of regulatory regions.

The DNA sequence of a gene is composed of three functionally distinct regions: the regulatory (or promoter) region, RNA-coding and the terminator region (see Figure 2.2). For some genes, the regulatory and coding regions can intertwine, which may further complicate the analysis. Determining and understanding the promoter structure is an important prerequisite to under-

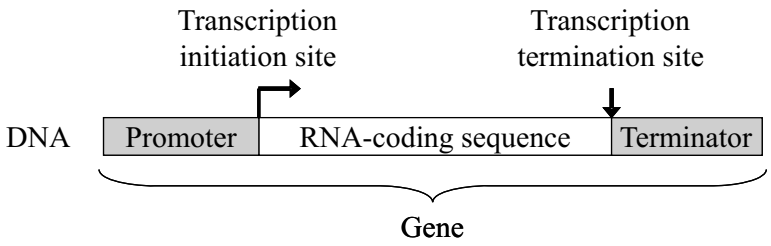


Figure 2.2: Simplified gene structure includes a promoter, RNA-coding, and terminator region.

standing gene regulation. The content of the regulatory region sequence determines which transcription factors will be recruited and bound to it. There are several possible scenarios for the start of transcription. The classic example is that the bound transcription factors are then themselves bound by an enzyme, the RNA polymerase. Together they open the DNA helix so that the RNA polymerase can move down the DNA. Another possibility is that the RNA polymerase sits idle on the DNA, waiting for a transcription factor to trigger the initiation of transcription. As the RNA polymerase travels along the DNA it assembles RNA – a transcript of the DNA. When transcription is completed, both the RNA transcript and RNA polymerase are released from DNA. Additional steps are then taken, depending on the type of RNA. For example, mRNA produced in the nucleus must undergo processing steps such as removal of introns (noncoding stretches of DNA which get transcribed but not translated), and (alternative) splicing of exons together (exons are stretches of DNA that are translated into chains of amino acids forming a protein) [Kimball, 2006]. The type, number, combinations of co-occurring transcription factors [Harbison et al., 2004; Yu et al., 2003; Bussemaker et al., 2001; Hannenhalli and Levy, 2002; GuhaThakurta and Stormo, 2001], and their timing [Lee et al., 2002] regulate the rate of gene transcription. The effect of a transcription factor can be positive, *i.e.*, it can facilitate the recruitment of the RNA polymerase, or it can have a negative, inhibitory effect by preventing the RNA polymerase from binding and starting transcription.

Enhancers are DNA sequence elements located thousands of base pairs upstream, downstream or even within the gene they control. Binding of "enhancer-binding" transcription factors is known to increase the rate of transcription of the gene by increasing the activity of the gene's promoter [Kimball, 2006;

Latchman, 1998]. Conversely, silencers are sequence elements to which "repressor-binding" transcription factors can bind to repress the expression of the gene they control. Direct binding of transcription factors in the immediate or distant regulatory region of genes is only one aspect of the complex mechanisms of gene expression regulation adopted by cells. Because there is a lack of other whole-genome data, most current computational studies on gene regulation focus on inference of relations between the content and structure of the gene regulatory region DNA sequence and gene expression measured using DNA microarrays.

Identification of gene regulatory regions and putative binding sites are the first crucial steps in such analyses. The regulatory region differs from the coding region in nucleotide and codon frequency. A codon is defined as three consecutive nucleotides that code for one of the twenty amino acids used for protein synthesis. This is successfully exploited by many promoter prediction algorithms (for an overview of such methods see Bajic et al. [2004]). Curated and computationally determined regulatory sequences are readily available for most model organisms, including *S. cerevisiae* (see www.yeastgenome.org) and *D. discoideum* (see http://dictybase.org).

The next important and well-studied step is the identification of transcription factors's putative and known binding sites in the regulatory regions of genes. These binding sites are short DNA sequences, comprising four to twenty nucleotides [Wasserman and Sandelin, 2004]. Most positions in the sequence are highly conserved (*i.e.*, have low sequence variation) and are frequent in the regulatory regions of co-regulated genes bound by the transcription factor. For computational analysis a matrix representation of binding sites is normally used. The matrix defines the frequency of the four bases (Adenine, Thymine, Guanine, and Cytosine) at each position in a binding site. The matrix is usually obtained from an aligned set of (putative) binding sites, and as such it represents an average sequence of the entire set of binding sites. The binding site can be presented to the user as a single consensus line, which gives the most frequent base at each position. Besides the four codes for bases A, T, G, and C, other standard IUB/IUPAC codes are allowed (*e.g.*, letter W codes for the two nucleic acids A and T forming the weak group, see Table 2.1). For computational analysis, this representation is not recommended because it can lead to loss of information. Sequence logos provide a richer, more informative graphic representation of binding sites [Schneider and Stephens, 1990] (see Fig-

Position	0	1	2	3	4	5	6	7
A	0	0	0	0	0	0.3	0.7	0
C	0	1	1	1	0	0.2	0.3	1
G	0	0	0	0	0.2	0.2	0	0
T	1	0	0	0	0.8	0.3	0	0
Single line consensus	T	C	C	C	T	W	A	C

Table 2.1: Matrix representation and single line consensus sequence of a putative transcription factor binding site.

Figure 2.3: Sequence logo representation of the binding site from Table 2.1.

ure 2.3). Contribution of the four bases at each position may be represented using heights of the letters, where the total height of a stack of letters may be proportional to the degree of sequence conservation measured in bits of information [Schneider and Stephens, 1990].

Data on experimentally confirmed or computationally inferred putative binding sites is available in public databases such as TRANSFAC [Wingender et al., 1996; Matys et al., 2006], EPD [Schmid et al., 2006], and SCPD [Zhu and Zhang, 1999]. When analyzing genes regulated by unknown regulators, one can find candidate or putative binding sites using local sequence alignment programs such as the MEME program [Bailey and Elkan, 1994]. These programs identify short, frequent subsequences in a given set of DNA sequences. A detailed description and evaluation of such tools has been reported recently by Tompa et al. [2005]. In their study, Tompa *et al.* comment that prediction of regulatory elements remains an extremely complex and challenging task for computational biologists. They report low absolute measures of correctness for the thirteen tools they tested. The algorithm, implemented in the tool Weeder Web [Pavesi et al., 2004], outperformed other tools in most tested domains and by most measures used in the assessment. However, the authors remain indecisive on the best tool, mainly because of problems with designing a good assessment. A big lack

of knowledge on transcription factor binding sites makes it difficult to construct representative data sets and to select the most appropriate statistics for evaluating the correctness of predictions. They suggest using "a few complementary tools in combination rather than relaying on a single one" [Tompa et al., 2005]. One striking fact is that the majority of tools perform much better on yeast data than on other species [Tompa et al., 2005] (*i.e.*, mouse, fly, and human). Again, this could be the result of the assessment bias described by the authors.

Most of contemporary methods for finding relations between gene structure and expression adopt the "group-by-expression" approach [Chiang et al., 2001]. They start by clustering genes based on their expression or use known gene functional annotation to form groups of genes, and then determine cluster-specific binding sites in each cluster (Figure 2.4a). Unsupervised methods for k-means and hierarchical clustering of gene expression [Eisen et al., 1998], supervised methods for gene function prediction, which were demonstrated to be useful in gene function prediction, such as Naïve Bayes, Decision trees [Curk et al., 2003], SVM [Brown et al., 2000], and rule induction for trend detection [Hvidsten et al., 2003] and temporal abstraction [Sacchi et al., 2005] can be applied to group genes. Subsequent steps of these "group-by-expression" approaches largely depend on the number and composition of initially identified gene clusters. Slight changes in clustering initialization or in method's parameters can result in possibly very different clusters. This can then consequently lead to discovery of different cluster-specific binding sites. Most common clustering approaches split genes into disjoint groups, forcing the discovery algorithm to use the information on each gene only when considering its corresponding cluster. This substantially limits the analysis, since it has been shown that genes can be regulated and respond in many different ways and perform various functions [Latchman, 1998; Ihmels et al., 2002]. Methods that do not strictly depend on the initial clustering of genes may have a distinct advantage in this respect.

An alternative to above is a "group-by-sequence" approach: starting with information about binding sites, this constructs groups of genes containing specific binding sites and continues by analyzing their expression (see Figure 2.4b). An example of such approach is a technique based on "Genome-Mean Expression Profiles" or GMEPs proposed by Chiang et al. [2001]. For a specific putative binding site, GMEP is defined as a weighted mean expression profile of

all genes that contain the binding site in their regulatory regions. Gene expression weights are proportional to the number of occurrences of the binding site in a gene. Weighting was introduced by the authors under the assumption that transcription factors might have a higher affinity to genes that contain multiple copies of their binding sites. This binding-site specific profile is then compared to the mean profile of all the examined genes. If the difference is not significant, a putative binding site is considered not to contain transcriptional information. Binding sites containing transcriptional regulatory information are thus expected to have their GMEP significantly different from the population mean. In spite of the simplicity, this straightforward method has been successfully applied to rediscover already known and also experimentally not yet confirmed single binding sites that regulate gene expression in many conditions [Chiang et al., 2001].

The approach of Chiang et al. [2001] fails to discover patterns that include combinations of two, three or more putative binding sites. This could be regarded as a major deficiency as it is biologically known that regulation of gene expression can be highly combinatorial [Wasserman and Sandelin, 2004; Qiu, 2003; Birnbaum et al., 2001] and it requires the coordinated presence of many bound transcription factors (*e.g.*, see pages 426-432 in [Alberts et al., 1994]). More advanced methods thus try to infer rules that describe the content of regulatory regions with more than one putative binding site [Beer and Tavazoie, 2004; Pilpel et al., 2001]. An early attempt in this direction was the work by Pilpel et al. [2001] where the authors used a set of 356 known or putative binding site sequences and regulatory regions for 4483 genes in *S. cerevisiae*. For each individual and all pairs of binding sites a score of coherence of gene expression observed under several different conditions is calculated. Expression coherence score was defined as a measure of overall similarity of the expression profiles of genes containing the binding site or pair of binding sites. To derive this score, their method first computes the Euclidean distances between all pairs of genes (number of pairs is $P = 0.5 \cdot K \cdot (K-1)$, where K is the number of genes that contain a given single or combination of two binding sites). The score is then defined as p/P, where p is number of gene pairs whose Euclidean distance is smaller than a threshold distance D. Threshold D is determined by random sampling of distances among all pairs of 100 genes and selecting D as the lowest value in the fifth percentile of the distribution of these distances. Next, the

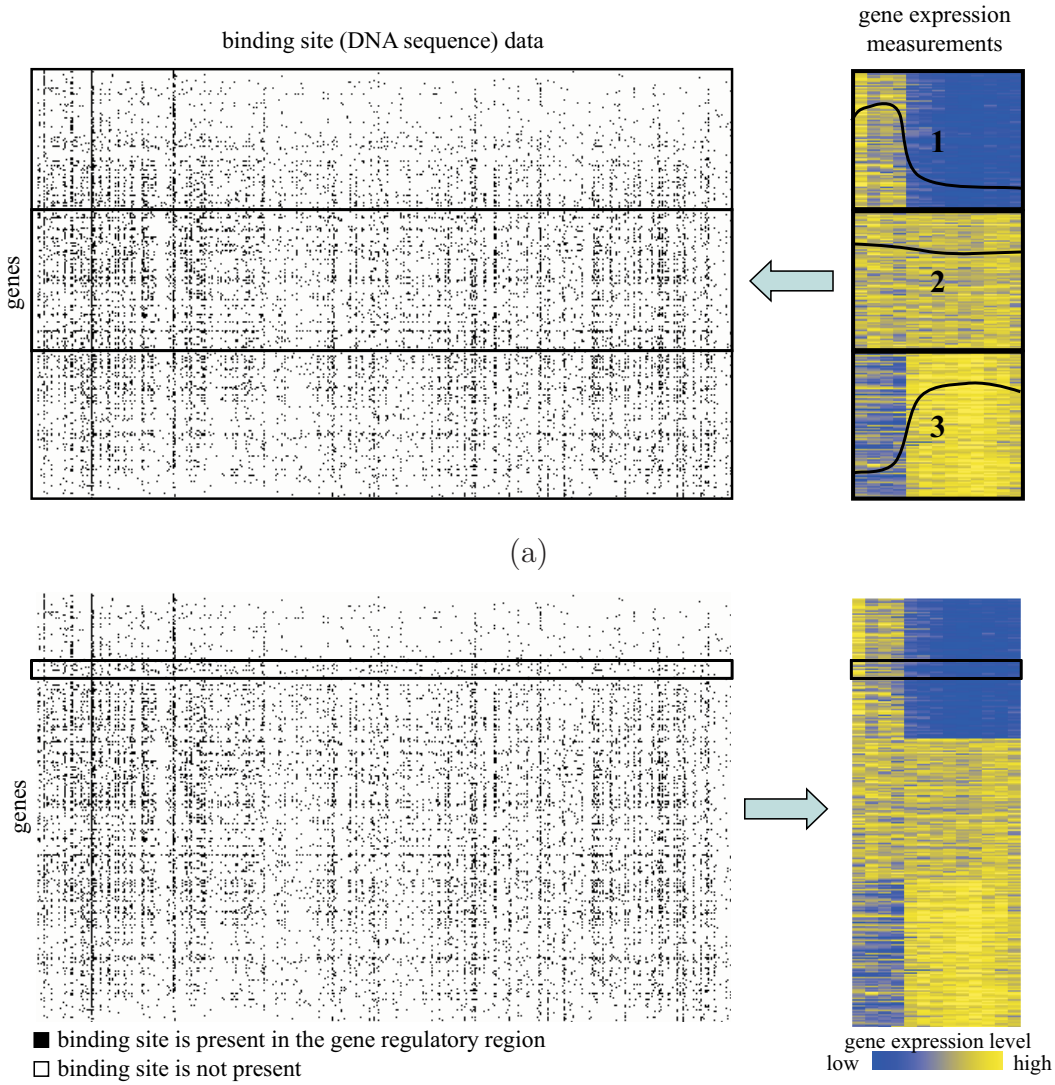
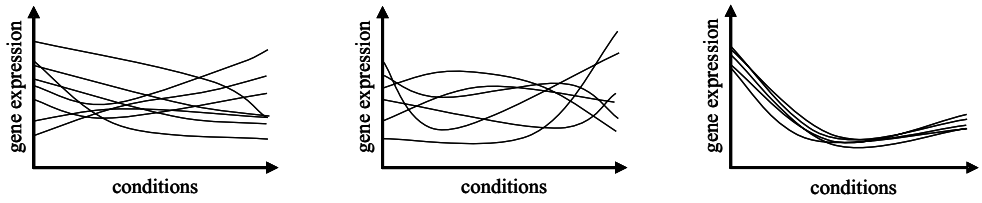


Figure 2.4: a) Group-by-expression and b) group-by-sequence approaches for finding relations between gene structure and expression. Binding site data in this example includes only the presence of a couple hundred binding sites in the regulatory region of genes. DNA microarray gene expression data is a time course of thirteen time points.

method checks for 'synergistic' pairs of binding sites. A pair of binding sites is considered synergistic, if the expression coherence score of genes containing both binding sites is significantly greater than that of genes containing either binding site alone (see Figure 2.5). Synergistic pairs of binding sites can then



(a) (b) (c)

Figure 2.5: "Synergistic" pair of binding sites. a) Expression profile of genes with binding site S_1 but not site S_2 in their regulatory regions. b) Genes with S_2 but not S_1. c) Genes with both sites (S_1 and S_2) form a more coherent group than genes with either binding site alone. Thus, sites S_1 and $S_2$are said to be synergistic.

be linked together to form a "synergy map" [Pilpel et al., 2001], which provides for a global overview of transcription networks, and helps the user to discern higher combinations of more than two binding sites.

The above approaches use exhaustive search and examine all possible combinations of binding sites. Because of a normally huge number of possible combinations, they most often concentrate on a search for pairs of binding sites. Exploration of higher-order combinations through exhaustive search is most often not feasible due to combinatorial explosion. For example, the number of all possible combinations of three binding sites, from a base of thousand binding sites available for modeling, quickly grows into hundreds of millions.

The cell's transcriptional program can also depend on absolute or relative orientation, order [Terai and Takagi, 2004], distance between binding sites and other landmarks in the regulatory region (*i.e.*, the translation start ATG, transcription start site TSS) [Harrison and DeLisi, 2002]. While Pilpel et al. [2001] observed that position and orientation of binding sites within the regulatory region is also correlated with expression, no systematic approach was proposed to identify such structural patterns. The idea was to some extent explored further by Beer and Tavazoie [2004] where more complex patterns that included positional information were considered. Authors showed that successful prediction of gene expression from sequence requires diverse and complex patterns that include constraints on binding site orientation, relative position and sequence similarity and included them within the predictive rules using operators for conjunction, disjunction and negation. To find such patterns, they first clus-

tered 2587 genes into 49 expression patterns measured in 255 conditions that include various environmental stresses and cell cycle data. They then applied a probabilistic approach (*i.e.*, Bayesian network) to describe relationships of probabilistic dependency between sequence features (presence and absence of 615 specific binding sites, their orientation, order and spacing between them) and 49 expression patterns. Using five-fold cross-validation they showed that the Bayesian network modeling approach correctly predicts expression patterns for 73% of 2587 genes used in *S. cerevisiae*. Bayesian networks were constructed iteratively. Constraints between binding sites already included in the network were added or removed in each iteration. The process was stopped when no modification improved the probability that the network is correct, for the given data. The method is highly combinatorial and examines many different refinements in every iteration. The authors report the running time of eight hours on a cluster of eight 2.4 GHz processors for the relatively small case study described above. While they have demonstrated good predictive accuracy, the approach has some limitations and problems. First, the initial choice of 49 clusters of gene expression was made at the beginning of the analysis, and was not repeated in the cross-validation. The choice on 49 clusters is somehow arbitrary and for reasons already stated could have significantly biased the inference. Second, probabilistic models, even if shown to be good predictors, are sometimes hard to interpret by the end user. Examples of two discovered patterns are given in Tables 2.2. Table 2.2a was interpreted by the authors as an example of conjunction, and Table 2.2b as disjunction of the presence of the two binding sites S_1 and S_2. Notice, however, that the difference between these two patterns is not clear. What is the threshold that distinguishes between conjunction and disjunction? Furthermore, values in Table 2.2a could suggest that binding site S_2 does not play a major role in this case, and a simple rule could be inferred instead, requiring only the presence of S_1. It is then up to the end user to develop an appropriate interpretation, that is, to convert the pattern to an operational rule. This may be easy for patterns with only two terms, but can get rather complicated with patterns presented in larger tables that include many terms.

Recent methods for analysis of regulatory regions include using SVM on a large set of 26 heterogeneous data types [Holloway et al., 2006], using probabilistic networks [Segal et al., 2003b,a, 2001; Friedman, 2004] to identify regulatory modules (*i.e.*, sets of genes) and their condition-specific regulators, us-

S_1	S_2	P
0	0	0.01
0	1	0.22
1	0	0.67
1	1	1

(a)

S_1	S_2	P
0	0	0.01
0	1	0.75
1	0	0.59
1	1	1

(b)

Table 2.2: Example combinations of two binding sites. The sites were reported by Beer and Tavazoie [2004]. The presence or absence of binding sites S_1 and S_2 in the promoter region of a gene is indicated by 1 or 0. Column P gives the probability, as estimated by the Bayesian network, of a specific expression pattern (patterns not shown) if the gene has a given combination of binding sites in its regulatory region. a) This example can be interpreted as a conjunction: the probability for the specific expression pattern is high only if both sites "S_1 AND S_2" are present. b) Disjunction: the probability for a specific expression pattern is high if any of two sites ("S_1 OR S_2") are present.

ing statistical motif discovery approaches combined with analysis of sequence conservation and motif positioning [Down et al., 2007], probabilistic clustering of regulatory sequences [van Nimwegen et al., 2002], fuzzy k-means clustering [Gasch and Eisen, 2002], transcription factor-centric clustering using expression data [Zhu et al., 2002], linear and step-wise regression between motifs and gene expression [Conlon et al., 2003].

The above description of the state of research in combining sequence and gene expression information motivated the research described in this Thesis. To overcome the limitations of the state-the-art approaches, we have developed a heuristic rule-based search method that is able, within a reasonable computation time, to identify complex symbolic structural patterns of gene regulatory regions. The proposed rule-based clustering method is guided by the information on the similarity of gene expression and explores only the most promising and coherent subgroups of genes with similar regulatory content.

Recent research on the properties of networks has revealed the existence of basic building blocks of most networks [Milo et al., 2002]. Feed-forward loops, bifurcations, chains, and other network properties, such as the network scale-free property of node distribution [Barabasi, 2002; Barabasi and Oltvai, 2004], can be found in a wide variety of networks, ranging from technological, social to biological networks. The associated graph theory, developed to identify graph properties of interest [Batada et al., 2007; Bertin et al., 2007], has al-

ready proved very useful in understanding and gaining deeper insight into the structure and properties of complex networks in many fields of science, *e.g.*, for human disease see the paper by Goh et al. [2007]. The visualizations of models inferred with rule-based clustering, which we propose in Chapter 5, provide a starting point for such analyses.

2.4 Summary

In this Chapter we have presented two main problems addressed in this Thesis: methods for automatic (re)construction of gene networks, which should include solving another important subproblem of analysis of gene regulatory regions. Since solving both problems requires searching in a highly combinatorial solution space, heuristic approaches are needed to find suboptimal solutions.

Chapter 3

Phenotype characterization and preliminary experiments

In this Chapter we present some initial experiments we have performed which motivated and guided the development of the various approaches presented in this Thesis.

3.1 Computational phenomics

Phenotype refers to the organism's morphological, biochemical or physiological properties. It describes the organism's total physical appearance, its specific traits, or behavior that may vary among individuals. These standard characterizations are relatively easy to observe, and as such widely used in classical genetics to reason about gene function. Classical genetics has greatly depended on observation of mutant phenotypes (*e.g.*, "mutant grows", "does not grow", "sporulates", "cells aggregate", *etc.*). For example, in functional genomics, gene function is inferred by controlling the environment and observing the phenotype under changed activity of the gene, *e.g.*, the phenotype of a wild-type organism and the phenotype of a knock-out deletion mutant are compared. With the introduction of new genome-wide techniques in biotechnology and high-throughput experimentation, manual characterization of a classical morphological phenotype is no longer practical, and at the same time also insufficient, because it usually carries relatively little information about the complete state of the organism. Present technology allows designing a highly complex and fully automated image acquisition and analysis system to quantify specific morpho-

logical characteristics of an organism for specific studies (*e.g.*, to determine the roles of genes involved in growth of the organism). In would be in principle impossible to design a system that would capture all the necessary morphological and structural features needed to describe the complete state of an organism, which is required for genome-wide scale studies [Zupan et al., 2006]. Moreover, some changes in the state of the organism may not be reflected in morphology. Classical phenotypes may therefore be inadequate for whole-genome studies, because they do not encode the complete state of the organism at a sufficient resolution that would allow the detection of differences in the organism's state which are induced by changes in the environment or genotype.

Below we give an overview of two types of genome-wide gene profile data and show that they can be used in the analysis of gene function, and compare them with standard gene expression profiles, and investigate their ability to characterize gene function. Our initial exploration in this direction [Curk et al., 2005b] was inspired by the investigation of Stuart et al. [2003] where they have proposed a method to cluster genes based on the "guilty-by-association principle." The method, called gene co-expression networks, was used to measure the correlation between similarity in gene expression profiles (gene co-expression) and similarity in gene function annotation. In addition to performance measures originally proposed by Stuart et al. [2003], we here propose to use receiver operating characteristic (ROC) analysis [Provost and Fawcett, 2001] to measure the concordance between gene profile data and functional annotation, which we believe to be more appropriate. Our preliminary investigations described in this section show that using particular gene characterizations (*i.e.*, a specific type of gene profile data) results in better predictions of specific gene functions, which motivated the choice of particular gene characterization in development of the computational approaches in this Thesis. Additionally, we show that each type of gene profile data is more suitable for predicting separate and functionally linked sub-graphs in the ontology of gene functional annotation [Ashburner et al., 2000].

3.2 Types of computational phenotypes

Collection of DNA microarray data on gene expression, measured under different experimental conditions or time points, is a type of data normally used

to infer gene function. This type of data is often referred to as *gene expression profile*. The underlying assumption, referred to as the *guilty-by-association* principle, states that genes with similar expression profiles should have similar functionality. When predicting the function of a gene, the annotation shared by the majority of genes with similar profiles is determined and assigned to the gene whose function is being predicted. Clustering of gene expression profiles [Eisen et al., 1998] is the most often used method that applies to this principle.

Two other data types, data on mutant transcriptional profiles and data on mutant sensitivity profiles, both made possible by recent technological advances, provide alternatives for gene association analysis. For both data types, gene function is related to the phenotype of its corresponding mutant.

The *transcriptional profile (phenotype)* of a mutant is defined as a collection of DNA microarray measurements of gene expression of all genes in the mutant strain. Changes in gene expression levels of individual genes are presumed to reflect specific cellular states and most gene expression profiling studies try to identify genes that respond to specific conditions or treatments and do not use data on mutant expression. As already stated, the idea that expression of all genes could be used as an indication of cellular state, has recently gained a substantial attention [Bittner et al., 2000; Alizadeh et al., 2000; Hughes et al., 2000]. It has been shown that the mutant expression profile could serve as a surrogate for an universal phenotype [Van Driessche et al., 2005; Hughes et al., 2000; Hughes, 2005] and as such is believed to be very informative for assigning gene function. Whole-genome expression profiling of mutants thus holds great promise for rapid genome function analysis. Instead of associating gene function to its expression pattern under different conditions, as it is done when analyzing gene expression profiles, one can consider the entire DNA microarray profile of a mutant strain as an indicator of function of the mutated gene. The utility of this approach has been successfully demonstrated in yeast by Hughes et al. [2000] and in cancer cell characterization [Bittner et al., 2000; Alizadeh et al., 2000]. The reason to use whole-genome mutant phenotypes as opposed to a specific gene expression profile also follows from the finding that (single) gene expression and gene function may show very little correlation on a global scale. This correlation was reported to occur in less than 10% of the cases [Winzeler et al., 1999].

The *mutant sensitivity profile* is defined as a quantitative characteristic of

the mutant strain under specific treatment (*i.e.*, sensitivity to a specific treatment). This type of mutant-based phenotype can be obtained by using uniquely "barcoded" viable deletion mutants. The "barcoding" technique allows to distinguish and to measure the relative abundance of mutants in a pool of all viable deletion mutants. This provides an indication of the sensitivity and fitness of each mutant under a specific environmental change or treatment. A quantitative sensitivity profile of a mutant is thus defined as a collection of such measurements under various treatments or conditions [Brown et al., 2000].

In our preliminary experiments with these phenotypes we have used publicly available microarray gene expression data sets, sensitivity profile data sets and annotations of gene function on yeast *S. cerevisiae*. Data on gene expression profiles was taken from a study by Spellman *et al.* where they have measured gene expression in 73 different time points of the *S. cerevisiae* cell-cycle [Spellman et al., 1998]. For mutant transcriptional profiles we have used a compendium of whole-genome gene expression DNA microarray measurements of 300 diverse mutants and chemical treatments in *S. cerevisiae*, as reported by Hughes et al. [2000]. Data on mutant sensitivity profiles were made available by Brown et al. [2000] and includes sensitivity measurements of 4756 viable strains, measured in 51 diverse treatments (cytotoxic or cytostatic agents).

To test the results of our predictions we have used the existing functional annotations of *S. cerevisiae* genes. We considered 79 Gene Ontology (GO) terms [Ashburner et al., 2000] which were annotated to at least 10 genes. We also used a subset of 28 high-level GO "slim terms" that best represent the major biological processes, functions, and cellular components found in *S. cerevisiae* (data available at http://www.yeastgenome.org).

3.3 Gene co-expression networks

To quantify the degree of correlation between various characterizations of gene profile data and functional annotation, we have used gene co-expression networks [Stuart et al., 2003]. Gene co-expression networks were originally proposed as a straightforward computational method to cluster genes based on similarity in their gene expression profiles which are measured across various conditions (cell cycle, temperature shock, acid shock, *etc.*). The gene co-expression network can be represented as a graph where nodes represent genes. Two genes

in the network are connected if the similarity in their characterization (gene profile) is above a certain user-defined threshold. The method uses a distance (similarity) function to measure distance between pairs of genes. Stuart *et al.* used Pearson correlation to measure similarity among gene expression profiles. We have also used Pearson correlation to assess the similarity for the other two phenotypes used for gene characterization.

The structure of the network largely depends on the selected similarity threshold. By varying the threshold different networks arise: from relatively unconnected networks, with edges relating only genes with most similar profiles, to highly connected networks that include edges linking genes with low or zero similarity. Based on the assumption that genes with similar characterization profiles are also functionally related, one can expect that genes with similar function would form interconnected groups. Stuart *et al.* proposed two measures to quantify this property of a network. The first measure is *coverage*, which reports on the proportion of genes belonging to a selected functional class that are also connected to at least one gene from the same class. The other measure is *accuracy*, which is expressed as the proportion of edges that connect two genes from the observed class over the number of all edges connecting to genes in that class, *i.e.*, number of edges connecting two genes from the observed class as well as edges connecting a gene from the observed class with a gene outside the class. For example, a gene co-expression network of seven genes (G1-G7) for an arbitrarily selected threshold (not given) is shown in Figure 3.1. There are five genes annotated to the observed class (G1, G2, G5, G6 and G7, colored in blue), but only two of them are connected to each other (G1 and G2). The class coverage of this network is then $2/5 = 0.4$. There is only one edge connecting two genes in observed class (edge G1-G2). There are four edges coming from genes in the observed class (edges G1-G2, G2-G4, G5-G4, G7-G4). The accuracy of the network is then $1/4 = 0.25$.

Using an appropriate method for graph layout visualization (*e.g.*, the program Pajek by Batagelj and Mrvar [2003]) of gene co-expression networks can reveal structural information such as outliers or tight gene clusters. In addition, varying the threshold also provides a general overview of the relation between a specific gene function and gene profile similarity. In this work we do not report on network visualization, because we are primarily interested in the network's ability to group genes with same functional annotation. Therefore, we only

build networks and use them to measure their predictive ability on a set of selected gene functional annotations.

The main difficulty with the measure of network performance proposed by Stuart *et al.* is that it does not provide for a single quantitative measure of success, but it reports two values (*i.e.*, on coverage and on accuracy). This complicates the comparison of the predictive ability of different networks. In particular, we would like to know which type of gene profile data better relates genes in a selected functional class, and thus require a single-valued and clear scoring method.

We therefore propose an alternative scoring to that used by Stuart et al. [2003]. In this, we borrow from a well-known method established in statistics and recently much used in the field of machine learning. The approach is called receiver operating characteristic (ROC) analysis [Provost and Fawcett, 2001; Vuk and Curk, 2006; Fawcett, 2003] and is widely used to measure the ability of a probabilistic classification model to discriminate among classes. In a typical setting with two classes, one class is represented with value zero (also called negative class), and the other class with value one (positive class). The classification method assigns a value (score), in the interval between zero and one, to each example. This value indicates a likelihood that the example belongs to one class (*i.e.*, to the positive class represented with value one). Examples are then ordered by decreasing score from which a ROC graph is formed and the area under the ROC curve (AUC) is calculated.

ROC curves are plotted in the "FP-rate and TP-rate" space (see Figure 3.2b for example). False positive (FP) rate is the number of examples falsely assigned to the positive class (those examples are actually negative) divided by the number of all negative examples. True positive (TP) rate is the number of correctly predicted positive examples divided by the number of all positive examples. A "sliding" threshold is used to calculate the FP-rate and TP-rate values at each possible threshold. The threshold is initially set to the highest score, and subsequently decreased to lower scores. At each step, predicted examples, with score equal or higher to the threshold, are used to calculate the FP-rate and TP-rate, and to plot a point of the ROC curve at the corresponding coordinates.

The calculated value of area under the ROC curve (AUC) can be between zero and one. A value of one indicates a perfect ability of the tested model

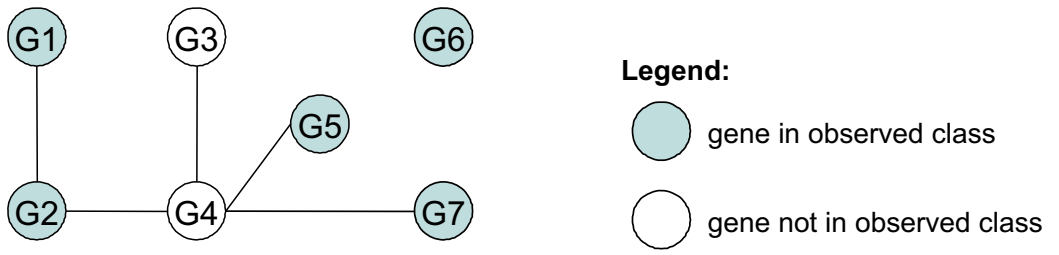


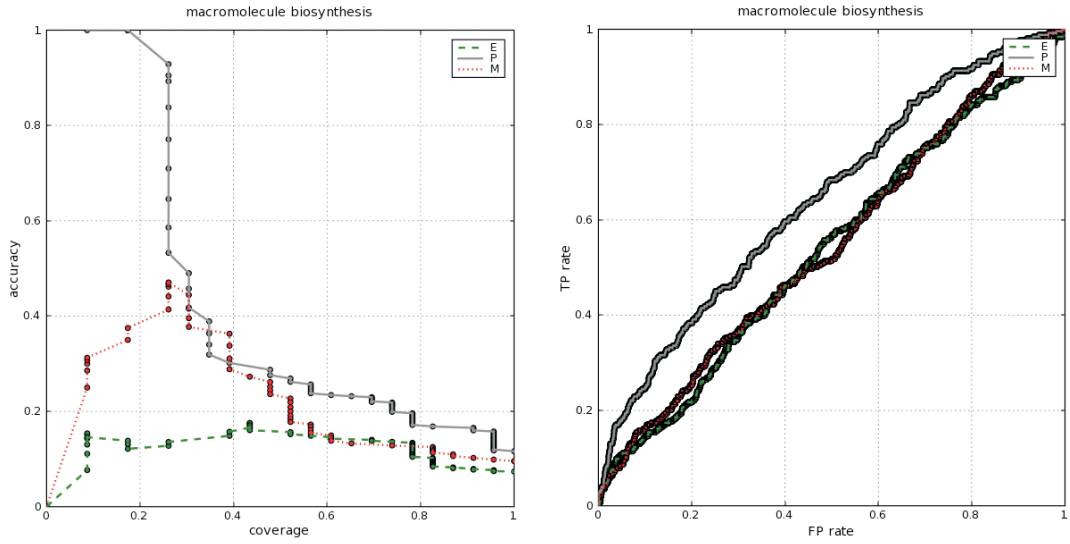
Figure 3.1: An example gene co-expression network. Seven genes (G1-G7) are shown. Genes with a correlation score above a selected threshold (not shown) are connected by an edge. Observed class coverage is $2/5 = 0.4$, accuracy is $1/4 = 0.25$.

to discriminate between the two classes (the ROC curve intersects the point of zero FP-rate and perfect TP-rate of value one), a value of 0.5 indicates random guessing – the model cannot distinguish between classes. For details on ROC analysis see papers by Provost and Fawcett [2001]; Fawcett [2003]. AUC is a single value representation of the model's discriminative ability and as such more suitable for direct comparisons of the predictive ability of different models then the metric(s) proposed by Stuart et al. [2003].

In our work, instead of focusing on genes we focus on edges when modeling gene co-expression networks. In this setting, an edge connecting two genes from same class under consideration is considered to be a true positive (TP) example. An edge connecting two genes outside the class is considered a true negative (TN). Edges connecting one gene in class with another gene outside the class are considered false positives (FP). Applying ROC analysis on gene co-expression networks is then straightforward. All edges in a fully connected network are examples. The score assigned to each edge is the calculated similarity of the two genes connected by the edge. All edges are considered in the ROC analysis without the need for selecting a (similarity) threshold.

3.4 Visualizing the predictive performance and Gene Ontology relations

The predictive performance of gene co-expression networks can be visualized with the coverage-accuracy graph proposed by Stuart et al. [2003]. The results



(a) (b)

Figure 3.2: a) Coverage-accuracy plot and b) ROC plot of gene co-expression networks for GO term "macromolecule biosynthesis." Results achieved on all three data types are shown (E – gene expression profile, P – mutant sensitivity profile, M – mutant transcriptional phenotype).

of ROC performance analysis can be displayed in form of a performance graph which can also be used to visually evaluate and compare classifiers.

Figure 3.2 shows an example of the two plots for the group of genes annotated to the GO term "macromolecule biosynthesis." While one could conclude from the coverage-accuracy plot in Figure 3.2 that the gene co-expression network built from mutant sensitivity profile (P) data performs best, it is unclear which of the remaining two networks is second best. The gene co-expression network built from mutant transcriptional phenotype data (M) may appear to better perform than the network constructed from gene expression profile data (E) (see Figure 3.2a). However, the ROC curve in Figure 3.2b clearly shows that using mutant sensitivity profile data (P) results in the best gene co-expression network, while the other two data types produce gene co-expression networks with equally poor discriminative ability for the group of genes involved in "macromolecule biosynthesis."

Complete results for a selection of GO slim terms, with at least ten genes annotated to the GO term, are given in Table 3.1. Each row in Table 3.1 shows the area under the ROC curve (AUC) achieved by gene co-expression networks

M AUC	E AUC	P AUC	GO term
0.736	0.598	0.752	C_mitochondrial envelope
0.631	0.670	0.805	C_ribosome
0.613	0.630	0.648	F_structural molecule activity
0.603	0.572	0.572	C_endoplasmic reticulum
0.583	0.538	0.572	C_mitochondrion
0.581	0.475	0.524	P_cell wall organization and biogenesis
0.575	0.506	0.523	P_DNA metabolism
0.574	0.566	0.751	P_protein biosynthesis
0.554	0.523	0.526	P_organelle organization and biogenesis
0.552	0.543	0.547	C_endomembrane system
0.551	0.547	0.497	C_membrane
0.549	0.517	0.507	F_protein binding
0.542	0.599	0.514	P_protein modification
0.540	0.536	0.522	P_cell cycle
0.536	0.488	0.494	F_DNA binding
0.533	0.497	0.513	F_hydrolase activity
0.529	0.524	0.513	C_cytoplasm
0.520	0.503	0.505	F_transcription regulator activity
0.512	0.494	0.500	P_transport
0.508	0.502	0.493	C_nucleus
0.505	0.505	0.495	P_transcription
0.504	0.597	0.477	P_lipid metabolism
0.504	0.522	0.511	F_transferase activity
0.497	0.622	0.525	F_oxidoreductase activity
0.497	0.508	0.508	P_morphogenesis
0.494	0.476	0.449	P_signal transduction
0.487	0.610	0.475	P_conjugation
0.483	0.514	0.561	P_response to stress
0.476	0.492	0.601	P_vesicle-mediated transport
0.475	0.494	0.470	F_enzyme regulator activity

Table 3.1: Predictive performance of gene co-expression networks built using all three types of gene profile data. Results for GO slim terms on data for mutant transcriptional phenotypes (M), gene expression profile (E), and mutant sensitivity phenotype (P) are shown. GO term prefix indicates the term's GO aspect: molecular function (F), biological process (P), and cell compartment (C).

for a specific GO term annotation, for each of the three computational phenotypes used to build a gene co-expression network. For example, the term "mitochondrial envelope" from the cell compartment ("C") aspect of GO can be reasonably well predicted by a gene co-expression network built from mutant transcriptional (M) or mutant sensitivity phenotype (P) with AUC value ~ 0.74, but the same GO term can not be well predicted from gene expression profile data (E), where AUC is less than 0.6. To compare the predictive ability of gene co-expression networks built on the three different types of data we have made pair-wise comparisons of the predictive performance of gene co-expression networks for each GO term. These comparisons are summarized in Table 3.2, and can answer the question on which type of data is more suitable for building gene co-expression networks that best predict specific GO terms.

M - E	GO term
-0.125	F_oxidoreductase activity
-0.122	P_conjugation
-0.092	P_lipid metabolism
-0.057	P_protein modification
-0.031	P_response to stress
-0.019	F_enzyme regulator activity
-0.019	F_transferase activity
-0.017	F_structural molecule activity
-0.016	P_vesicle-mediated transport
-0.011	P_morphogenesis
...	...
0.008	P_protein biosynthesis
0.016	F_transcription regulator activity
0.018	P_transport
0.018	P_signal transduction
0.031	P_organelle organization and biogenesis
0.032	F_protein binding
0.036	F_hydrolase activity
0.048	F_DNA binding
0.069	P_DNA metabolism
0.105	P_cell wall organization and biogenesis

M - P	GO term
-0.176	P_protein biosynthesis
-0.125	P_vesicle-mediated transport
-0.078	P_response to stress
-0.035	F_structural molecule activity
-0.028	F_oxidoreductase activity
-0.011	P_morphogenesis
-0.007	F_transferase activity
0.005	F_enzyme regulator activity
0.010	P_transcription
0.012	P_transport
...	...
0.018	P_cell cycle
0.020	F_hydrolase activity
0.027	P_organelle organization and biogenesis
0.027	P_lipid metabolism
0.028	P_protein modification
0.042	F_protein binding
0.042	F_DNA binding
0.045	P_signal transduction
0.052	P_DNA metabolism
0.056	P_cell wall organization and biogenesis

P - E	GO term
-0.134	P_conjugation
-0.120	P_lipid metabolism
-0.098	F_oxidoreductase activity
-0.085	P_protein modification
-0.027	P_signal transduction
-0.024	F_enzyme regulator activity
-0.014	P_cell cycle
-0.011	F_transferase activity
-0.010	F_protein binding
-0.009	P_transcription
...	...
0.004	P_organelle organization and biogenesis
0.006	F_DNA binding
0.006	P_transport
0.016	F_hydrolase activity
0.017	P_DNA metabolism
0.018	F_structural molecule activity
0.047	P_response to stress
0.049	P_cell wall organization and biogenesis
0.108	P_vesicle-mediated transport
0.185	P_protein biosynthesis

Table 3.2: Pair-wise comparison of predictive performance of gene co-expression networks from Table 3.1. Top and bottom ten differences in AUCs achieved on various types of gene profiles (M *vs.* E, M *vs.* P, and P *vs.* E) are shown and sorted by increasing difference of AUCs.

Looking at the first column ("M - E") and first row, the difference -0.125 in AUCs for "F_oxidoreductase activity" indicates that the GO term "oxidoreductase activity" from the gene function aspect of GO can be better predicted with a gene co-expression network that was built using the gene expression profile data (reported AUC using E in Table 3.1 for the term is 0.622) than the network

built from the mutant transcriptional phenotype data (reported AUC using M in Table 3.1 for the term is 0.497).

To visualize and compare the performance of gene co-expression networks we have used the Radial coordinate visualization (RadViz) method proposed by Ankerst et al. [1996]. The RadViz visualization can map a set of m-dimensional points inside a unit circle on a two-dimensional plane. Each of the m dimensions is represented by a special point on the unit circle, called anchor. Anchors are normally placed equidistantly on the unit circle. How a data point is projected inside the circle is determined by its values in each dimension. An intuitive explanation is the spring metaphor, where each data point is said to be connected to all anchors by springs. The stiffness of each spring is determined by the value of the data point in the corresponding dimension, the higher the value the stiffer the spring is. The point is positioned where equilibrium is reached.

In our study, each GO term (functional class) represents a data point in the RadViz plot (*e.g.*, see Figure 3.3). The three dimensions or anchors represent the measured AUCs of gene co-expression networks build from the three corresponding types of gene profile data (*i.e.*, 'E' for the gene co-expression network built using gene profile data, 'M' for mutant transcriptional phenotype data, and 'P' for mutant sensitivity profile data). GO terms are placed depending on the achieved AUCs. GO terms close to the center are equally well characterized by all three types of gene profile data. GO terms closer to an anchor are better predicted by the data type represented by the anchor. For example, points closer to the mutant-based transcription phenotype anchor M (*e.g.*, the term "signal transduction" from the GO aspect "biological process") can be better predicted by gene co-expression networks build using mutant transcriptional phenotype data than by gene co-expression networks built using the other two data types. Points in Figure 3.3 (and in all other subsequent figures) which are closest to the same anchor are of same color (and shape). This is just to indicate those GO terms that are better predicted with gene co-expression networks built from the corresponding data (*e.g.*, red circles are GO terms closest to the anchor of mutant transcriptional phenotype M data, green rectangles are GO terms closest to the gene expression profile data E, and gray triangles are GO terms closest to mutant sensitivity profile data P).

The predictive performance of gene co-expression networks on all three data types is shown in the RadViz visualization in Figure 3.3. Mutant transcrip-

tional profiles were the most predictive for the GO term "signal transduction" that contains genes coding for regulation of signaling. This finding is in agreement with the fact that the function of a signaling gene (*e.g.*, a transcription factor) is best determined by the effect on its downstream target genes or other messengers in the signaling cascade, which ultimately lead to a change in the functioning of the cell. Mutant sensitivity profile was the most predictive for GO terms "vesicle-mediated transport" and "response to stress". These genes are important in the response of yeast cells to the chemicals used for generating the mutant phenotypic profile. Therefore, this case demonstrates the importance of the biological context of the experiment for the information value that can be derived. Surprisingly, we have found that mutant transcriptional profiles (M), which are presumed to encode the global state of the organism, were surprisingly less informative than expected.

We then tested whether GO terms which are close to each other on the RadViz plot are also close in Gene Ontology. To perform this analysis, we superimposed the Gene Ontology hierarchy over the RadViz plot by connecting GO terms that are also, either directly or indirectly, connected in the Gene Ontology graph. Note that connections in GO are directed and are always pointing from the parent GO term node to its children (more specific) GO term nodes. We then calculated the level of clustering of GO terms better predicted by each type of profile by calculating the S/A ratio: number of directed connections among GO terms better predicted by the same gene profile data type (S) over the number of all connections pointing from those same GO terms (A). The results for the three different annotation aspects of Gene Ontology are presented in Figure 3.4.

Graphs in Figure 3.4 support our claim that some functional annotation (GO terms) are better characterized with a specific type of gene profile data. The majority of arrows are pointing away from the center of the RadViz graphs, shown as the "root" node of Gene Ontology. This indicates that the prevalence of a specific gene characterization approach is greater for more specific functions (which are placed closer to anchors). GO terms, better predicted by a specific gene profile data type, are also clustered in all three ontology (sub)graphs (Figures 3.4 a-c), one for each aspect of GO. This provides additional support for our claim on the relation between gene-characterization approaches and their utility for prediction of specific functional classes.

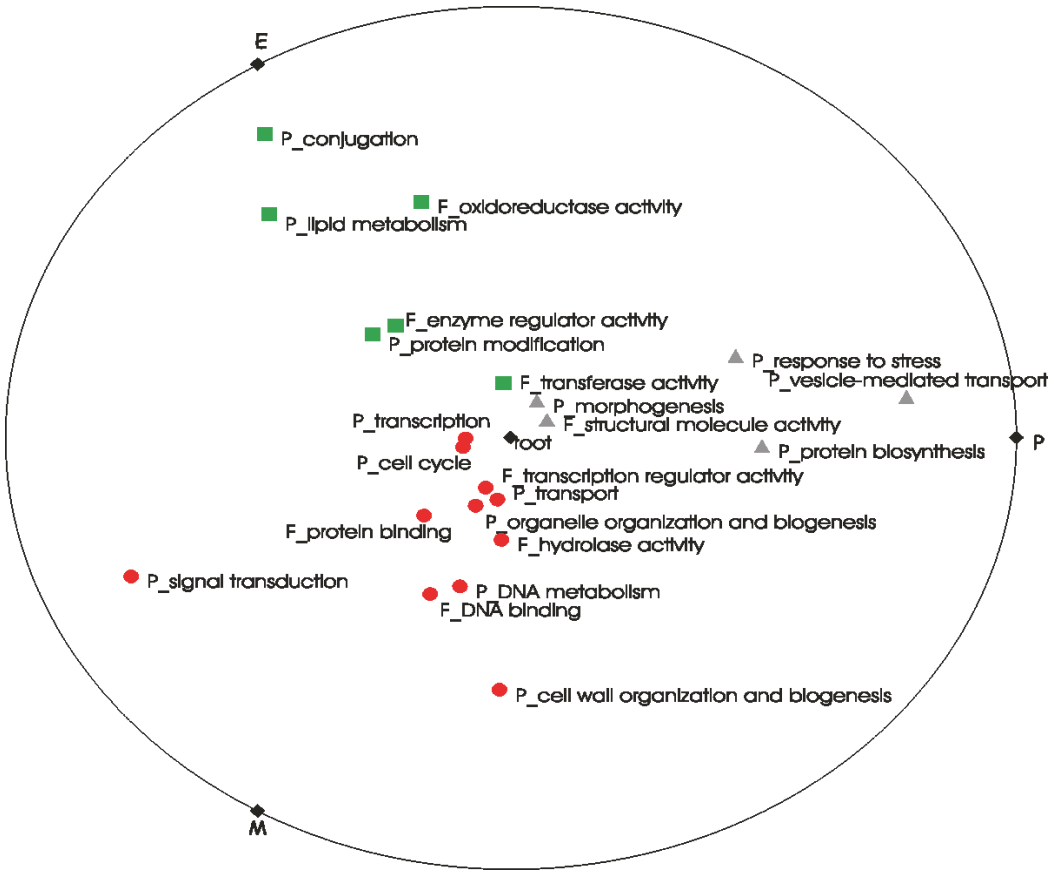
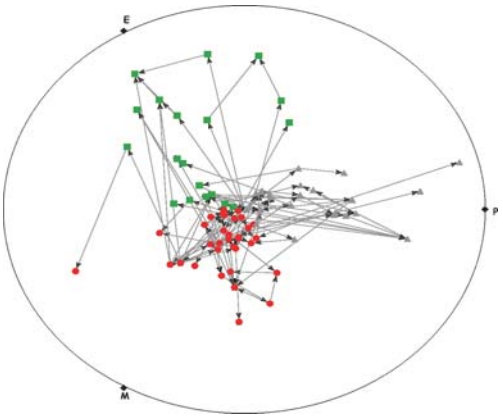


Figure 3.3: RadViz plot of the predictive performance of gene co-expression networks, for all three types of data. GO terms closest to an anchor are of same color and shape. Only GO slim terms with at least ten annotated genes are shown.

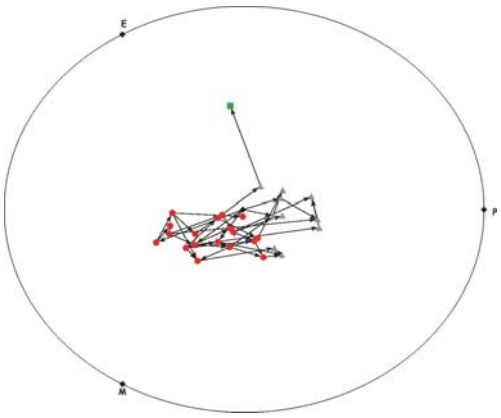
3.5 Discussion

In this Chapter we have introduced the concept of computational phenotype. We have extended gene co-expression networks, which were initially developed by Stuart et al. [2003] for gene expression profile data. In their work, Stuart *et al.* assumed the "guilty-by-association" principle, which states that correlated expression patterns of (evolutionary highly conserved) genes under diverse conditions imply functional relation. Although using expression data that originated from mutants Hughes et al. [2000], their conclusions were based only on gene-expression profiles. Mutant-based phenotypes were not investigated. Here we showed how gene co-expression networks can be extended to handle the other two, mutant-based types of gene profile data, *i.e.*, mutant transcriptional phe-



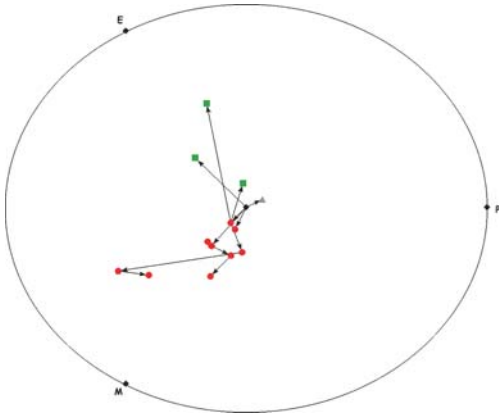
GO terms closest to anchor	clustering ratio	(S/A)
P (gray triangle)	0.789	(30/38)
E (green square)	0.486	(18/37)
M (red circle)	0.627	(52/83)

(a)



GO terms closest to anchor	clustering ratio	(S/A)
P (gray triangle)	0.875	(7/8)
E (green square)	nan	(0/0)
M (red circle)	0.738	(31/42)

(b)



GO terms closest to anchor	clustering ratio	(S/A)
P (gray triangle)	nan	(0/0)
E (green square)	nan	(0/0)
M (red circle)	0.800	(8/10)

(c)

Figure 3.4: RadViz plots of GO terms from a) biological processes, b) cellular components, and c) molecular functions aspects of Gene Ontology. Anchors are indicated by diamond points (E – gene expression profile, P – mutant-based sensitivity profile, M – mutant-based transcriptional phenotype). GO terms better predicted with E, M, or P are shown as green boxes, red ellipses, and as gray triangles, respectively. A black diamond in the center of the plot indicates the root of GO. Directed arrows show the parent → child relations of GO terms. GO terms with at least ten annotated genes that are appearing in the mutant data are shown.

notype and mutant sensitivity profile data. We have built gene co-expression networks using the three different computational phenotypes for the budding yeast *Saccharomyces cerevisiae*.

Overall, we found no clear "winner" or most informative data type when we compared the results of gene co-expression networks built from classic gene expression profiles and the two mutant-based profile types. We have observed that the two mutant-based profile types (*i.e.*, observing global expression patterns of mutants under the same conditions, or observing the sensitivity of mutants under different treatments) produce gene co-expression networks with better discriminative ability for function prediction (see Table 3.2) than those networks built on gene expression profiles (*i.e.*, observing gene expression under different conditions). One conclusion that also needs to be drawn from this analysis is that, when studying a particular function, there may be a clear difference between the two approaches that could be explained with existing biological knowledge. This is a clear indication that all three sources of experimental data should be used in order to successfully predict specific gene functions. Further work includes investigating and developing ways to automatically learn how to combine all three profile types for better function prediction. Further along these lines, we have shown examples how entire subgroups of gene functional classes can be better predicted from different types of gene profile characterization.

The principal novelty of the work presented here is the direct comparison of the utility of gene expression profiles and two mutant-based phenotypes for gene function prediction using the area under ROC curves for gene co-expression network, and the analysis of obtained scores with RadViz visualizations. Gene expression and mutant-based transcriptional profiles were first studied together and qualitatively compared by Hughes et al. [2000]. They were found to complement each other, which is in agreement with our study. We extended this to mutant sensitivity profiles. The utility of mutant-based phenotypes depends on development of appropriate high-throughput technology, and with recent advances and rise in reported mutant-based studies we strongly believe that such phenotypes can and will complement the traditional gene expression profiles in functional genomics.

Chapter 4

Rule-based clustering and feature construction

In this Chapter we propose a clustering method that uses an attribute-based representation of examples and calculates the intra-cluster similarity on a subset of attributes, with the goal to identify distinctive clusters of examples through logic assertion on the remaining set of attributes. The discovered model is thus composed of a set of IF-THEN rules. To construct assertions in the conditional part of the rule, the method incorporates an on-the-fly feature construction step which is suitable for complex, feature and background-knowledge rich problem domains. An important advantage of the proposed approach is its ability to discover overlapping clusters, that is, clusters that may share a subset of examples. We also address time-complexity of the method and describe means for quantitative evaluation of the approach.

4.1 Motivation and goal

Although the method proposed in this Chapter is general and could be applied to other problem areas, the principal motivation came from its application in bioinformatics and the analysis of gene regulatory sequence and expression data. Namely, we developed the approach in order to overcome the difficulties associated with current approaches which are used for relating gene sequence to gene expression (details on the methods are described in Section 2.2). The principal drawback for a set of approaches that start with clustering of gene expression is their potential sensitivity to the selection of the type of clustering method and

the choice of parameter values, especially the number of clusters. As stated in Section 2.2, these can greatly influence any subsequent inference of descriptions of discovered clusters. For these reasons, rule-based clustering heavily relies on finding patterns in gene regulatory sequences instead, and it uses information on gene expression to guide the search in the space of all possible gene clusters. The space of all possible clusters is determined by the user-defined descriptive language (*i.e.*, the language is used to encode the conditional part of a rule which identifies a cluster).

The goal of rule-based clustering is to find clusters of similar examples for which a common cluster-specific symbolic description can be inferred. Note that unlike common clustering methods, like hierarchical clustering and k-means algorithm, the proposed methods uses a distinct set of attributes to reason on similarity of examples, and is uses another set of attributes to infer symbolic descriptions – assertions that are common to a set of examples within the same cluster (see Figure 4.1 for the two sets of attributes). Another distinction is that we allow examples to be assigned into several clusters. In this sense, the method bears similarity to the principal idea of subset discovery approaches, again with the difference that one subset of attributes is used for cluster description and another to reason on similarity of examples.

The output of the method is a set of example clusters, each described by a set of one or many rules of the following form:

IF *description* **THEN** *prototype*.

The conditional part of a rule is a symbolic description that defines the cluster membership (*i.e.*, it describes all covered examples that match the symbolic description). Prototype defines the properties of the cluster. The prototype is computed from examples in the training set that are covered by the rule using a user-defined prototype function or it can be simply represented with a list of all matching examples in the training set. In the context of gene regulatory sequence and expression data analysis, the conditional part describes the regulatory sequence similarity of a group of genes, and the consequence of the rule describes their gene expression profile. Here, the goal is to identify groups of genes with same pattern(s) of sequence elements in the regulatory region that also have similar gene expression profiles.

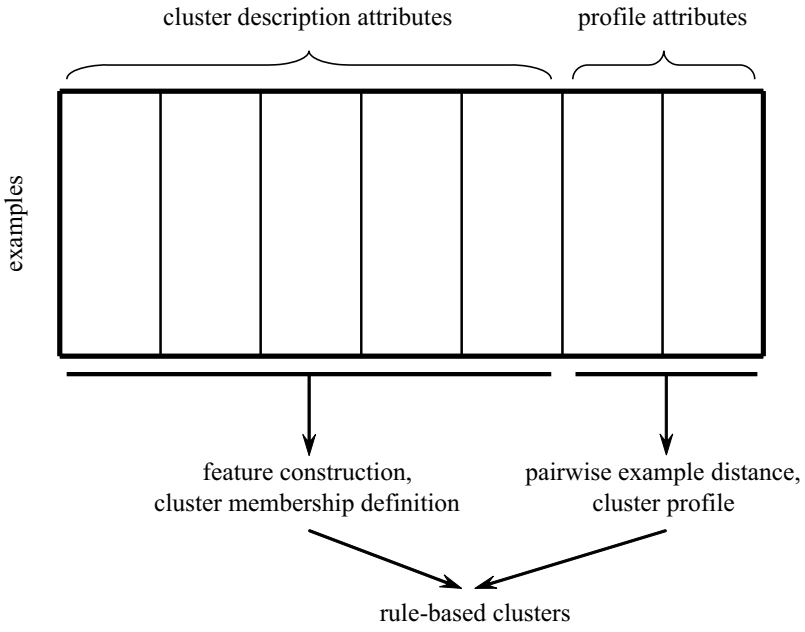


Figure 4.1: Two types of attributes needed for rule-based clustering.

Exhaustive search for rules, even with a relatively simple descriptive language, can quickly grow into a prohibitively hard problem due to combinatorial explosion. The main distinctive feature of the proposed rule-based clustering method is its ability to efficiently derive rules describing complex patterns that may include three or more features by starting from a base set of thousands of features. This is achieved by the heuristic search, explained in more details in the next section. The search is guided by the information on example distance and only the most promising parts of the vast rule-space are searched. Each next step in the cluster discovery process is selected based on example similarity in the currently discovered groups. Further refinements of rules are performed on only those rules describing the most promising clusters.

With overlapping clusters, which can be discovered by rule-based clustering, we are likely to find examples that jointly appear in a number of clusters. Obviously, our approach would benefit from post-processing, as joint cluster membership of a subset of examples may indicate several shared properties that may be overlooked if we rely on disjoint cluster coverage. Moreover, such examples can be of special interest, because they may indicate cases that should be studied in many contexts of other examples. In the biological context of genes,

this might be an indication that a gene is independently regulated by two or more regulators, or that it can respond to different conditions. In Chapter 6, for example, we show one such example for yeast, where functional groups of genes can be described by many cluster-specific rules.

The proposed rule-inference method was inspired by the covering CN2 algorithm [Clark and Nibblet, 1989] for supervised machine learning, and the approach of clustering trees developed by Blockeel et al. [1998]. CN2 can infer rules that relate a description of examples to the value of a special attribute called a class. Extensions of CN2 were recently proposed that use example weighting, with which the algorithm can identify overlapping clusters, to a certain extent. Similarly to the method developed in this Thesis, clustering trees find clusters of similar examples, where similarity is defined by a distance function on a subset of attributes. However, clustering trees return non overlapping clusters. The proposed rule-based clustering method combines both approaches. It is able to identify overlapping clusters of similar examples and at the same time infer a symbolic description of each cluster.

Similar to CN2 our method uses beam search to find the best rule. The beam is a set of L currently best rules to be further refined (parameter of algorithm in Figure 4.2). In the originally proposed CN2 algorithm the value of L is normally set in the range from ten to twenty. The small beam size greatly limits the space of rules that will ever be explored but also makes the search procedure run in acceptable run times. When a currently best rule is found, examples covered by the best rule are removed from further consideration. The rule inference procedure is then restarted to search for rules that cover the remaining examples. In the example weighting version of CN2, covered examples are not removed, but their weights are decreased. A lower example weight makes it less likely to be reconsidered in subsequent steps of rule search. Crucially, this prevents CN2 from reusing those same examples and inferring the same best rule over and over again in the next iterations of rule search. In practice, it can still happen that the same best rule is identified in two or three consecutive iterations before the weights of covered examples decrease enough and other examples (and rules) start to be explored by the CN2 search algorithm. The proposed rule-based clustering method uses a larger beam than CN2 normally uses and it searches for new rules until refinements are possible and result in significantly more coherent (sub)clusters. No actual example coverage is considered and examples

weights are not adjusted in these search steps, since it would prevent exploring certain parts of the rule space.

At the same time and independently to our work a similar idea to predictive clustering was proposed by Ženko [2007]. They also extended the method of clustering (decision) trees, developed by Blockeel et al. [1998], to existing methods for learning rules. Namely, they use "example weighting" extensions of the CN2 covering algorithm, for which they propose an "error weighted covering" scheme – in every iteration of the covering algorithm, the weights of examples are changed proportionally to the prediction error that a newly added rule makes on those examples.

4.2 Heuristic rule search

The algorithm for rule-based clustering is shown in Figure 4.2. The two sets of attributes (shown in Figure 4.1), that are given in the input example table E and required by the algorithm, are defined indirectly by the two user-defined functions $fc(\ldots)$ and $d(\ldots)$. The input function $fc(\ldots)$ encodes the user-defined, and background-knowledge dependent, feature construction operators which are defined on the first set of attributes (*i.e.*, on the "cluster description attributes" in Figure 4.1). Function $genFeature(\ldots)$ uses these operators to construct new features that are subsequently used for rule refinement (algorithm, lines 6 and 7). Similarity between pairs of examples is calculated with a user-defined distance function $d(\ldots)$ on the second set of attributes (*i.e.*, on the "profile attributes" in Figure 4.1). Both sets of attributes guide the rule search. The descriptive language, determined by the first set of attributes and by the feature construction operators, determines which clusters of examples can be formed and explored, defining the "language bias." The search among possible cluster descriptions is guided by the calculated pairwise example distance; the method aims to obtain clusters of examples whose pairwise distance is as small as possible. While in all applications described in this Thesis the two sets of attributes do not overlap, this is not required. This is left to the user when defining the language for cluster description and distance function.

The method requires a set of "target" examples T. This is usually a small subset of examples that the user wants to cluster into subgroups and find rules describing their distinctive, cluster-specific features. This can prove useful when

the user is mainly interested in a subset of examples and wants to analyze them in the context of all other examples. If all examples are of equal interest to the user, then the target set T must include all examples. However, in the latter case the user should be aware that entire subspaces of rules and examples might be missed because of limited beam size.

Rule-based clustering uses beam search to infer rules (see algorithm in Figure 4.2). Beam B is an ordered list of inferred rules that should be considered for further refinement. The order of rules in the beam is determined by a score that is calculated with the function *potential*(...). The beam is limited to a maximum size of L top-scored rules. The main loop of beam search is given in lines 3-14 in algorithm. The loop stops when beam is empty. Initially, the beam includes only one rule "True" (line 1) which describes all examples. This description is subsequently refined as are any other rules that get added into the beam.

The call of function *pop*() in line 4 removes the currently best rule from beam and assigns it to variable R_b. Rule R_b is refined into many different rules R_n (line 7) by adding different conditions on a new feature M_k. New features are generated by calling function $genFeature(\ldots)$ in line 6 (see next section for details on feature generation).

Each newly inferred rule R_n is tested with a call (in line 8) of function $accept(R_n,\ R_b,\ T,\ N,\ d())$. The function returns true if all two criteria are met. First, the new rule R_n must cover at least N examples from the target set T. Second, the average pair-wise intra-cluster similarity of the newly formed cluster described by rule R_n must be significantly greater that the intra-cluster similarity of the cluster formed by the "parent" rule R_b. This last criterion does not say anything about the size of the newly formed cluster (*i.e.*, it does not mandate the newly formed clusters to be of smaller size). Depending on the descriptive language used, the newly formed cluster can include less or more examples than the original cluster, as long as the second criterion is met (*i.e.*, the descriptive language can be defined to allow new constraints to be added conjunctively, disjunctively, or both).

The significance of increase in intra-cluster similarity – more precisely, the decrease of variance of the intra-cluster pair-wise distance, calculated with function $d(\ldots)$ – is tested using the F-test statistic:

$$F = \frac{SS_B}{n_B - 1} / \frac{SS_N}{n_N - 1}$$

Algorithm *RBC*

Input:
E - example table
T – set of target examples
$fc(\ldots)$ - feature construction operators
$d(\ldots)$ - distance function
K – maximum size of list R of discovered rules
L – maximum size of list B (beam)
N – minimum number of target genes in cluster
P – significance level for decrease in cluster variance
D – maximum average intra-cluster distance
M – maximum average number of rules covering one example
Output:
R – ordered list of discovered K best rules

```
1   B ← [True]; B is an ordered list (beam) of L best rules for
    further refinement
2   R ← []; R is an ordered list of K best discovered rules
3   WHILE B ≠ [] DO
4           R_b ← B.pop()
5           refined ← False
6           FOR EACH M_k IN genFeature(E, fc(), R_b)
7                   R_n ← refine(R_b, M_k)
8                   IF accept(R_n, R_b, T, N, P, d()) THEN
9                           S ← potential(R_n, d())
10                          B.add(R_n, S)
11                          refined ← True
12                  S ← score(R_b)
13                  IF not refined and S <= D THEN
14                          R.add(R_b, S)
15  R ← selectMostGeneral(R)
16  R ← filterByCoverage(R, M)
17  return R
```

Figure 4.2: Rule-based clustering inference algorithm using beam search.

where SS_B and SS_N are sums of squared differences from the mean inside the cluster (*i.e.*, the intra-cluster variance) of examples covered by the parent rule R_B and examples inside the refined rule R_N, respectively. Values n_B and n_N are the total number of examples in each of the R_B and R_N clusters, respectively.

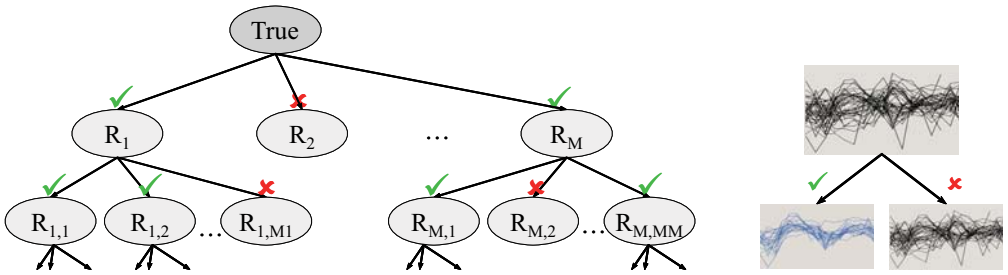


Figure 4.3: Rule refinement is the basic step of the proposed search algorithm. a) An example representation of explored rule-space. b) Change of intra-cluster variance after rule refinement determines whether the refined rule can be accepted.

A *pvalue* is calculated from the F score, and it is used to determine whether the increase in intra-cluster similarity is significant for a given significance level P (parameter given by user). Figure 4.3a shows how different refinements of same initial rule lead to the exploration of different clusters, each defined by a rule. Figure 4.3b shows how the intra-cluster similarity of examples covered by refined rules (cluster profiles in the bottom two nodes) must increase significantly (a significant increase is marked with a check-mark) compared to the cluster of the parent rule (top node). Only in such cases a refined rule can pass the test in function *accept*(...) and be added to the beam for further refinements, otherwise the rule is completely discarded from further consideration.

Refined rules that meet the two criteria are added into beam B (line 10) and are considered for further refinements. The position of each newly inferred rule R_n that is added into beam B is determined by the score S. This score is calculated with a call in line 9 of function $potential(R_n, d())$. Since the goal is to discover the most homogeneous clusters of examples, each rule in beam – which is waiting to be further refined – is not scored by the average intra-cluster similarity of all the examples covered by the rule, but by its potentially most homogeneous subset of examples. We call this heuristic measure the "potential" of a cluster, and it represents an approximation of the maximum intra-cluster similarity that can be achieved with further refinements of a given cluster. The score, or potential, is calculated in function *potential*(...) by taking $k \cdot N \cdot (k \cdot N - 1)/2$ shortest distances between examples in the cluster and calculating their average value. Parameter N is the number of minimal examples allowed in a

cluster, the default value of parameter k is normally set to 2. This scoring proved to greatly improve the selection of most promising rules for further refinement and consequently guiding rule search.

If a rule can not be further refined into any other rule that also passes the acceptance test (flag $refined$ in algorithm), it is then scored with function $score(\ldots)$. Function $score(R_b)$ returns the average intra-cluster distance of examples covered by rule R_b. If the score S is lower than the value of parameter D (parameter set by user), the rule is added into the final ordered list of rules R where only K best rules are kept (parameter K given by user). Rules in list R are ordered according to the calculated score, cluster with the lowest intra-cluster distance is first.

The final post-processing (lines 15 and 16) of rules found by beam search is performed by functions $selectMostGeneral(\ldots)$ and $filterByCoverage(\ldots)$. Details are given in Section 4.4.

Note, because the algorithm starts with all examples (*i.e.*, rule "True" covers all examples) the discovered rules can still cover examples outside the target set, even if the target set does not include those examples. The method can be applied to search for examples that were initially left out of a target set but should have been included based on their description and similarity.

Each cluster discovered by rule-based clustering can be, in principle, described by more than one rule. Relying on the assumption that clusters for which more cluster-specific descriptions can be found are more likely to be "true" clusters and not just a random result, this need to be considered when using rules to make predictions. Rules describing clusters, for which more cluster-specific rules have been found, should have a bigger contribution to the final prediction (weight) than those rules describing clusters for which only one or a few rules have been found. This can be achieved by treating each discovered rule separately. Matching rules, describing the same cluster, will have a bigger contribution to the final prediction. When using the inferred model to predict an example, all rules, that match the example, contribute equally to the final prediction. The final prediction is an "average" of conclusions of all matching rules. This "average" is formed using the same function needed to generate a prototype example from a set of examples.

4.3 On-the-fly feature construction

Description-rich problem domains can contain a large number of attributes or the given operators for feature construction can generate a large number of new features. In such cases, the total number of features to consider for rule refinement can be too big to be generated in advance, before rule search starts. Moreover, many such derived features are context dependent. They are defined only on a particular set of examples, and checking them on all other examples is a waste of computational resources. For these reasons, rule-based clustering uses an on-the-fly feature construction approach, which is seemingly incorporated in the search algorithm (see call of function $genFeature(\ldots)$ in line 6 of the rule-based clustering algorithm in Figure 4.2).

Function $genFeature(E, fc(), R_b)$ uses information on examples covered by the "parent" rule that has to be refined. By doing so, it limits the number of new features that will be used to refine the rule. The feature construction operators $fc()$ are thus invoked on only the subset (in the context) of examples from table E which are covered by the "parent" rule.

As already stated, rule-based clustering algorithm starts with a single rule in beam B, the conditional part of which reads "True" and the rule describes all examples. The initial rule is refined by adding conjunctive constrains from all single original attributes (*i.e.*, "True AND A" is logically equivalent to "A"). Feature construction is not "activated" in this case since it could generate a huge number of new features and render the search extremely slow. However, feature construction, along with the original attributes, is used to derive all subsequent rules. The existent conditional part of the rule can be extended by conjunctively or disjunctively adding new constraints. For example, the conditional part "$A_1 = v_1$" of a rule describing examples with value of attribute A_1 equal to v_1 can be refined into "$A_1 = v_1$ AND $A_2 = v_2$," requiring the value of an additional attribute A_2 to be equal to v_2. That same initial rule can be refined, using a disjunctive operator into "$(A_1 = v_1)$ OR $A_2 = v_2$." Using a feature construction operator, that same rule can be refined into "$A_1 = v_1$ AND $f(A_1, A_2) = v_2$." The new rule in this case requires an additional constraint on the relation between the two attributes A_1 and A_2, which is encoded by a feature construction operator $f(X, Y)$ as provided by the user.

When a new constraint is added disjunctively the current implementation of rule-based clustering, for practical reasons, encloses the conditional part of the

rule to be refined into parentheses. This of course greatly limits the descriptive language. The logical expression in the conditional part can only be a single, linear branch, if the expression would be shown as a tree. We have opted for this, because such prevailingly linear and degenerated trees greatly speed up and simplify the rule refinement steps. They are also easier to check when trying to match them to a set of examples. At the same time, allowing (some) disjunctive operations can still result in more informative rules.

4.4 Finding a subset of most general rules

After the beam search is completed, list R contains the top K discovered best rules. The rules are ordered by their score, with rules describing most coherent groups on top of the list. Even if size of list is limited by the user (with parameter K, set by user) it can still be potentially large. The two post-processing steps, $selectMostGeneral(\ldots)$ and $filterByCoverage(\ldots)$ invoked in lines 15 and 16 of algorithm in Figure 4.2, reduce the number of rules and, consequently, clusters returned by rule-based clustering.

Since beam search can find many descriptions for the same cluster of examples, the set of descriptions of a cluster has to be shrunk to include only the most general descriptions of each cluster. The algorithm $selectMostGeneral(R)$ (not given) groups all rules covering the same cluster of examples into separate sets. Each set S of rules is then processed by the function *selectMostGeneralInCluster*(S) that is given in Figure 4.4. The function starts by arbitrarily selecting a rule R from S (lines 2-4) and placing it in the final set of rules F. Then, each remaining rule R in S is compared to the current set of rules in F using an user-defined operator $general(T_1, T_2)$, described in details in the next paragraph. This is repeated until there are no rules in S (lines 5-18). If any of the rules already included in the set F is more general than current rule R, rule R is not added into F, as it is too specific (lines 10 and 13). Conversely, if current rule R is more general than any rule in F, then that rule is removed from F (line 15-17), and rule R is added into F. The final set of rules, returned by $selectMostGeneral(\ldots)$, is obtained by merging all rules in one ordered list, where rules are ordered by their scores.

Function $general(T_1, T_2)$ performs the basic step for generalizing a set of descriptions, *i.e.*, it compares two descriptions (conditional parts of rules): T_1

and T_2. The most general and usually also shorter description is preferred when comparing two descriptions. This heuristic is based on the widely used minimal description length heuristic, or Ocam's razor. For example, between two rules that describe the same cluster of examples, with conditional parts "A" and "B AND A" respectively, we would like to retain the former rule that requires only the simplest condition "A." It can also happen that rules "A AND B" and "B AND A" are discovered by the search procedure. In this case only one of the two logically equivalent rules is kept. Both examples are possible because of the nature of the search procedure, where the same final description can be inferred by following different branches of the search space. Thus, the user must define function $general(T_1, T_2)$ to return "True" if description T_1 is more general than T_2. If the two descriptions have no terms in common, $general(T_1, T_2)$ the function must return "False." Also, if the two descriptions are logically equivalent, the function must return "True".

The second post-processing step of rule-based clustering involves keeping a subset of best rules that does not report more than M rules on average for each example (parameter M is set by user) as this would overwhelm the user when exploring the results. Function $filterByCoverage(R, M)$ is called in line 16 of the rule-based clustering algorithm in Figure 4.2. The function starts by assigning zero cumulative coverage to all examples and it then traverses the ordered list of rules R. For each rule it checks the current average cumulative coverage of examples matched by the rule. If average cumulative coverage is less than parameter M, then the rule is selected and the examples' cumulative coverage is increased accordingly. This is done by adding one to cumulative coverage of all genes described by the rule. Otherwise the rule is discarded. This procedure selects the final list of best-ranked rules R and prevents reporting too many rules to the user. This list is returned by rule-based clustering in line 17 of algorithm in Figure 4.2.

Note, if we restricted the example cumulative coverage during rule search (*e.g.*, by using example coverage weighting decay as applied by CN2) we may prevent discovering some overlapping groups. The search would run the risk of not considering some (shared) examples in further rule search because of their quickly gained coverage by rules discovered in early steps of the search.

Algorithm *selectMostGeneralInCluster*

Input:
S – set of rules
function $general(T_1, T_2)$
Output:
F – subset of most general rules selected from set S

```
1   F =
2   R ← chooseRandom(S)
3   S ← S − {R}
4   F ← F ∪ {R}
5   WHILE S ≠ {} DO
6           R ← chooseRandom(S)
7           S ← S − R
8           tooSpecific ← False
9           FOR EACH E IN F
10                  IF general(E, R) THEN
11                          tooSpecific ← True
12                          break
13          IF tooSpecific THEN
14                  continue
15          FOR EACH E IN F
16                  IF general(R, E) THEN
17                          F ← F − {E}
18          F ← F ∪ {R}
19  return F
```

Figure 4.4: Algorithm for selecting the subset of most general rules describing a cluster of examples.

4.5 Space and time complexity of the algorithm

Search is the main task performed by rule-based clustering, thus contributing the majority of space and time complexity of the algorithm. The space complexity is measured as the maximum number of nodes that have to be stored in memory during search [Bratko, 2001]. This is determined by two parameters K and L given by the user (see algorithm of rule-based clustering in Figure 4.2). Parameter K determines the maximum number of rules that will be reported

to the user. Parameter L determines the size of the beam used during search performed by the rule-based clustering algorithm. The model generalization and rule selection steps performed after search (lines 15 and 16 in algorithm in Figure 4.2) can be done in-place, and thus do not require additional space.

Time complexity of a search algorithm measures the number of nodes generated during search [Bratko, 2001]. Time complexity of rule-based clustering algorithm is determined by the size of the beam used (parameter L), by the significance threshold (parameter P given by user) used to test the decrease in variance when refining rules, and by problem domain specific properties including the number of original attributes and their values (A) in the problem domain, average number of new features (F) that can be constructed with operators for feature construction on a subset of examples, number of target examples (N_t), and also number of all examples (N_{all}) in the problem-domain.

Since the overall complexity of an algorithm is domain-dependent, we provide upper bounds for the critical components of rule-based clustering, similarly as is done for the CN2 algorithm in the paper by Clark and Nibblet [1989]. Time complexity of the basic step of rule-based clustering is lower than the time complexity of the CN2 algorithm. While CN2 refines all rules in the current beam, rule-based clustering refines only the current best rule in beam. In this respect, rule-based clustering can be said to perform best-first search. Each rule can be refined in number of different ways ($A+F$). The difference in time complexities of the basic step between the two algorithms is exactly the length of the beam used by CN2. The time complexity of CN2 algorithm is $L \cdot (A + F)$, while time complexity of the basic step for rule-based clustering is $A + F$. However, in practice rule-based clustering has a longer execution time, because covered examples are not removed during rule search, but it continues until all rules in beam are refined.

The worst case for CN2 is that the best rule found at every step always describes a minimum number of examples. Let call this number M. The search is reiterated N_t/M times, until all target examples are covered. Let K be the average number of nodes explored by CN2 in each step. The total number of nodes searched is thus $K \cdot N_t/M$.

The worst case for rule-based clustering is that each refined rule covers one example less than the original rule. In such case, search has to be repeated $(N_t - M)$ times for each rule in beam, that is $(N_t - M) \cdot L$. If each rule can be

refined K/L times, then the total number of nodes searched is $(N_t - M) \cdot L \cdot K/L = (N_t - M) \cdot K$, which is in the same time complexity class as CN2 is.

4.6 Evaluation of models inferred with rule-based clustering

Cross-validation, leave-one-out, bootstrapping [Braga-Neto and Dougherty, 2004; Efron, 1983], and other standard machine learning evaluation methods are normally used to evaluate the predictive performance of a learning method [Mitchell, 1997]. Here we describe how we have used k-fold cross-validation to evaluate the rule-based clustering method. Initially, the data is split into k subsets. Cross-validation is performed in k steps. For each step a different set of $k-1$ subsets is used to learn (*i.e.*, to build) a model. The remaining one subset, not seen by the algorithm during learning, is then used to test and evaluate the model. Finally, results from all k folds are considered when reporting on the performance of the method either in terms of its classification accuracy, area under the ROC curve (AUC) or other measures for classifier performance.

Because the model returned by rule-based clustering is a set of rules describing the discovered clusters of examples, testing such model requires testing each rule on each test example from the given test set. Testing the model thus boils down to the basic operation of testing a rule on a (test) example. Distance, or the prediction error, between the rule's conclusion and the selected test example must be calculated first. The error is calculated depending on how the rule's conclusion is encoded. If the rule's conclusion is represented by a single, prototype example, then only one distance is calculated and considered, *i.e.*, the distance between the test example and the rule's prototype. In case the conditional part of the rule represents a set of examples, the distance is calculated as the average distance between the test example and each example in the rule's conclusion. The same distance function, given by the user for model inference, is used to calculate distance for evaluation purposes. If the tested example matches the conditional part of the rule, the calculated prediction error is added to the set of positive predictions. Otherwise, the calculated error is added to the set of negative predictions.

When all K steps of cross-validation are done and all inferred rules have been tested on all (test) examples, histograms of the positive and negative sets

can be drawn. Ideally, one would expect average zero error in the set of positive predictions, and average maximum error in the set of negative predictions. Such outcome would indicate that the inferred rules perfectly predict the matching test examples. At the same time it would also mean that all other examples, which were not matched by inferred rules, are completely different from the examples described by the rules. Reporting the observed distribution of distances (errors) between predicted and actual examples and its derived measures, such the average predicted error, can be used to drawn conclusions on the predictive ability of the method on a particular problem domain data set. The closer the two distributions are to the ideal distribution, the more we can trust the method's ability to discover rules with predictive value.

In this evaluation schema, clusters described by more rules get a bigger weight. There are more rules describing them and for this reason contributing more values to the two histograms of distance between predicted and actual example.

4.7 Summary and discussion

In this Chapter we have proposed a method for rule inference, called rule-based clustering. The method combines the classical CN2 rule-inference search procedure [Clark and Nibblet, 1989] and the method of clustering trees [Blockeel et al., 1998], and is able to identify overlapping subgroups of similar examples. Each identified cluster is described by a set of symbolic descriptions encoded in form of IF-THEN rules. This approach greatly differs from the standard cluster-first approach, where examples are first clustered based on their similarity, and then an attempt to infer the description of each cluster is made.

Example similarity is calculated with a user-defined function on a given subset of attributes. Another set of attributes and, optionally, operators for feature construction, are required for the inference of symbolic descriptions. Here, background knowledge on the problem domain, encoded in the feature construction operators and in the selection of attributes given by the user, can be crucial for successfully solving a specific problem.

The search done by the proposed method is performed using a large beam, and it is guided by a heuristic which prefers to refine rules describing clusters with higher potential to form even more coherent subclusters (and thus falls

under informed search). The proposed rule-based clustering methods can also incorporate an on-the-fly feature construction. Only new features, that are possible on the current subset of examples, are used to refine the current rule.

The model generated by rule-based clustering is a set of rules. The modeling done with rule-based clustering is both descriptive and predictive. Due to the symbolic language used to encode the conditional part of the rule, the user can gain new knowledge by studying the patterns found and presented with the inferred rules. By observing the number of rules found describing each cluster of genes, the user can decide on which discovered clusters should he focus in subsequent steps of the analysis. We also showed how to use the standard machine learning method of k-fold cross-validation to evaluate the predictive ability of inferred rules.

The rule-based clustering method requires a number of parameters, *i.e.*, length of the beam, significance level when testing decrease in variance after refinement of a cluster, *etc.* Future work includes finding ways to minimize the number of parameters, transforming them to make them more intuitive to the end user, and making the method more adaptive to different problem domains, with little user intervention by setting parameters. For example, the distance threshold parameter D given by the user could be replaced by a parameter limiting the percentage of most similar groups discovered by the method that get reported to the user. As search progresses, the method could keep track of the intra-distance of all groups discovered so far, and automatically set parameter D to keep only an arbitrarily selected percentage (set by user) of all discovered clusters. Another, more technical aspect of future work is to use a full conjunctive and disjunctive language, whereas currently disjunctively terms are handled in a limited way (see last paragraph in Section 4.3).

Chapter 5

Interpretation through visualization

Using proper visualization can greatly influence and augment the way one explores and studies the inferred models and underlying data. Good visualization is crucial when a rich descriptive language is used for describing discovered patterns, and where complex models are inferred from the data. This is the case with rule-based clustering which is able to discover a large number of descriptions (rules) covering overlapping clusters of examples. A good visualization of discovered patterns and clusters of examples may allow the user to discover high-order structure and observe other properties of inferred patterns and underlying data.

To aid in the explorative data analysis [Tukey, 1977] supported by rule-based clustering, we have developed three straightforward but nonetheless useful types of visualization, each emphasizing a different aspect of the underlying structure of the model. Used together, they can provide for a better insight into the common structural features and properties of discovered rules and example clusters.

The three proposed visualizations are rendered in the form of a graph. Depending on what aspect of the rule-based clustering model they visualize, we refer to them as example, cluster-and-rule, and feature networks.

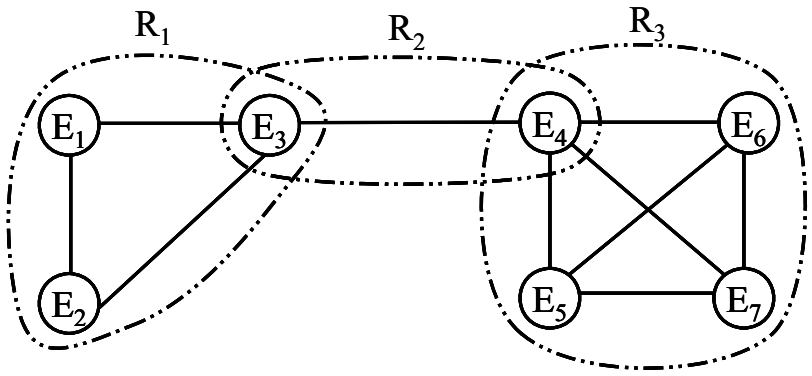


Figure 5.1: Example network. Nodes represent examples. Two nodes are connected if both are described by same rule.

5.1 Example network

Visualizing the graph of an example network is the simplest and intuitive way to present the rule-based clusters (see Figure 5.1). All rules constituting the model are visualized. In this representation nodes represent examples. Two nodes are connected with an edge if the two corresponding examples are covered by the same rule. The weight of an edge can be determined by the calculated distance between the two examples connected by the edge. The weight of edges can also be determined by the number of rules that support the connection (*i.e.*, number of rules that cover the two nodes).

Visualizing many highly overlapping groups can quickly render this visualization saturated. By showing only the edges above (or below) a user selected weight threshold, the user can visually explore both sides of the node connectivity spectrum, *i.e.*, nodes that are forming tight clusters, and nodes that are the least connected. Both cases can provide insight for further study.

5.2 Cluster and rule networks

The next level of abstraction is a graph of example clusters (see Figure 5.2). Here, nodes represent clusters of examples. In this visualization, two nodes are connected with an edge if at least N_F examples appear in both clusters (examples are said to be shared by the two clusters). The parameter N_F is set by the user. The size of the node can indicate the size of the cluster. The

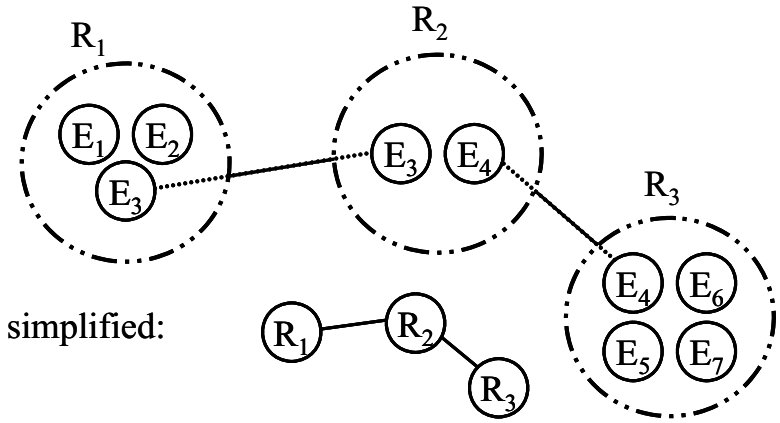


Figure 5.2: Cluster (and rule) network. Nodes represent groups of examples that belong to the same cluster (*i.e.*, are described by same rules). Two nodes are connected if an arbitrary number of examples are shared.

thickness of lines representing edges in the graph can be proportional to the number of examples shared by the two clusters (nodes). This visualization is useful for easy identification and exploration of examples that are shared by the discovered clusters. By varying the threshold, one can observe how the initial clustering of examples breaks into less connected subgroups. Our hypothesis is that such graph can further reveal the structure of examples in the data, and that examples belonging to the same subgroup, identified in this network, may be related. This can extend the possible group membership beyond clusters identified by a single rule. Clusters, for which many descriptions (rules) exist, will be highly connected among themselves, since they will be sharing many examples.

Because rule-based clustering can return many descriptions (rules) for the same cluster of examples, it is usually more useful to visualize only a subset of all rules representing the same cluster, *e.g.*, by showing only top K shortest rules describing the cluster (parameter K given by user).

5.3 Feature network

The last level of abstraction is a feature network. Here, nodes represent terms (or features) forming the individual rules, *i.e.*, parts of rule patterns that impose constraints on the object description (see Figure 5.3). Two nodes are connected

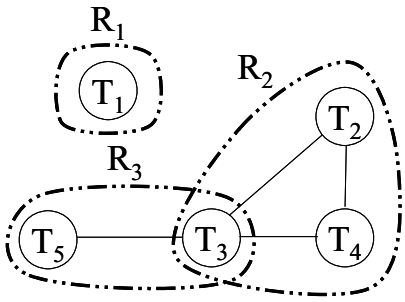


Figure 5.3: Feature network. Nodes represent parts of the conditional part of rules. Two nodes are connected is both terms are appearing in the conditional part of the same rule.

if they appear in conditional part of the same rule.

While Figure 5.1 and 5.2 represent the same three rules and some one can observe some cluster overlap of examples, they do not provide any information about the composition of the rules's conditional parts and reasons for cluster overlaps. Let's say that rule "$R_1 = (M_1 == v_1)$" requires the value of feature M_1 to be v_1, rule "$R_2 = (M_2 == v_2)$ and $(M_3 >= v_3)$ and $(M_4 <= v_4)$" determines the values of the three features, and rule "$R_3 = (f(M_3, M_5) == True)$" requires a function between the two features to hold true. Example coverage of rules is shown in graphs in Figure 5.1 and Figure 5.2. Only by showing the feature network visualization in Figure 5.3 one can explore the descriptive details about the discovered clusters of examples. Feature network can thus be used to identify common and also rule-specific terms or features appearing in discovered rules, leading to the identification of potentially general and also group-specific features.

5.4 User interface design

Explorative data analysis [Tukey, 1977] requires software tools that allow interactive exploration of discovered data patterns. Linking the discovered patterns to existent knowledge, which is stored in readily available databases and web pages, can further support the user when exploring the results and deciding on further steps of the analysis based on the newly gained knowledge from the discovered data patterns.

While software environments for interactive data analysis, like the machine

learning and data mining suit Orange [Demsar et al., 2004a,b], with extensions for functional genomics [Curk et al., 2005a], offer great flexibility in customizing the analysis workflow, their on-the-fly visualization approach (visualization when the results of analysis are available) also require every step to be computed relatively quickly. Computationally intensive methods that require longer running times, as is the case for rule-based clustering method, should be run in a non-interactive way. However, when results are ready, the user must be able to explore them interactively.

Presenting results in the form of web pages has proven user-friendly and does not require installation of any dedicated software tool. In particular within the field of bioinformatics, web pages that summarize the results of analysis have become a standard way of communicating analyses results to researchers. Web pages are easy to generate and offer enough richness of expression to present results in an easy-to-understand graphic form. They also allow simple integration and intuitive linking to knowledge and databases published on the internet, which can provide additional support for reported results.

We have designed a guided, domain-independent interface to present results of rule-based clustering that uses all three types of visualizations. The presentation can be augmented by providing problem-domain specific visualizations, making it more intuitive and informative. For example, in the analysis of gene regulatory regions, we have used visualization of gene expression profiles and visualizations of gene regulatory regions, which are described by the conditional parts of rules.

The main web page, with results from a rule-based clustering analysis, starts with the list of target examples given by the user. Examples, covered by the inferred rules, are visualized in a parallel plot or some problem domain-specific visualization, showing only attributes on which distance between examples was calculated. This visualization offers an initial overview of the actual similarity of covered examples. Covered target and covered non target examples are then listed. Target examples, for which no description was inferred, are listed separately. Whenever listing examples, their symbolic name, if available as a meta-attribute, should be listed instead of a unique id (*e.g.*, symbolic names are preferred over I.D.s when listing genes). Each listed example should also be linked to a separate page containing details about the example.

Next, the list of discovered clusters is given in a table. Each row represents

one cluster. The number of inferred rules for the cluster, a score of the cluster average intra-distance and its variance, and a list of covered examples are given. Links to subsequent pages with details for each rule describing the cluster, and links to details on covered examples are also given.

The three graphs (cluster, feature and example networks) are then displayed. Nodes in the graphs, representing clusters, features or examples are linked to subsequent pages with corresponding details, briefly described next. These visualizations allow the user to visually identify clusters or examples of interest and, by following the associated links, to learn the details of each element.

The page with details on an example contains all information on the example. The page links to discovered clusters that include the example. The page also links to any additional supporting online information or databases.

The page with details on a discovered cluster lists all examples forming the cluster. Examples are display in a parallel plot or other problem-domain specific visualizations, similarly to the visualization on the main page. The page should also include the list of inferred rules describing the cluster. Problem domain-specific visualizations can be used to display the descriptions encoded by the rules. Other, cluster specific information can be displayed and links to online supporting pages and databases can be provided. For example, in the analysis of gene regulatory regions, links to the Gene Ontology Term Finder tool can be given which allow the user to further analyze the cluster of genes and identify cluster-specific annotation.

The page with details on features contains information on the feature, and links to supporting online material. The page should also list and allow the user to visit pages with detail on all rules and clusters containing the feature.

For an example web page with all elements described above, see the collection of analyses of gene regulatory regions, available at this web page: http://bubble.fri.uni-lj.si/dicty/index.html. See Section 6.6 for an example where all three visualizations are used in order to discover patterns in data.

5.5 Summary and discussion

The analysis and modeling of complex problems require proper presentation and visualization of computationally discovered data patterns. This can greatly support the user's data exploration, understanding, and gaining of new knowledge.

With proper visualization, the user can quickly connect pieces of new knowledge into a bigger picture. The discovered data patterns, deemed important by the user, offer new hypotheses to test. Although the proposed visualizations are relatively trivial, they should provide a good presentation of discovered complex data patterns, and allow the user to identify any higher structure, beyond the one present in individually reported clusters. In Chapter 6 we show a few such examples.

The media used to present results also play a crucial role in explorative data analysis. Presenting the results in form of web pages, with links to supporting data, knowledge and other online tools, has proved to be user-friendly and extremely valuable when modeling complex and description-rich problem domains.

Given a potentially high number of discovered clusters and descriptions, it is vital to offer an interactive and guided exploration of results. The proposed interface and presentation of results provides the user with an initial overview of the results, and it also allows the user to zoom in and learn all the details on a selected subset of examples.

Chapter 6

Experimental applications of rule-based clustering

Although the rule-based clustering method presented in this work can be applied to many various problem domains, we here focus on its application in bioinformatics and analysis of gene regulatory regions. The goal of the analysis is to cluster genes into subgroups and infer a symbolic description of each cluster that relates gene expression under various experimental conditions to the structure of regulatory regions.

We performed our analyses on the data on two different model organisms: budding yeast *Saccharomyces cerevisiae* and slime mold *Dictyostelium discoideum*. We used cross-validation to empirically evaluate the predictive performance of rule-based clustering on this problem domain, and estimated the statistical significance of inferred models. Using the inference of rule-based clusters and computational approaches for their evaluation, we can attempt to answer the following important biological questions:

- What is the nature of gene regulation? Is it combinatorial (*i.e.*, does it require combinations of a small number of transcription factors) or highly specialized (*i.e.*, does it require a large number of highly specialized factors)? We can try to answer this question by testing and comparing the predictive ability of models inferred using descriptive languages of varying complexity.

- What part of the promoter region bears the highest correlation with expression? Does the gene's coding region include any information that can

be used to better predict gene expression? We show how to computationally determine the regulatory region that includes the most information on gene expression.

- Are there other mechanisms that regulate gene expression? An example is the secondary structure of promoter regions which could be included into modeling with rule-based clustering.

Focusing on applying rule-based clustering for the analysis of gene regulation patterns, we start by introducing a hypothesis language for describing gene regulation regions (which is encoded in the conditional part of a rule) and the gene expression data used for calculating cluster similarities. The computational analysis and evaluation of the predictive ability of inferred model are given next.

6.1 Descriptive language for structure of gene regulatory region

First, we need to formalize a rich descriptive language that can be used to describe the structure of regulatory regions. The proposed language can be used to describe the presence of putative binding sites, place limits on distance of a putative binding site from transcription and translation start site (ATG) and other landmarks, define distance and the relative and absolute orientation of a putative binding site relative to a given reference point in the gene. All elements in the proposed descriptive language are based either on known examples of experimentally confirmed regulatory structures or are hypothesized and described in biological textbooks [Alberts et al., 1994; Latchman, 1998]. See Figure 6.1 for a schematic representation of descriptive elements of a regulatory region.

The assertions on the structure of gene regulatory regions are composed of terms. Test of presence of a known or putative binding site (*e.g.*, site S_1) is noted by a term "S_1." The test can also include the constraint on orientation of the site. Notations "S_1+" or "S_1-" are used for positive (sense) or negative (non-sense) orientation of the binding site (S_1 in this example) relative to the reading direction, respectively.

Number of occurrences of a binding site is stated as "$\#(S_1)$ *op* N_1," where *op* can be any of the standard mathematical operations: equals, less than, less or

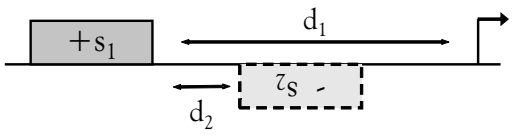


Figure 6.1: Elements of the descriptive language used to describe the gene regulatory region. With the proposed language one can express constraints on distance (d_2) between two binding sites (S_1 and S_2), distance from ATG (d_1) and orientation of binding sites in the sense (S_1+) or non-sense direction (S_2-) relative to the reading direction of the gene.

equal than, greater than, greater or equal than. The position of the binding site S_1+ relative to a landmark M can be stated as "S_1+ @$p_1..p_2(ref{:}M)$." Values p_1 and p_2 denote the distance interval, measured in nucleotides, and reference site M can be either a binding site (*e.g.*, S_2 or S_2+), start of translation (ATG), or start of transcription (TSS).

Single terms can be combined either conjunctively or disjunctively to form more complex descriptions of the structure of the regulatory region encoded by the conditional part of a rule. For example, requiring the presence of two binding sites, where the position of the first binding site is relative to ATG and the position of the second binding site is relative to the first, would be stated as "S_1@$p_{11}..p_{12}(ref{:}ATG)$ AND S_2 @$p_{21}..p_{22}(ref{:}S_1)$."

A disjunctive constraint on the presence of either of two binding sites is stated as, for example "S_1 OR S_2." Negation of terms is also allowed. For example, to forbid the presence of binding site S_3 in the regulatory region, one would state "$not(S_3)$."

6.2 Genomic data

For the analysis of budding yeast *S. cerevisiae* we have used publicly available data at SGD (Saccharomyces Genome Database, see www.yeastgenome.org). The curated collection of transcription factor binding data, available at SGD, comes from the study by Harbison et al. [2004], where they used genome-wide location analysis, phylogenetic analysis of conserved sequences, and prior knowledge to identify sequence elements that are bound by regulators under various conditions and that are also conserved among *Saccharomyces* species. In some examples we have also used transcription factor data from a previous study done

by Lee et al. [2002]. The collection of gene expression "Expression Connection" at SGD includes some of the most known published and publicly available microarray assays performed on yeast. We show example of analyses done on data sets on peroxisome assembly and function studied by Smith et al. [2002], environmental stress and starvation studied by Gasch and Werner-Washburne [2002], and cell cycle studied by Cho et al. [1998].

Genome sequence data for slime mold *D. discoideum* was downloaded from the organism's web page at http://dictybase.org [Eichinger et al., 2005]. Gene expression data was in part obtained from publicly available and published sources [Van Driessche et al., 2005; Van Driessche, 2004; Van Driessche et al., 2002; Booth et al., 2005], and also directly from our collaborators at Baylor College of Medicine in Houston, Texas. The data that we analyzed consists of fifteen microarray measurement assays of 4081 genes in wild type and in fourteen mutants, where one or two genes were deleted (*i.e.*, single and double mutants). The assays were done in a different number of biological and technical replications. The fourteen mutants used are: *acaA*-, *acaA*- *pkaC*+, *comA*-, *comB*-, *comC*-, *pkaC*-, *pkaR*-, *pkaR*- *regA*-, *pufA*-, *pufA*- *pkaC*-, *pufA*- *pkaR*-, *regA*-, *yakA*-, *yakA*- *pufA*-, where the minus sign indicates a deletion mutant, and the plus sign indicates genes induced to over express. We report on analyses done for each assay separately. We also provide results from the analysis where weighted average gene expression data from all assays was used.

6.3 Empirical evaluation of rule-based clustering

To evaluate the predictive performance of rule-based clustering for the analysis of gene regulatory regions, we have performed five-fold cross-validation as described in Section 4.6. For transcription factor binding site data we have used the data published by Harbison *et al.* on 102 transcription factors binding to 1749 genes [Harbison et al., 2004]. We have used gene expression data from nineteen microarray assays available at SGD's expression connection (ftp://genome-ftp.stanford.edu/pub/yeast/data_download/systematic_results/expression_data/expression_connection_data). A distance matrix of pair-wise Pearson correlation of gene expression was calculated on microarray data from each study. The final distance matrix, used in the evaluation of rule-based clustering, was calculated

as the average of distance matrixes over all studies. The evaluation was performed on the set of 1749 genes with measured gene expression and which are also known to be bound by at least one transcription factor (as given in the data from Harbison *et al.*). We modeled the regulatory region spanning from positions -600b to 0b relative to translation start site (ATG).

All genes were considered to be target genes by the rule-based clustering. All genes were randomly split into five subsets that were used for cross-validation. The rule-based clustering method was run five times. Each time, different four subsets of genes were used as the learning set, on which rule-based clustering identified clusters and inferred a model (set of rules). The remaining one subset was used to test the model, as described in Chapter 4.6. Each inferred rule (in the model inferred from the learning set) was then tested on all test genes.

The results of testing are summarized in two histograms. One histogram shows the distribution of distances (errors) between predicted and actual gene expression for the test genes that match the rules. The second histogram shows the distribution of distances (errors) between predicted and actual gene expression when a rule does not match the test gene. The histograms were obtained by first calculating the distance ($1.0 - Pearson\ correlation$) between the gene expression predicted by the rule and the actual expression of the tested gene (prediction error). The calculated error value was added into the histogram of matching or into the histogram of non-matching rules, depending whether the tested rule matched the test gene or not. The two histograms obtained on yeast data, accumulated from all five folds of cross-validation, are plotted in Figure 6.2 and Figure 6.3.

Ideally, the distribution of error (distance) for matching rules (Figure 6.2) should be a single bar at distance zero. Similarly, the distribution of non-matching rules (Figure 6.3) should peak at maximum error (near value two, for the distance function we have used), as explained in Chapter 4.6. Calculating the average error and standard deviation can be used to describe and summarize the predictive ability of the inferred model. In practice, as is the case on yeast data, a normal distribution of the prediction error around a center is observed. This is due to many facts, including the incomplete and noisy gene expression (relative mRNA abundance) data, incomplete and noisy transcription factor binding site data, which can be either measured with DNA microarrays or computationally inferred from promoter sequence data. In the

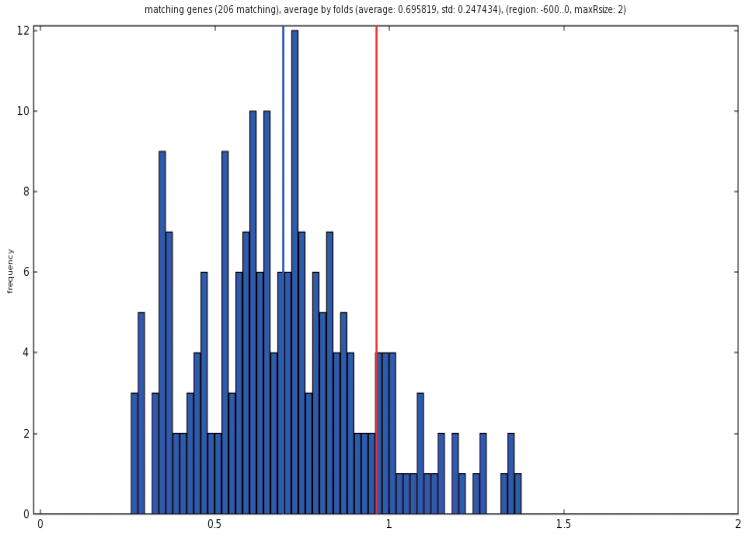


Figure 6.2: Distribution of prediction error for 206 matching test genes. Distances between gene expression predicted by a rule and actual gene expression of tested genes are shown. Only prediction errors on test genes that match the rule are shown. Blue vertical line indicates the average error. Red vertical line indicates the average distance among all pairs of genes.

former case, the noise is due to the experimental setup, in the latter it is due to the search algorithm and modeling formalism applied to model the putative transcription factor binding sites. The data on *S. cerevisiae*, which was prepared in the study by Harbison et al. [2004], and used for this analysis is far from complete. It includes data on binding sites for only 102 transcription factors out of an estimated total 203 transcription factors, with those 102 transcription factors binding to the promoter region of $\sim$ 1700 target genes out of a total $\sim$ 6000 genes identified in the yeast genome. The observed error of the proposed model is also due to the incompleteness of the descriptive language used in this case. The language does not include chromosomal location of genes (entire regions of the chromosome can be silenced), it ignores the presence and proximity of enhancer and silencer elements, it does not consider the stability of mRNA, and many other biologically important aspects, but for which there is a lack of experimental data. Some of these issues are discussed in the conclusion of this Chapter (see Section 6.9). The properties of the clustering method

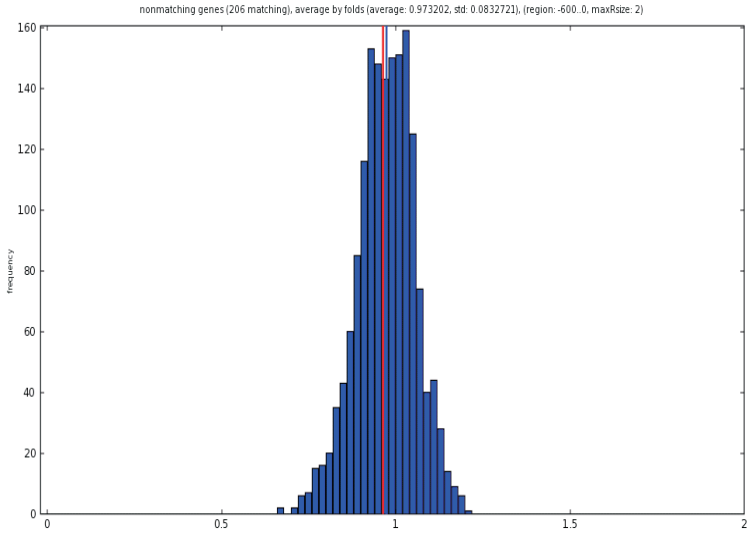


Figure 6.3: Distribution of prediction error for non-matching test genes. Distances between gene expression predicted by a rule and actual gene expression of tested genes are shown. Only prediction errors on test genes that do not match the rule are shown. Blue vertical line indicates the average value of the error distribution. Red vertical line indicates the average distance among all pairs of genes.

itself, mainly the method's ability to discover relatively small and overlapping clusters of genes, also contribute to the problem to some extent. Rules describing overlapping clusters should not contribute much to the error distribution of matching test genes. In such cases, when many overlapping clusters are discovered and the rules describing those clusters all equally well predict unseen genes, the error should be minimal and distribution for matching test genes skewed towards zero. However, those same rules will greatly determine and shift the distribution of non-matching test genes to lower values than the expected value of two (for the distance function we have used), making the distribution appear similar to the distribution of all pair-wise distances.

Another basic, but important metric is gene coverage, *i.e.*, the number of genes that are matched (predicted) by at least one rule. In this empirical evaluation, rule-based clustering was able to identify rules that well predicted the expression of ~ 210 genes (out of 1749 used, or $\sim 12\%$).

6.4 Comparison to distance distribution in a random model

The significance of an inferred model can be assessed by comparing it to a random model. For each inferred rule in the model, a random set of test genes is selected. The randomly drawn set is of same size as the set of test genes which truly matches the rule. This random set of genes from the test set (test genes) can be said to match a random rule. The random rule still describes the same set of genes from the learn set of genes, but predicts other (random) test genes. We can measure the prediction error (or distance) on the random test sets, and generate a histogram similar as the one for the truly matching test genes. Repeating this procedure a number of times for each rule produces an estimation of distribution of prediction error of a random model with same characteristics as the actually inferred model (*i.e.*, this way the coverage of rules is preserved). The significance of the inferred model can be then determined by comparing the error distribution of the inferred model to the distribution of a random model. Figure 6.4 shows the distribution of prediction error on the yeast data set we used for evaluation of the method. Comparing the distributions in Figure 6.2 (inferred model) and Figure 6.4 (random model) clearly shows that the model predicted with rule-based clustering is far from random. Figure 6.5 shows the distribution of pair-wise distances for all possible pairs of genes. This is the distribution of pair-wise distance one can expect to observe in a set of randomly drawn genes.

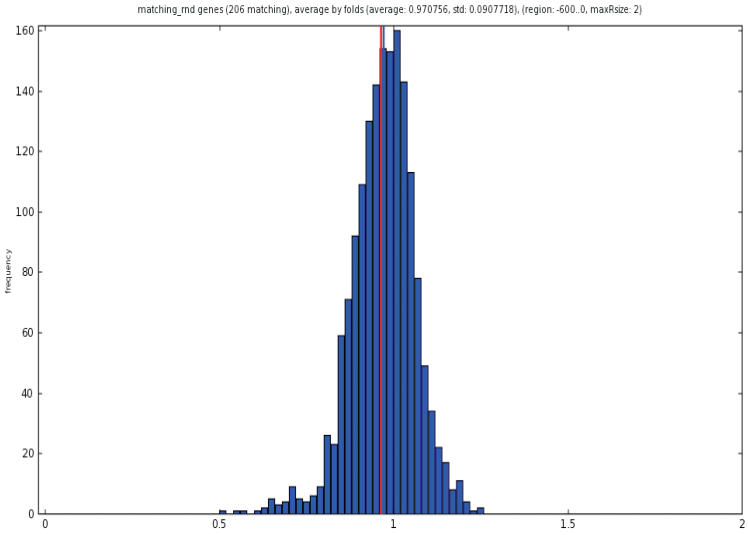


Figure 6.4: Distribution of prediction error of a random model. The random model was generated to have similar characteristic to the true model from Figure 6.2. Red vertical line indicates the average distance among all pairs of genes.

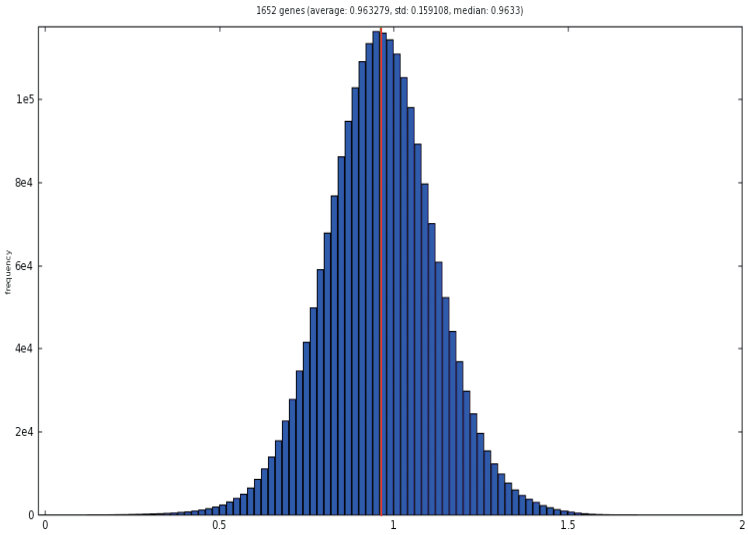


Figure 6.5: Distribution of all pair-wise distances in gene expression. Red vertical line indicates the average distance among all pairs of genes.

6.5 Influence of descriptive language on the predictive power

We have evaluated the various elements forming the descriptive language used to model the regulatory region for prediction of gene expression. We report on how each language element affects and contributes to the predictive ability and gene coverage of the inferred models. Individual elements and combinations of descriptive elements used to infer the best predictive models can shed some light on the nature of gene expression regulation. They provide an indication on the number of transcription factor binding sites that are required, and on the relations that hold among the binding sites.

We have used the same transcription factor binding site data and gene expression distance matrix of Pearson correlation that we have used for the evaluation of rule-based clustering, described in the previous section. We have varied the complexity of the descriptive language, each time running five-fold cross-validation and reporting the observed average correlation (average prediction error) between gene expression predicted by the model and actual gene expression for matching rules. We also report the predictive gene coverage achieved by the model, *i.e.*, the number of genes for which gene expression could be predicted. We have tested, in various combinations, the following elements of the descriptive language:

- test of presence of a binding site,
- orientation of a binding site,
- number of occurrences of a binding site in the promoter region,
- relative distance of a binding site relative to translation start site (ATG),
- relative distance of a binding site relative to another binding site,
- conjunction of terms,
- disjunction of terms,
- negation of terms.

By comparing the predictive ability of each language element, and the combinations thereof, we were able to determine which elements most contribute to

the predictive ability and coverage of inferred models. See Table 6.1 and Figure 6.6 for results. Figure 6.6 shows that the average prediction error is around 0.7 (*i.e.*, average Pearson correlation of 0.3) and it varies very little depending on the descriptive language used. However, coverage – the number of predicted genes – even though still relatively low compared to the total number of modeled genes (see Section 6.3. for details on the incompleteness of transcription factor binding site data), changes dramatically based on the kind of language elements used to model. Using only elements with which presence, orientation or number of occurrences of individual binding sites can be described results in the poorest prediction ability. Less than forty genes can be predicted. The distance of binding sites from the translation start site (ATG, referred to as "ATG dist") is the single most informative element. By using this descriptive element of the promoter regulatory region alone, gene expression of 200 genes can be predicted. Distance between pairs of binding sites (site pair distance) is the second most important element with which rules covering approximately 100 genes can be inferred. Using information about orientation of binding sites, in combination with other features, improves the model's ability to predict more genes at a slight expense in average error, for all of the cases. Also, using information on the number of binding site occurrences improves the model in all cases. Using all elements in combination (*i.e.*, orient, count, pair dist. and ATG dist., see Figure 6.6) gives the highest coverage of ~ 250 genes. From these results, one can conclude that distance of transcription factor binding site to ATG is, at least in yeast, the single most informative descriptive element which should be used when predicting gene expression from the content of the regulatory region. This finding is in agreement with many reports on the non-uniform positional distribution of transcription factors binding sites relative to ATG for many species. For example, Down et al. [2007] used a purely computational, statistical motif discovery approach NestedMICA and observed a non-uniform distribution of discovered motifs (*i.e.*, putative transcription factor binding sites) relative to ATG in *D. melanogaster*. They observed $\sim 60\%$ of motifs (70 out of 120 motifs) having a peak in the first 400b upstream of ATG. Harbison et al. [2004] used genome-wide location analysis data, data on phylogenetically conserved sequence and other published evidence on *S. cerevisiae* to augment the motif discovery algorithm for prediction of transcription factor binding sites. They also report on a sharply peaked distribution of binding sites around position -

200b relative to ATG, with the majority of the transcriptional regulator binding sites lying between -500b and -100b relative to ATG. Other examples, just to name a few, include the strong positional bias of E-box in *D. melanogaster* close to ATG [Hulf et al., 2005], the requirement of MSE core promoter sequence elements in *S. cerevisiae* to reside in the region -300b to -75b relative to ATG for specific genes to be induced [Jolly et al., 2005], and motifs in the promoters of chemosensory receptor genes in *C. elegans* [McCarroll et al., 2005]. Some of these studies (*e.g.*, the study by Jolly et al. [2005]) list examples where the position of a motif relative to ATG is important for function, and report on observations where if same motif is positioned outside a specific promoter region relative to ATG, then it has no or little influence on gene expression. In the next Section 6.6 we show that most of the regulatory information is encoded by motifs residing in the regulatory region between -300b to 0b relative to ATG.

To determine how size of the inferred rules influences the predictive ability, we have run rule-based clustering three times, each time allowing a different maximum size of inferred rules. Rules could be formed from only one, two or three terms. Figure 6.6 show results obtained with the same descriptive language, but with varying maximum rule size, are connected by a line. Inferring longer rules results in slightly lower prediction error, but at the cost of lower number of genes being predicted (see Table 6.1 for details). Longer rules, if formed by conjunctively added terms, as it is in the case reported here, tend to cover fewer genes than shorter rules. Namely, the algorithm requires that each added term (*i.e.*, rule refinement) results in a rule that describes a more coherent group of genes than the group covered by the original rule (group coherence is measured based on gene expression). Consequently, smaller groups of genes tend to be more coherent than larger groups, which also diminish the average prediction error when comparing the measured and predicted gene expression.

The small effect of the length of inferred rules on the model's predictive ability is an indication that gene expression regulation is not highly combinatorial, at least for the yeast genes that could be predicted. On average, information on 1.3 transcription factor binding sites in a promoter region is needed to predict gene expression of the gene. Also, on average, data on 26 TF binding sites, appearing in 30 different combinations with average length of combination 1.17 (median is 1.0), is needed to predict the expression of ~ 230 genes. These results suggest that regulation requires specialized transcription factors.

Finally, either allowing terms to be disjunctively added or using negation of terms did not improve the predictive ability of inferred models (results not shown). The model returned by rule-based clustering implicitly encodes disjunction, because it is treated as a collection of disjunctively joined rules when used for prediction. This can in part explain why explicitly allowing disjunction for rule refinement does not improve the model further. When testing different variations of the descriptive language that allow negation of terms, the inferred models contained few rules formed with negated terms (on average, $\sim 2\%$ of all the rules forming a model). Overall, negation of terms does not greatly influence the predictive ability of inferred model. Both two elements of the descriptive language only make the search run longer and do not drastically improve the performance of inferred models. Disjunctively added terms, when added to a rule, can only increase gene coverage and thus prevent the monotonous shrinking of discovered clusters, making the search run longer than when only conjunctively added terms are allowed. Negated terms usually describe large groups of genes (*e.g.*, in our case, if there are hundred genes with a motif present in their promoter region, then the negation of the motif's presence covers all other ~ 1600 genes) which consequently requires more computational time when those terms are used for rule refinement.

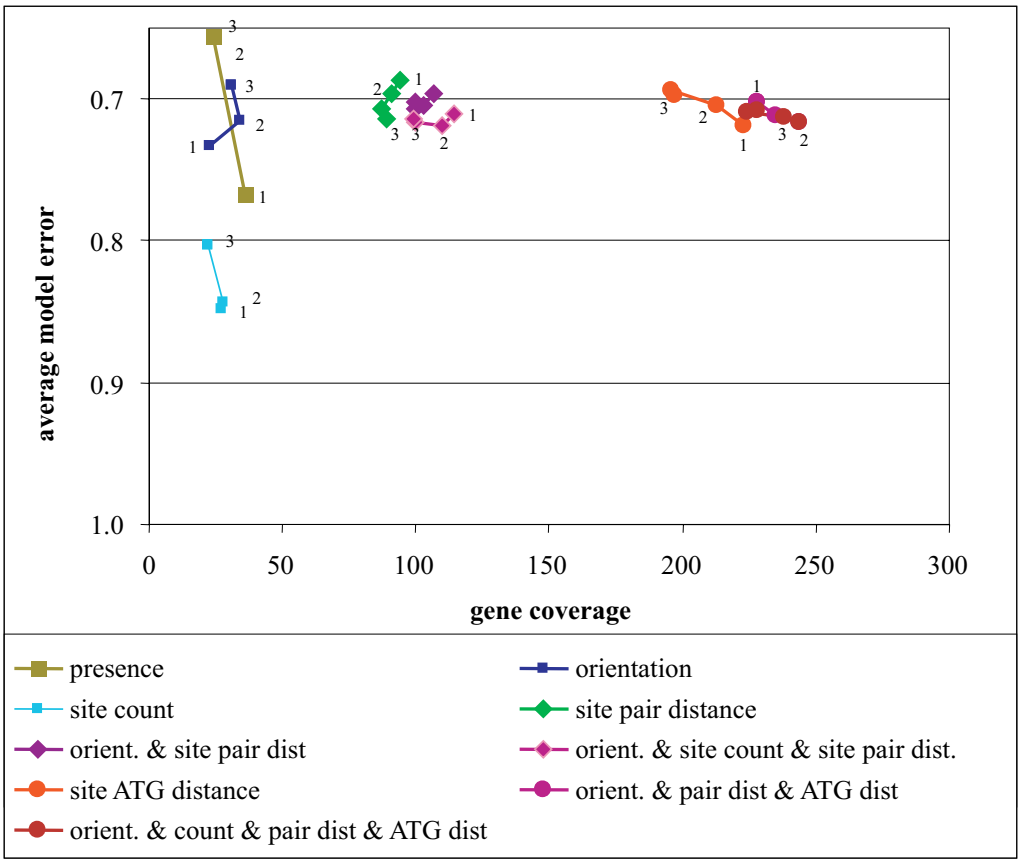


Figure 6.6: Average prediction error and gene coverage for different combinations of descriptive elements used to model the regulatory region. Number of predicted test genes (coverage) and average prediction error (= 1.0 − *Pearson correlation*) achieved for each case are shown. Results obtained for different maximum allowed size (indicated by numbers 1,2,3) of inferred rules are connected by a line.

max. rule size	AND	OR	NOT	site presence	site orientation	site count	site pair distance	ATG distance	mean matching error	std	coverage	mean nonmatching error	std	mean random rule error	std
1	True	False	False	True	True	False	True	True	0.71645	0.236	244	0.97130	0.076	0.97207	0.102
2	True	False	False	True	True	False	True	True	0.70135	0.241	228	0.97179	0.082	0.97078	0.099
3	True	False	False	True	True	False	True	True	0.70082	0.240	227	0.97165	0.082	0.97084	0.096
1	True	False	False	True	True	False	False	True	0.72002	0.236	224	0.97295	0.081	0.97214	0.105
2	True	False	False	True	True	False	False	True	0.69582	0.247	206	0.97320	0.083	0.97076	0.091
3	True	False	False	True	True	False	False	True	0.70171	0.240	209	0.97298	0.080	0.97177	0.095
1	True	False	False	True	True	False	True	False	0.71220	0.241	115	0.97182	0.137	0.97396	0.171
2	True	False	False	True	True	False	True	False	0.71254	0.235	99	0.96914	0.131	0.96863	0.167
3	True	False	False	True	True	False	True	False	0.70765	0.237	101	0.96902	0.132	0.97013	0.167
1	True	False	False	True	True	False	False	False	0.74364	0.276	44	0.97595	0.184	0.97570	0.184
2	True	False	False	True	True	False	False	False	0.68381	0.238	42	0.97521	0.167	0.97625	0.189
3	True	False	False	True	True	False	False	False	0.69627	0.238	40	0.97482	0.169	0.97454	0.189
1	True	False	False	True	False	False	True	True	0.71710	0.235	246	0.97128	0.076	0.97109	0.101
2	True	False	False	True	False	False	True	True	0.70991	0.238	228	0.97202	0.082	0.97207	0.104
3	True	False	False	True	False	False	True	True	0.71664	0.235	225	0.97224	0.083	0.97205	0.114
1	True	False	False	True	False	False	False	True	0.71911	0.234	223	0.97294	0.081	0.97282	0.105
2	True	False	False	True	False	False	False	True	0.69303	0.238	196	0.97344	0.085	0.97342	0.094
3	True	False	False	True	False	False	False	True	0.69734	0.237	197	0.97319	0.085	0.97150	0.098
1	True	False	False	True	False	False	True	False	0.72042	0.241	108	0.97158	0.134	0.97289	0.170
2	True	False	False	True	False	False	True	False	0.71146	0.226	90	0.96938	0.128	0.97142	0.166
3	True	False	False	True	False	False	True	False	0.71119	0.221	93	0.96946	0.128	0.97063	0.166
1	True	False	False	True	False	False	False	False	0.76759	0.286	36	0.98476	0.178	0.97825	0.179
2	True	False	False	True	False	False	False	False	0.65601	0.251	24	0.98386	0.158	0.98129	0.188
3	True	False	False	True	False	False	False	False	0.65601	0.251	24	0.98386	0.158	0.98068	0.186

Table 6.1: Predictive error and gene coverage for different descriptive elements used to model the regulatory region. Results from five-fold cross-validation for various combinations of descriptive language elements where only conjunction (column AND is "True") was used (results for other variations of the descriptive language are not shown).

6.6 Computational identification of the most informative regulatory region

Using a similar approach as for the identification of the combination of descriptive elements that form the best descriptive language, we can identify the span (*i.e.*, interval) of the regulatory region from which the best models can be inferred in terms of prediction accuracy and gene coverage. Ideally, the interval should be defined relatively to the transcription start site, but because of the lack of this kind of experimentally measured data, we had to rely on using the transcription start site (ATG) as a landmark for the regulatory region.

Subintervals in steps of 300b from -900b upstream to +900b downstream of ATG were tested. The subinterval had to include some regulatory sequence, *i.e.*, it had to start at positions -900b, -600b or -300b relative to ATG. We used the best descriptive language identified in the previous section (the subinterval used there spanned from -600b to 0b relative to ATG). For each subinterval, the average prediction error and gene coverage using five-fold cross-validation were calculated. Numerical results are shown in Table 6.2 and a graphical rendering is shown in Figure 6.7. The results show that the subinterval with the highest product of normalized prediction correlation and gene coverage is observed in the subinterval -900b to +600b relative to ATG. This indicates that some regulatory

from (b)	to (b)	average prediction correlation ($= 1.0 - error$)	coverage	coverage (normalized)	correlation · coverage
-900	900	0.3364	144	0.6154	0.1885
-900	600	0.2969	231	0.9872	0.2668
-900	300	0.3059	208	0.8889	0.2476
-900	0	0.3160	174	0.7436	0.2139
-900	-300	0.3086	84	0.3590	0.1009
-900	-600	0.2301	8	0.0342	0.0072
-600	900	0.2843	227	0.9701	0.2511
-600	600	0.3016	198	0.8462	0.2324
-600	300	0.2765	234	1.0000	0.2517
-600	0	0.2974	228	0.9744	0.2639
-600	-300	0.3247	81	0.3462	0.1023
-300	900	0.2664	178	0.7607	0.1845
-300	600	0.2865	166	0.7094	0.1851
-300	300	0.2850	164	0.7009	0.1819
-300	0	0.2815	171	0.7308	0.1873

Table 6.2: Average prediction correlation ($1 - prediction\ error$) and gene coverage for different subintervals in the regulatory region. Columns one and two denote the span of the subinterval. The row with the most informative subinterval, from -900b to 600b, is marked in bold.

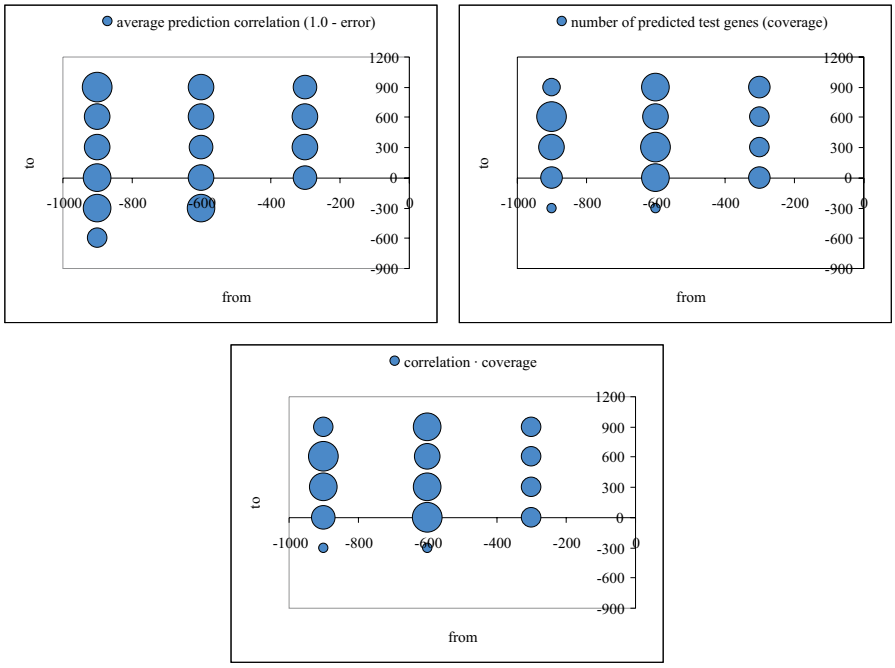


Figure 6.7: a) Average correlation and b) gene coverage achieved using data from various subintervals of the regulatory region. c) Product of the two normalized prediction error and gene coverage measures. Axes "from" and "to" indicate the start and end of the regulatory subinterval used, respectively.

information may be also encoded downstream from translation start, *i.e.*, in introns or exons of genes, which is in agreement with some experimental findings (*e.g.*, see Neznanov et al. [1997]). Another important observation that can be drawn from results in Table 6.2 is that most of the regulatory information may reside in the subinterval -300b to 0b relative to ATG. The expression for the majority of genes (73% of the maximum number of predicted genes in the subinterval -600b to 300b) can be comparably well predicted by observing transcription factor binding sites in the subinterval -300b to 0b only (see last row in Table 6.2). However, because of same reasons as stated in Section 6.3 (*i.e.*, noise in gene expression data, noise and incompleteness of transcription factor binding data, and a limited descriptive language which due to the lack of experimental data, *etc.*), the reported results provide only a partial answer on where regulatory information resides. Same reasons may in part explain the observed difference (*i.e.*, low average prediction correlation) between measured gene expression mRNA levels and promoter activity predicted by the model.

6.7 Analysis of *Saccharomyces cerevisiae* data

In the microarray transcription profiling study by Smith et al. [2002] budding yeast *S. cerevisiae* cells were induced to proliferate peroxisomes – organelles found in most organisms and cell types that compartmentalize several oxidative reactions – as a result of cell's regulated response to absence of glucose or glycerol and exposure to fatty acid oleate as the sole carbon source. Each gene in the data set is described with a transcription profile that consists of six microarray measurements from oleate induction time course and two measurements in "oleate *vs.* glucose" and "glucose *vs.* glycerol" growth conditions. In total, we used eight microarray measurements of gene expression to calculate the distance between genes. We defined the distance function to be $1.0 - Pearson\ correlation$ in gene expression for the given gene pair.

For the target group we selected the set of 224 genes that were identified in the study to have similar expression profiles to those of genes involved in peroxisome biogenesis and peroxisome function. The goal of our analysis was to further divide the target group into smaller subgroups of genes with common elements in promoter structure and possibly identify genes that were inadvertently left out but should have been included in the target group based on their expression and promoter structure [Curk et al., 2006a].

The analysis included information on 2135 putative binding sites that were identified using a local alignment software tool MEME [Bailey and Elkan, 1994]. To obtain putative binding sites, we initially clustered genes, using top-K clustering, into clusters of ten genes. The clustering was done using sequence similarity. Motifs discovered with MEME from each cluster were then merged, and only distinct motifs were kept (details on a similar procedure for *D. discoideum* data are described in the next Chapter 6.8). Using the program MAST [Bailey and Elkan, 1994] we determined the presence of the 2135 putative binding sites in promoter regions for $\sim$ 6700 yeast genes. This analysis has been performed and published [Curk et al., 2006a] before we have obtained the results on the most informative regulatory region reported in the previous section. This is the main reason why we have used the standard one thousand bases (1Kb) promoter regions, taken upstream from the translation start site (ATG), instead of the region spanning -900b to 600b relative to ATG, which should yield better

results. Nonetheless, results reported here provide a good example of usability and the ability of the rule-based clustering method to efficiently discover interesting patterns. The search identified ∼302,000 matches (*i.e.*, occurrences) of putative binding sites. These data was then used to infer rules with rule-based clustering. The algorithm searched for rules describing groups with at least five target genes ($N = 5$) and average group intra-correlation above 0.5 (*i.e.*, the maximum allowed intra-distance was set to $D = 1.0 - 0.5 = 0.5$). We limited the rule search beam to one thousand best rules for further refinements (parameter $L = 1000$). Distances between binding sites were rounded to increments of 40 bases; the maximum possible distance of 2Kb (given the promoter length, relative distances can be in range from -1Kb to +1Kb) was thus reduced to 50 ($= 2000b/40b$) different values. This largely reduced the number of possible subintervals that needed to be considered when inferring rules.

The search resulted in 41 rules describing and dividing 114 target genes (out of total 224 target genes) into 37 subgroups (see Figure 6.8). No rule could be found for the remaining 110 target genes. Most discovered gene groups are composed of five genes with high pairwise intra-group correlation (all are above 0.927). Many genes are shared (overlap) among the 37 discovered groups resulting in six major, unconnected groups visible in Figure 6.8 and Figure 6.9. Seven genes outside the target group were also identified by the method (marked in red in Figure 6.8). For example, the smallest eight gene group, in the top-left corner marked with "1" in Figure 6.8 includes two outside genes (*INP53* and *YIL168W* - also named *SDL1*). Gene ontology analysis shows that *INP53* is involved together with two target genes (*ATP3* and *VHS1*) in the biological process of phosphate metabolism. Gene *SDL1* is annotated to function together with the group's target gene *LYS14* in the biological process amino acid metabolism and other similar parent GO terms (results not shown). These examples confirm the method's ability to identify functionally related genes that were not initially included in the target group. Details about inferred rules, describing the regulatory regions and gene expression profiles of genes from the two groups, marked as "1" and "2" in Figure 6.8 and Figure 6.9, are shown in Figure 6.10 and Figure 6.11.

The majority of discovered rules include conditions that are composed of three terms, each term describing a putative binding site's orientation and distance relative to ATG or binding sites included in the rule. An exhaustive

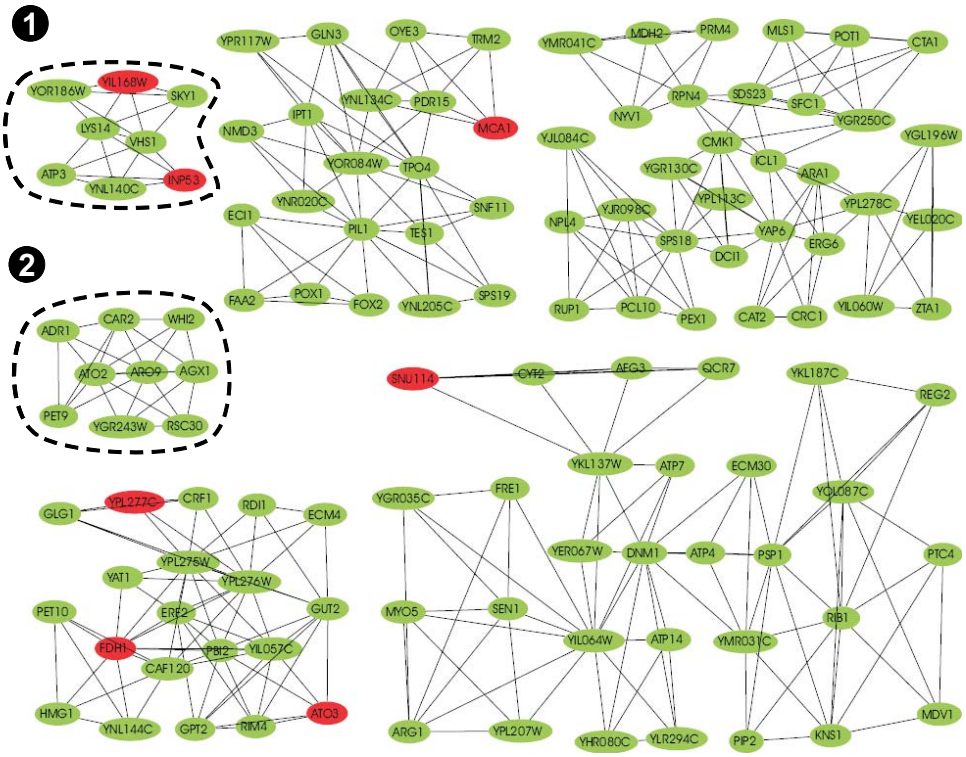


Figure 6.8: Gene network inferred on the peroxisome data set. The 37 discovered clusters form six unconnected groups of genes. In green are 114 target genes, in red are seven outside genes. The two selected groups are marked as "1" and "2."

search for all possible rules composed of three binding sites with defined orientation (three possible values: positive, negative, no preference) and distance (50 different values) would require checking a relatively huge number of rules:

$$\binom{2135 \cdot 3}{3} \cdot 50^3 \approx 5.47 \cdot 10^{15}$$

In this analysis, our method checked $2.11 \cdot 10^9$ of the most promising rules, which is less than 0.00004% of the entire three-part rule space. The search took 40 minutes on a Pentium 4, 3.4 GHz workstation.

In another study by Gasch and Werner-Washburne [2002] yeast cells were subjected to diverse environmental conditions and gene expression in their response to environmental conditions was measured. A set of 900 genes forming the Environment Stress Response (ESR) set was chosen by the authors based on clustering analysis of gene expression. For the target set of genes required by rule-based clustering, we have used the set of 281 genes with increased gene

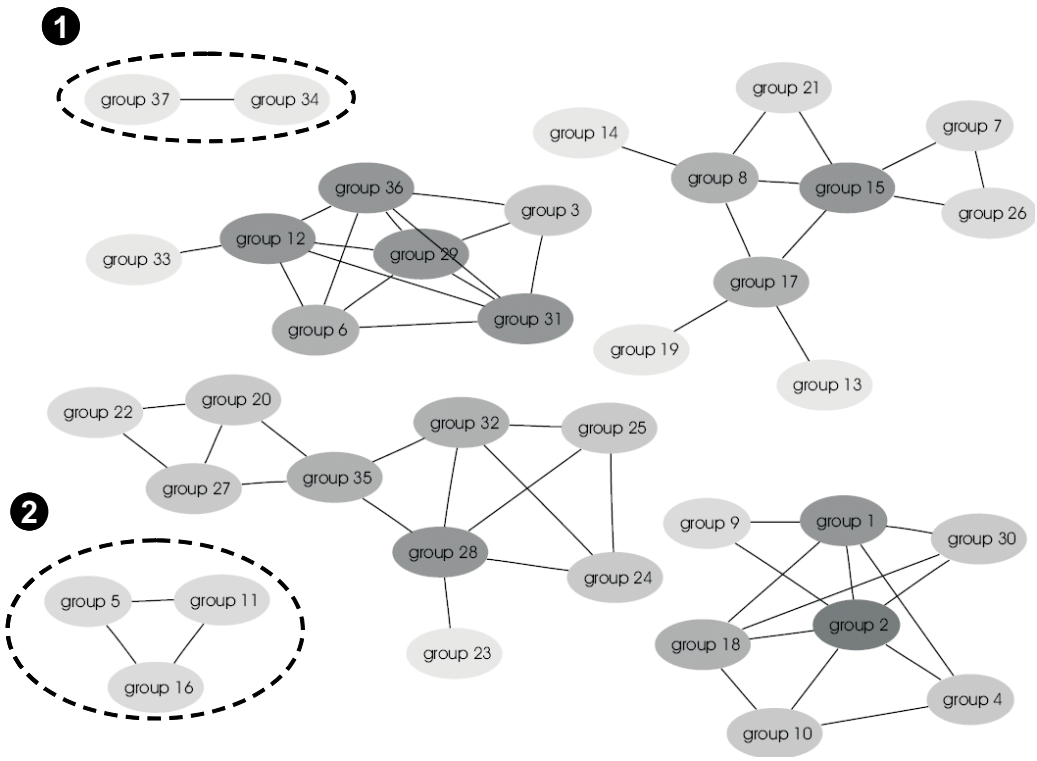


Figure 6.9: Cluster graph of 37 clusters discovered in peroxisome data set. Same two clusters of genes from Figure 6.8 are marked as "1" and "2."

expression as reported by the study. We analyzed the target genes in the context of the entire genome [Curk et al., 2006c]. For this analysis we have used the transcription factor binding data from the study by Lee et al. [2002] to describe the regulatory region of genes. We required rule-based clustering to return clusters no smaller than four genes (parameter $L = 4$) with intra-cluster correlation above 0.45 (*i.e.*, parameter $D = 1.0 - 0.45 = 0.55$). The beam size was set to $L = 1000$. Distances between binding sites were rounded to 40 bases. Rule-based clustering returned a set of clusters. Each cluster is described by rules with conditional part describing two binding sites. The longest description found includes constraints on four binding sites. Figure 6.12 shows only rules requiring the presence of binding sites (inferred rules with other constraints are not shown). The clusters are connected based on the transcription factor binding data from the study by Lee et al. [2002]. Genes, coding transcription factors, are connected by red arrows to their known target genes.

Examining the gene network in Figure 6.12 one can notice two overlapping

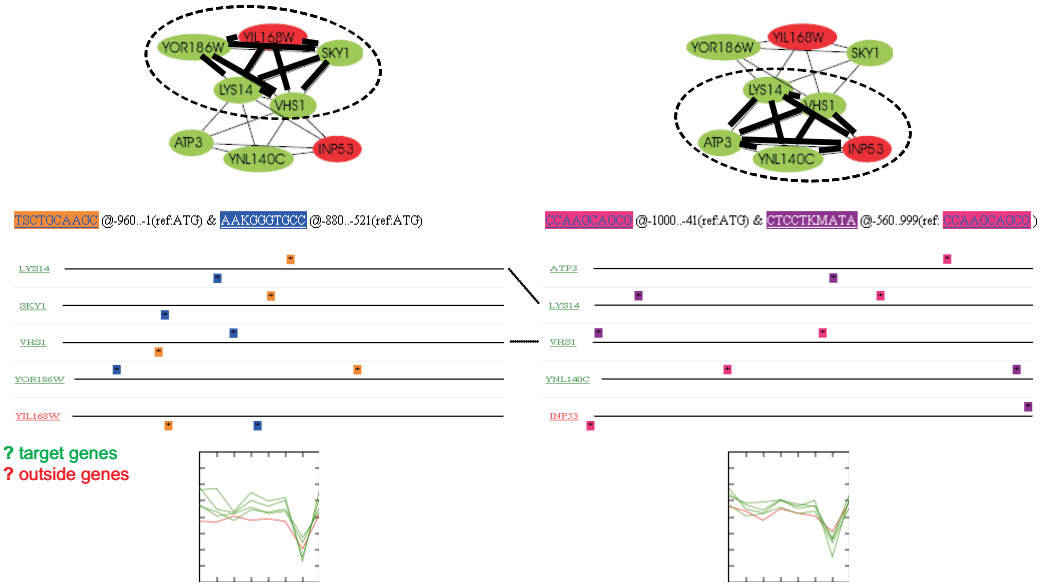


Figure 6.10: Two inferred descriptions (one in each column) of subclusters of the eight genes forming cluster marked "1." Regulatory regions and gene expression profiles of the two clusters (named "group 37" and "group 34") forming cluster "1" in Figure 6.8 and Figure 6.9 are shown.

clusters, each described by its rule, "*Swi5* and *Zms1*," and "*Dot6* and *Mot3*," respectively. This can be easily seen in the rule network visualization (middle left panel, where one can also see that there is one more group of many overlapping clusters, corresponding to genes colored in yellow in the gene network). The two clusters, each formed by four genes, are encircled (in the bottom right part of the top panel in Figure 6.12). The two clusters have in common gene *YML100W* (also named *TSL1*). The annotated biological process in Gene Ontology for the three genes (*YDR258C*, *YPL203W*, *YBR285W*), described by the conditional part of rule "Swi5 and Zms1," is "cellular protein metabolism." The biological process annotation of the three genes (*YBR230C*, *YPL087W*, and *YJR104C*) described by "Dot6 and Mot3" is "response to stress." The shared gene *YML100W* has both annotations ("enzyme regulator activity" and "response to stress"). The visualization of the regulatory regions, showing the four transcription factor binding sites (*Swi5*, *Zms1*, *Dot6*, and *Mot3*), is shown in the bottom of Figure 6.12. There one can see gene *YML100W* having the binding sites present in both discovered clusters. This is an example of the ability of rule-based clustering to discover genes in functionally overlapping clusters.

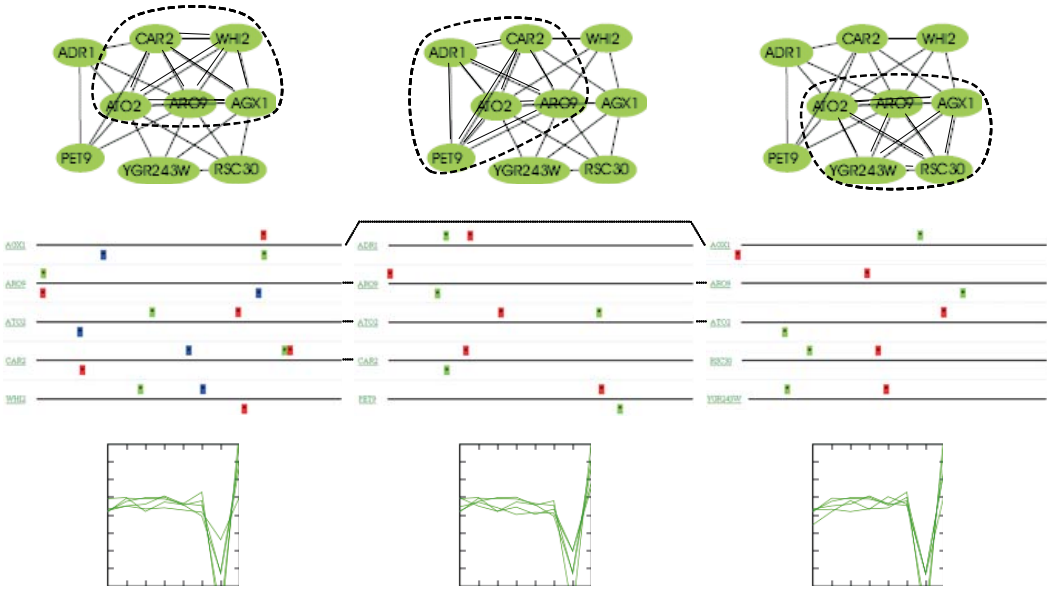


Figure 6.11: Three inferred descriptions (one in each column) of subclusters of the nine genes forming cluster marked "2." Regulatory regions and gene expression profiles of the three clusters (named "group 5," "group 11," and "group 16") forming cluster "2" in Figure 6.8 and Figure 6.9 are shown.

The last example analysis of yeast data was performed on *S. cerevisiae* mitotic cell cycle data studied by Cho et al. [1998]. This example shows that all three types of proposed visualization can be useful in the explorative analysis of regulation of gene expression. Here we have used the TRANSFAC database [Wingender et al., 1996; Matys et al., 2006] as a source of putative binding sites to describe the regulatory regions of 799 genes found to be involved in the mitotic cell cycle by Cho *et al.* Motifs are referenced by their TRANSFAC I.D. The goal of the analysis was to cluster the 799 genes into smaller subclusters and identify genes similar to genes already annotated to be involved in the mitotic cell cycle. Rule-based clustering was thus able to discover 360 rules covering 509 genes. Looking at the gene (example) graph in Figure 6.13 one can see that most genes are covered by more than one rule. No apparent structure can be seen in the gene graph. However, only by using the visualization of rules, shown in the bottom panel, a rich and complex structure of overlapping clusters appears. Details on motifs used to describe features of two groups (group 55 and group 75 in the rule visualization) are shown in top middle and right panels.

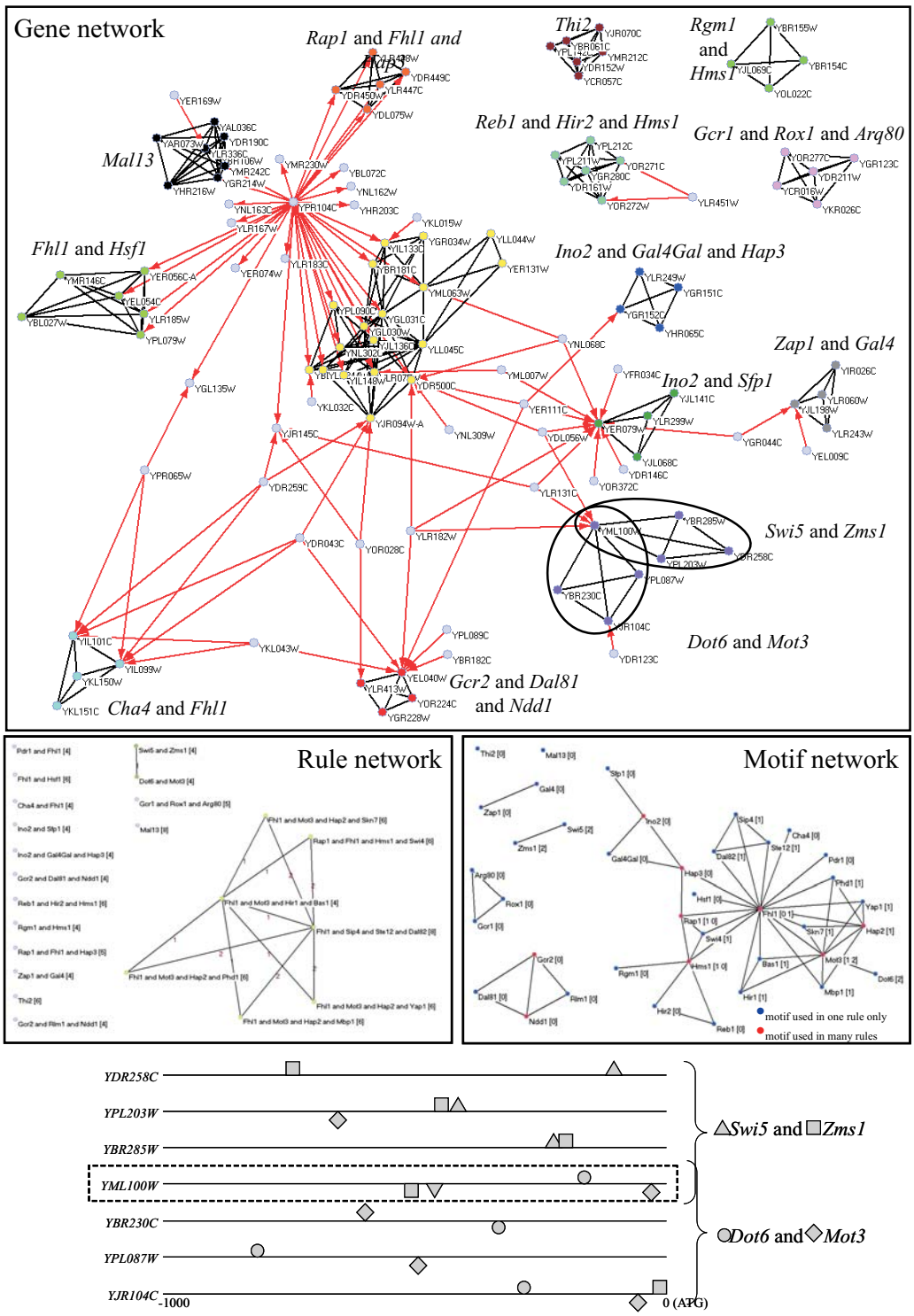


Figure 6.12: Gene, rule and motif networks inferred on environmental stress gene expression data from Gasch and Werner-Washburne [2002]. Shown are details on the binding sites in the regulatory region of gene *YML100W* which is shared by the two clusters.

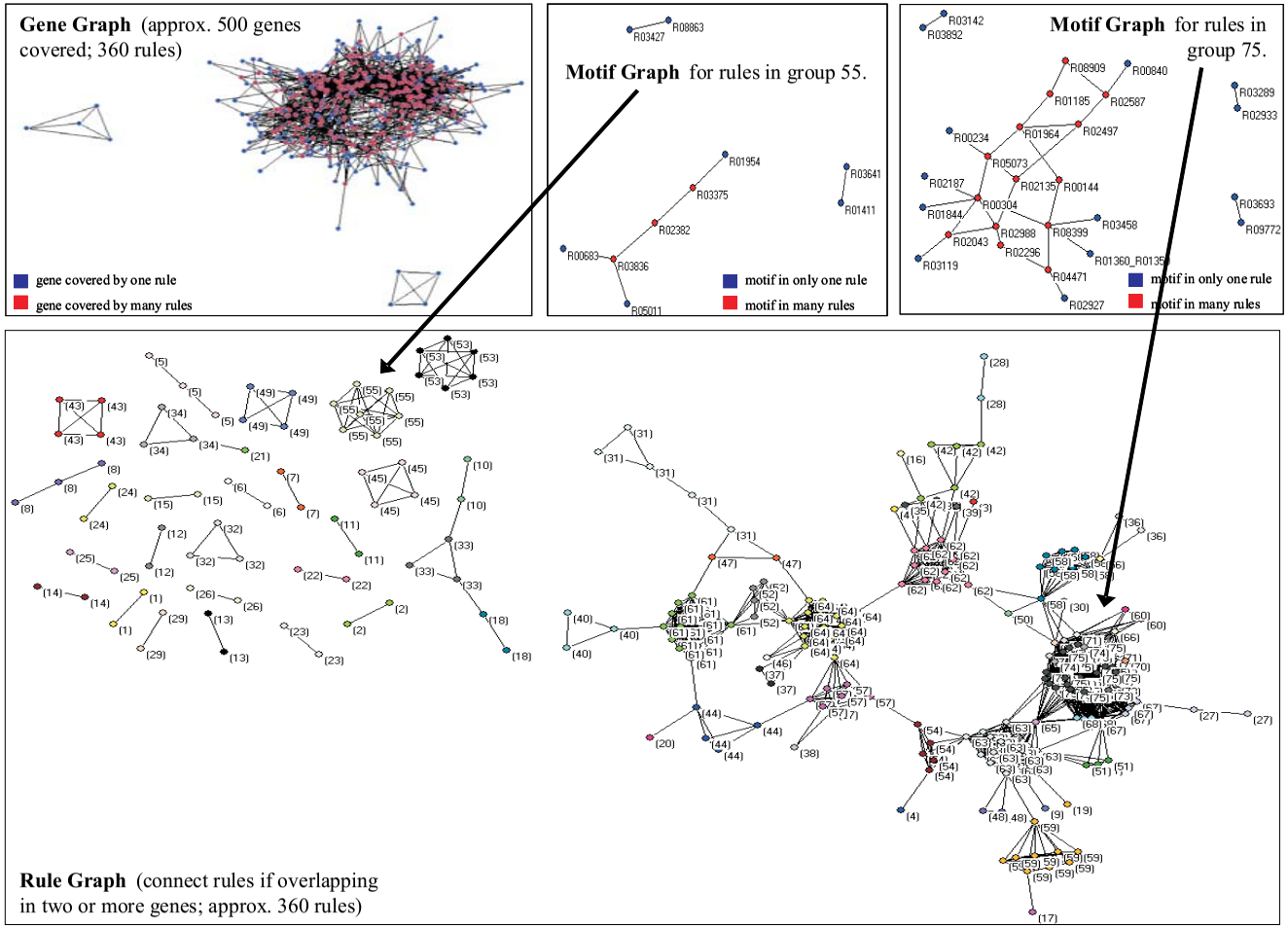


Figure 6.13: Details on the genome-wide analysis of *S. cerevisiae* mitotic cell cycle data from the work by Cho et al. [1998]. The information in a saturated gene graph (top left panel) can be made more intelligible by showing it as a rule graph (shown in bottom panel).

6.8 Analysis of *Dictyostelium discoideum* data

Since no genome-wide assay of transcription factor binding is yet available for *D. discoideum*, we had to rely on an *ab initio* method for the identification of putative transcription factor binding sites. For this task we have used the software tool MEME [Bailey and Elkan, 1994] for local sequence alignment. Program MEME can identify short sequences, referred to also as motifs, which are highly conserved and present in the regulatory region sequences of many genes. Running MEME on the entire set of 4081 promoter regions would be computationally too prohibitive. Since we wanted to discover motifs common to similarly expressed genes, we have applied top-K clustering to form smaller groups of genes on which MEME runs in reasonable time (a few minutes). Each of 4081 genes represents the center of a potential cluster of ten genes. The cluster includes nine genes with expression most similar to the central gene (next paragraph explains how similarity is calculated). After clustering, only distinct clusters are selected, *i.e.*, clusters that are generated around different central genes, but include the same set of genes, are considered only once. MEME was then run on each cluster, with parameters set to return at most ten motifs and search for motifs six to eighteen bases long, yielding a total of 42880 motifs. This set of motifs was reduced by comparing motifs to each other, and keeping only a subset of 14315 distinct motifs, where no two motifs are correlated more than 0.85. Correlation of two motifs is calculated as the average Pearson correlation of frequencies of the four bases at every position. If the two motifs are of different lengths, all alignments where the shorter motif is completely overlapping with the longer, are tried and the highest correlation is reported.

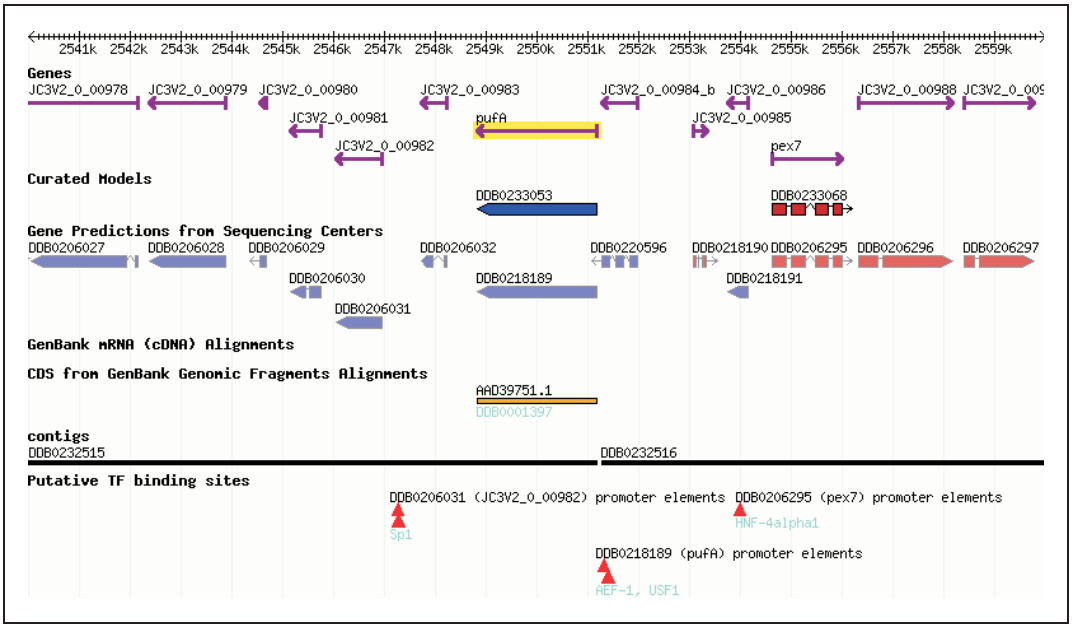
Gene expression data on *D. discoideum* from fifteen gene expression assays of wild-type and mutants was combined into a single distance matrix, which was used for top-K clustering and identification of putative binding sites, and also for rule-based clustering. For each gene expression assay, all pair-wise distances of gene expression were calculated and stored in separate distance matrixes. The final gene distance matrix was calculated as a weighted average of distances from the individual distance matrixes, for each gene pair. The number of DNA microarray measurement replicas performed in an assay was used as a weight when calculating the final weighted average distance matrix.

Assays with more replicas were given a proportionally bigger weight in the final distance matrix. For all analyses we have used the following distance function to calculate distance between a pair of genes: $1.0 - Pearson\ correlation$ of the two expression profiles.

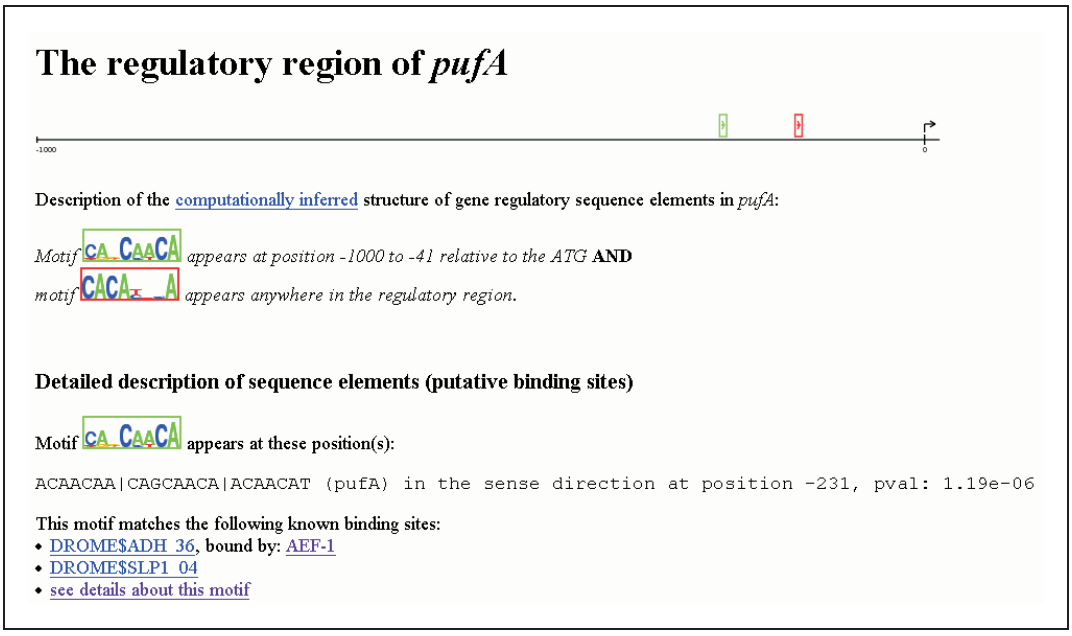
Because all gene expression data is used for the identification of putative binding sites (motifs), this precludes using these data in any subsequent cross-validation schema, for the evaluation of models obtained with rule-based clustering or any other method. Namely, the motifs were found in selected sets of genes with similar expression, and thus the motifs (already) carry information on gene expression. The entire method, top-K clustering for selection of motifs followed by rule-based clustering, should be evaluated in a cross-validation schema.

Here we only report the results of rule-based clustering (discovered patterns) done on *D. discoideum* data. On yeast, where transcription factor binding site data was obtained with other experimental techniques and by using other data types than the gene expression data used in our analyses, we can apply cross-validation and evaluate the predictive ability of rule-based clustering for gene regulatory region analysis. Running rule-based clustering and using wild type gene expression to calculate gene distance, on the set of 4081 genes identified 730 overlapping clusters, covering 1951 genes. The average intra-cluster correlation is 0.65 ± 0.13, with six to thirty genes forming the cluster. More genes (2704) were covered only on the "*yakA- pufA-*" double mutant data. Coverage on other mutants is as follows: 140 genes for "*acaA-*," 238 genes for "*acaA- pkaC+*," 390 genes for "*comA-*," 390 genes for "*comB-*," 598 genes for "*comC-*," 128 genes for "*pkaC-*," 441 genes for "*pkaR-*," 207 genes for "*pkaR- regA-*," 1466 genes for "*pufA-*," 783 genes for "*pufA- pkaC-*," 235 genes for "*pufA- pkaR-*," 622 genes for "*regA-*," and 293 genes for "*yakA-*." Number of clusters varies for each mutants data (not shown). For details see http://bubble.fri.uni-lj.si/dictyBase.

An intermediate file in the GFF format [Durbin et al., 2000] was generated that links web pages with results from rule-base clustering. The file is available at http://bubble.fri.uni-lj.si/dictyBase/browser_WT.txt. Uploading the file into the genome browser at http://dictybase.org should results in a new annotation track, called "Putative TF binding sites," in the genome browser. For a snapshot of the genome browser and the starting web page for gene *pufA* see Figure 6.14. Every gene's promoter element shown in the Genome Browser is linked to a



(a)



(b)

Figure 6.14: Annotation track for Genome Browser (http://dictybase.org) and web pages with results of rule-based clustering on *D. discoideum*. Wild type and mutant gene expression data were used. a) The annotation track, titled "Putative TF binding sites," links to web pages with results of rule-based clustering. b) Web page with detailed results for gene *pufA*.

web page with details on the inferred promoter structure, gene expression, list of genes with similar structure and expression, and links to TRANSFAC motif information. The user can follow links to results obtained on wild type and mutant gene expression data, by clicking on the appropriate gene expression panel (not shown).

When exploring genes of interest, the user can easily see if any gene is linked to other genes described by the discovered rules. This may prove useful when trying to determine a context of genes with similar promoter structure and gene expression to those genes of interest.

Results from a similar genome-wide analysis of same *D. discoideum* mutant data, presented in [Curk et al., 2006b], are also available at http://bubble.fri.uni-lj.si/dictyBase. The main page links to starting pages with results on individual mutants and wild type. The page includes all visualizations of the user interface described in Section 4.4 (*i.e.*, it includes cluster, feature and example networks).

6.9 Summary and discussion

Rule-based clustering was used to identify clusters of genes with similar expression and structure of the regulatory region. Evaluation of the method with cross-validation showed that the inferred descriptions of discovered clusters, encoded in a set of IF-THEN rules, have good predictive value. That is, besides providing a symbolic description, which links the structure of regulatory regions to gene expression, the inferred rules predict well the gene expression of unseen genes.

Using cross-validation we were also able to determine the set of features forming the descriptive language that is most suitable for modeling and achieving good gene coverage. Distance of transcription factor binding sites from the translation start site (ATG) proved to be the single most informative descriptive element. Data on 1.3 transcription factor binding sites per promoter is needed on average to predict gene expression in yeast, and inferred rules include constraints on average 1.17 transcription factor binding sites, with an average length of 1.4 terms (constraints) forming a rule. These values are slightly lower than those reported by Beer and Tavazoie [2004], where they observed that the optimal number is 2.8 terms (*i.e.*, parent nodes in a Bayesian network). The authors report that less or more complex rules resulted in lower predictive performance.

However, the transformation between our model and the model used by Beer *et al.* is not straightforward. Our results can be interpreted as an indication that regulation of gene expression is not highly combinatorial, but it requires relatively specialized transcription factors that can also act in some combinations. Again, caution should be taken due to the incompleteness of the transcription factor binding site data (taken from the study by Harbison et al. [2004]). We share the view expressed by Beer *et al.* that these results should be seen as an indication on the lower limit of the degree of combinatorial regulation, and that using other gene expression data from other experimental conditions, and a more complete transcription factor binding site data may yield more complex combinatorial rules.

Using a similar strategy, the most informative regulatory region for yeast was determined computationally. The region identified for yeast spans from -900 bases to +600 bases relative to ATG (translation start site). This result reconfirms the already known fact that the gene coding sequence may carry some regulatory information (*e.g.*, see [Neznanov et al., 1997]).

Finally, we showed a few examples of analyses done with rule-based clustering on *S. cerevisiae* and *D. discoideum* data. The examples on *S. cerevisiae* data on peroxisome assembly and function studied by Smith et al. [2002] and mitotic cell cycle studied by Cho et al. [1998] show how to use rule-based clustering and its associated visualizations to present the results and guide the user in the exploration of results. Starting with a general overview of identified clusters (Figure 6.8 and Figure 6.9) the user can then focus on specific rules describing individual subclusters (Figure 6.10 and Figure 6.11). A good example where, only by using all three proposed visualization, the user can gain insight into the structure of discovered clusters can be seen in Figure 6.13. Using only the basic gene graph visualization one would fail to see the rich structure of overlapping clusters present in the set of discovered clusters and appertaining rules returned by rule-based clustering.

Results from the whole-genome analysis done on *D. discoideum* data are linked to the organism's genome browser at http://dictybase.org. This allows the user to place genes of interest in a context of other genes identified and determined by rule-based clustering to be similar in promoter structure and gene expression.

All analyses reported here used data on the presence of putative or exper-

imentally confirmed binding sites in the regulatory region of genes and gene expression data. Other sources of genomic data, such as (predicted) promoter secondary structure, chromosomal location of genes, presence and proximity of enhancer and silencer elements, nucleosome organization [Segal et al., 2006], putative or known methylation sites (CpG islands) [Katoh et al., 2006], can be easily included in the analysis by adding descriptive elements (and appropriate operators for feature construction) to the descriptive language used for rule-based clustering. Further work also includes using data on known or predicted transcription start sites instead of the translation start site (ATG) used in all the analyses reported here. Both landmarks (transcription and translation start sites) should be included in the descriptive language and, by using the methods for evaluation described here, the individual contributions to the model's predictive ability of each should be estimated.

Chapter 7

Decomposition of gene expression profile signatures

In this Chapter we propose a heuristic algorithm for the decomposition of gene expression profile signatures. We give examples on two model organisms where the proposed decomposition method was successfully used to identify parallel pathways in gene networks. The method is implemented and available as a web application (see http://bubble.fri.uni-lj.si/microCOMB).

7.1 Introduction and related work

A gene expression profile signature is defined as the subset of all measured genes which best represents the cell's response to the condition under which gene expression was measured. In our description of the algorithm for the decomposition of signature profiles we will use data from microarray gene expression differential studies, where gene expression of all genes (*i.e.*, the transcriptional profile) in the test sample measured under some condition (*e.g.*, treatment, mutant, *etc.*) is compared to gene expression in a reference sample (*e.g.*, no treatment, wild type or gene expression from a reference pool). In such setting, the gene expression signature is composed of genes whose expression has changed (*i.e.*, increased or decreased compared to the reference expression) for more than a user-specified threshold. The algorithm for the decomposition of signature profiles can be also used for other types of gene profile data (computational phenotypes), such as measurements of absolute gene expression levels, mutant sensitivity profiles, *etc.*

The main motivation for the development of the proposed signature decomposition method was to complement the analysis done with rule-based clustering. While rule-based clustering is applied to discover clusters of genes with specific regulatory patterns in gene expression and structure of the regulatory region, the proposed decomposition of signature profiles allows detecting more global similarities between genes, by using the transcriptional phenotype of corresponding mutants. Another important motivation leading to the development of the method was to enable the researcher to easily compare and relate measurements from his own experiment(s) to a large collection of previously measured gene expression data from various assays (*e.g.*, SGD's gene expression connection data at www.yeastgenome.org includes $\sim$ 900 microarray measurements from 20 studies). Similarly to the BLAST algorithm, developed by Altschul et al. [1990], which proved extremely useful for rapid sequence comparison and sequence database search, the proposed method for the decomposition of gene signature was developed to offer an easy comparison and search in a gene profile signatures database.

Another motivation that lead to the development of the proposed decomposition method are recent reports on the "modular nature" of the cell's genomic program [Ihmels et al., 2002; Segal et al., 2003a], and developed methods for inference of epistatic relations from transcriptional phenotypes of mutants [Van Driessche et al., 2005] that are then used for inference of genetic networks We show that, when applied on a database of mutant expression data, the decomposition method can be used to describe a mutant's response as a combination of responses of other mutants. These inferred relations can then support the user in reasoning about the underlying genetic network and about the contributions of genes forming the transcriptional response in each component.

Related work include SVD (singular value decomposition) proposed by Alter *et al.* and Carter *et al.* [Alter et al., 2000; Carter et al., 2006, 2007] for discovering groups of genes with similar regulation and function, or similar cellular state and biological phenotype, called eigengenes and eigenarrays, respectively. Differently to SVD-based approaches, the proposed signature decomposition method is more general as it does not assume any underlying superimposition (*i.e.*, linear combination) of contributions of genes or signature profiles (*e.g.*, in all our examples we use the nonlinear function "*min*").

7.2 Decomposition algorithm

The user has to provide a "query" microarray gene expression measurement for which he wants to find a decomposition that describes the measurement as a combination of measurements stored in a database. There are no restrictions on the kind of condition, treatment or mutant for which the gene expression transcriptional profile was measured and included in the database; they all can be included in the database.

The decomposition algorithm is given in Figure 7.1. The query signature profile consists of genes with significantly altered query expression (*e.g.*, absolute gene expression above a user-defined threshold) (see algorithm in Figure 7.1., line 1). All other query genes, whose gene expression did not change greatly under the experiment's conditions, are not considered by the decomposition algorithm. All transcriptional profiles in the database are filtered using the same threshold criteria, and their signature profiles are generated. Additionally, only genes forming in the query profile are used to form the database signature profiles (algorithm, lines 2 and 3). Depending on the selected threshold, the number of genes to consider can be much smaller than the entire genome (*e.g.*, less than thousand genes, out of seven thousand genes in the genome, for experiments done on yeast *S. cerevisiae*), which makes further steps of the algorithm run faster. To reduce the number of combinations to explore, only N database signature profiles that are most similar to the query signature profile are taken (line 4). Among those signature profiles, all combinations of 1 to K signature profiles (also called components) are explored (loop in line 5). Each combination produces a "composed" signature profile S_c (loop in line 7). The composed signature S_c is then compared with the query signature S_q. The algorithm returns an ordered list (line 12), where at the top are the compositions most similar to the query signature. Inversely, a composition can be seen as a "decomposition" of the query signature into a combination of components.

Two important steps of the decomposition algorithm are combining the components into a final signature (see Figure 7.2) and testing the quality of a particular decomposition (algorithm in Figure 7.1, line 10) by comparing it to the query signature. After combining the given components of a potential decomposition of query signature (combination of components "one," "two," and "three" in Figure 7.2), the resulting "composition" (fourth column in Figure 7.2) should be as similar to the query signature (column five in Figure 7.2) as possible.

Different functions can be used to combine the components into a composition. In the description of the algorithm and in all our examples, we use the function "*min*" (algorithm in Figure 7.1, line 8) since we are minimizing the difference of gene expression between the selected component and the query signature for each gene in the query signature. Other functions such as "*average*," "*weighted sum*," *etc.* can also be used, requiring a few changes to the algorithm. As stated in line 8 of the algorithm in Figure 7.2, function "*min*" is defined as follows: for the given query gene and the given combination of database signature components, the component with most similar expression (*i.e.*, minimum distance) to the query expression, must be selected. If function "*average*" is used, then the expression in all components must be averaged into one value, which forms the composition's value.

Whichever function is used for combining components into a composition, the composition will differ from the query signature. The difference between the composed signature profile (S_c) and query signature (S_q) can be calculated and used to rank and select the best decomposition among all decompositions identified during search. Because we wanted the components of the best decomposition to include as many genes as possible (ideally, all genes in the query signature profile), and at the same time have a good correlation between the query and composed signature, we used this score function: $correlation \cdot coverage$, where *correlation* is Pearson correlation between the query and composed signature profile, and *coverage* is the number of genes forming the composed signature profile. Other score functions that can be used (correlation only, Euclidean distance only, *etc.*) are included in the web tool we have implemented for this method.

Applying an exhaustive, combinatorial search for the decomposition of signature profiles is not feasible due to the large number of combinations that need to be explored when searching for decompositions including three or more components ($K \geq 3$), and using a database of few thousand transcriptional profiles. For this we propose a heuristic search algorithm, where at the beginning N single components that most closely match the query signature are identified. All combinations of order from 1 to K, among the selected N components, are then tested with the goal to find the combination that best match the query signature profile.

Algorithm for decomposition of gene signature profiles

Input:
Q – "query" gene expression transcription profile
th – threshold for determining the signature profile
K – maximum number of components of a decomposition
N – number of most similar database signature profiles that can be used to form a decomposition
DB – database of gene expression transcriptional profiles

Output:
ordered list L of best decompositions

1 apply threshold th to select genes and form a query signature profile S_q
2 apply threshold filter on each profile in the database to form database profile signatures
3 in database profile signatures keep only those genes that appear in S_q
4 select N database signature profiles DB_{topN} that are most similar to S_q
5 **FOR EACH** combination C of $1..K$ signature profiles in DB_{topN} **DO**
6 reset composed signature profile S_c
7 **FOR EACH** gene G in S_q **DO**
8 find signature profile C_i in C with most similar gene expression to gene expression of G in S_q ($i.e., S_q[G]$)
9 $S_c[G] \leftarrow C_i[G]$; compose signature profile S_c
10 calculate distance between S_c and S_q
11 add decomposition C into L and order based on calculated distance (more similar decomposition on top of L)
12 **return** L

Figure 7.1: Algorithm for the decomposition of gene signature profiles.

7.3 Experimental evaluation

The decomposition method was successfully applied on several biologically interesting examples. Here we report on two examples done on the two model organisms: budding yeast *S. cerevisiae* and slime mold *D. discoideum*.

Figure 7.3 shows an example decomposition found by the algorithm applied to *D. discoideum* mutant data. The method identified genes (in this example the components represent mutants) that act in parallel and can be used to describe the signature profile of a common downstream gene ("*pkaC*-") in the genetic network for development in *D. discoideum*. The genetic network for develop-

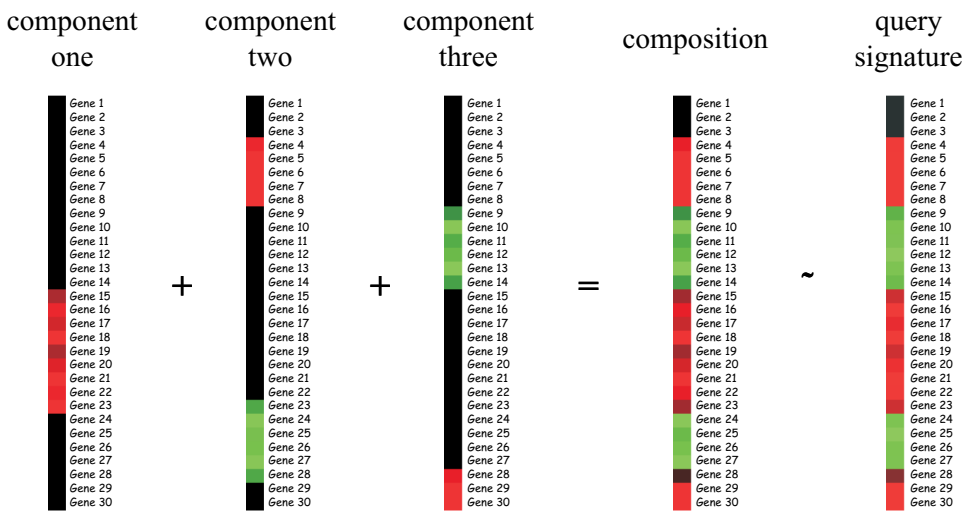


Figure 7.2: Decomposition (composition) of gene expression signature. Three components (component one, two, and three) are composed into one signature ("composition"), which is a good approximation of the query signature of interest. The composition operation (operator marked with '+') in this example is performed with function "*min*."

ment is shown in Figure 7.4. The decomposition algorithm was instructed to find the best decomposition of the expression from the "*pkaC*-" mutant, using a relatively small database including gene expression transcriptional profiles data on wild type and fourteen mutants. The best decomposition, shown in Figure 7.3, states that the transcriptional signature profile of mutant "*pkaC*-" can be well described as a combination of subsets of genes from mutants "*yakA*-" (1028 genes), "*acaA*-" (614 genes), and the double mutant "*pufA- pkaC*-" (574 genes). The two largest components, "*yakA*-" and "*acaA*-" are known to act in parallel and are upstream of gene "*pkaC*-" in the genetic network reported in Figure 7.4. The third component (the double mutant "*pufA- pkaC*-") is expected to be included because the double mutant is known to be transcriptionally similar to the "*pkaC*-". This was observed in the epistasis analysis reported by Van Driessche et al. [2005]. However, the best decomposition, identified by our method, shows that there are other mutants (parts of their transcriptional profiles) that are more similar to the "*pkaC*-" mutant signature profile. Besides providing a set of mutants (components) that are most similar to the query

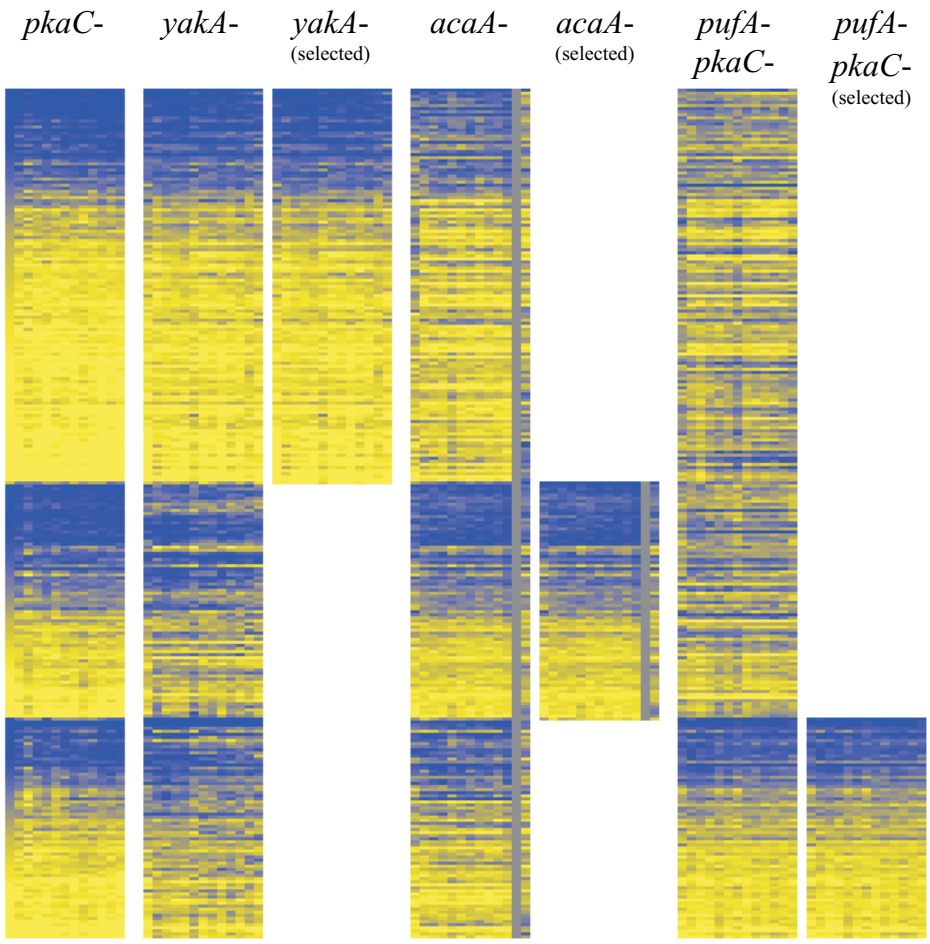


Figure 7.3: Decomposition of *pkaC-* mutant expression profile into three components. The expression of all genes in each component is shown. The expression of the subset of genes, forming the component, is repeated in the columns with caption "(selected)." Shown is the expression of 2216 genes (rows) in thirteen time points, eight genes are averaged and shown as one row.

mutant's signature, each set of genes can be further studied (*e.g.*, by performing an annotation enrichment analysis) to gain more insight about individual components and relations among components.

In this example, all gene expression profiles (query and database) include 4081 genes. The best decomposition found describes 2216 genes. In this example, gene expression was measured in a time series of 13 time points (from 0 to 24h, every two hours). Distance among gene expression time series was calculated with Pearson correlation.

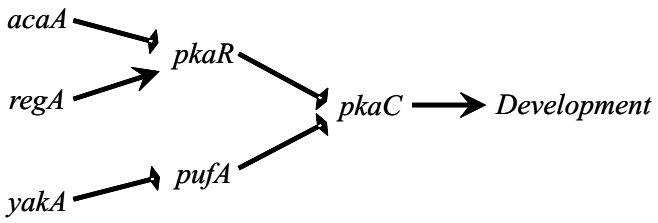
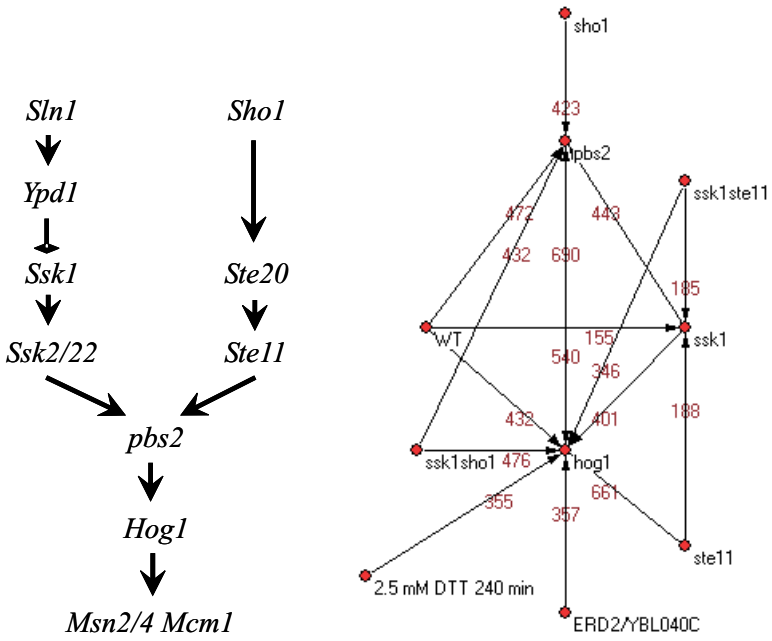


Figure 7.4: Genetic network for development in *D. discoideum*. This network was reported in the study of epistasis analysis by Van Driessche et al. [2005].



(a) (b)

Figure 7.5: Analysis of the HOG MAPK pathway in *S. cerevisiae*. a) HOG MAPK pathway presented in Roberts et al. [2000]. b) Graph summarizing the top ten decompositions found for mutant query signatures of *Hog1*, *Pbs2*, and *Ssk1* mutants.

A second, more extensive decomposition was performed on *S. cerevisiae* data. The database included all SGD's Gene Expression connection data ($\sim$ 900 transcriptional profiles from 20 studies) and data from the paper by Roberts et al. [2000] where they studied several MAPK (mitogen-activated protein kinase) pathways, which control changes in gene expression, cytoskeletal organization, and cell division. For this experiment we have selected the HOG MAPK path-

way, which responds to hypertonic stress, and for which data on most mutants of genes in the pathway was available. Using the decomposition algorithm we identified the best decompositions for *Hog1*, *Pbs2* and *Ssk1* mutants's expression profiles appearing in the pathway proposed by Roberts et al. [2000] and showed in Figure 7.5a. The first ten best decomposition of each mutant query profile were used to build a graph of decompositions (shown in Figure 7.5b). Directed edges (arcs) connect the query profile signature with components appearing in the first ten best decompositions of the query profile. The number on the arc is the average number of genes from the component, averaged across all decompositions forming the arc that is contributing to the decomposition of a query signature transcriptional profile. The arcs point from single components to the signature profile for which decomposition was found.

The largest arc links from *Hog1* to *Pbs2* (see Figure 7.5b), with an average 690 genes from *Hog1* being most similar in expression to *Pbs2*. Although not acting in parallel, but in a cascade (see Figure 7.5a), those two genes are the most similar, with an average of 690 genes from component *Hog1* contributing to the decomposition of *Pbs2*. The weight of the arc in the opposite direction is also comparably high (540 genes from *Pbs2* contributing to the decomposition of *Hog1*). More interesting are the arcs linking to *Hog1* and *Pbs2*. These are all genes that appear upstream of the two genes and are acting in parallel paths in the pathways. In the graph in Figure 7.5b, mutants *Sho1* and *Ssk1*, are two such cases that are connected to *Pbs2*.

7.4 Summary and discussion

The proposed method for the decomposition of gene expression (transcriptional) profiles was shown to be useful to study newly acquired transcriptional data and relate it to previous measurements of gene expression. We showed that the proposed algorithm is able to decompose a query signature profile into components. We also showed that such decompositions can prove useful for placing the query transcriptional response into a biological context. Similar to epistasis analysis, where entire mutant transcriptional phenotypes are compared to infer epistatic relations, the proposed method for decomposition tries to identify individual parts (components) of the cell's response to the environment, condition or treatment, which can be explained with data on other mutants.

Although designed for DNA microarray transcriptional phenotype data, the decomposition algorithm can be used on other types of data (*e.g.*, sensitivity profiles). We have implemented the decomposition method as a web-based tool (available at http://bubble.fri.uni-lj.si/microCOMB). For now, the tool includes a database on *S. cerevisiae* data only.

Further work on this topic include evaluating other functions to combine signature profiles, including "*average*," "*weighted sum*," and other. Devising good heuristics, which do not search the entire space of combinations, is also an important aspect for further work.

Chapter 8

Conclusion and further work

In this dissertation we proposed a set of computational methods for inference of gene networks from various data sources. For this, we addressed several important problems in gene network inference, including function prediction from different types of gene profile and phenotype data, methods for the analysis of gene regulatory regions, and methods for the decomposition of gene signature profiles.

Starting with the concept of "computational phenotype" we showed that different types of gene profiles are better predictors of different types of functional annotations. We directly compared the utility of gene expression profile and mutant-based phenotypes (gene characterizations) for gene function prediction, which we modeled with gene co-expression networks, and proposed to use ROC analysis to measure the ability of gene co-expression networks to discriminate among gene functional annotations. The results showed no single absolute winner. Moreover, we have shown examples where entire subgroups of gene functional classes could be better predicted from different types of characterization. This supports our claim that all sources of experimental data are needed for successful prediction of gene function. Further work includes investigating and developing ways to automatically learn how to combine different gene characterizations for a better prediction of gene function.

The main contribution of this Thesis is the new machine learning approach, called rule-based clustering. The rule-based clustering method combines the classical CN2 [Clark and Nibblet, 1989] rule-inference search procedure and the method of clustering trees developed by Blockeel et al. [1998], and is able to identify overlapping subgroups of similar examples. Each identified cluster can

be described by a set of symbolic description(s) encoded in form of IF-THEN rules. This approach greatly differs from the standard cluster-first approach, where examples are first clustered based on their similarity, and then a description of each cluster is attempted.

Two sets of attributes are required. The first set is used to calculate example distance with a user-defined distance function. Another set of attributes and optionally, operators for feature construction, are required for inference of symbolic descriptions. Here, background knowledge on the problem domain, encoded in the feature construction operators and in the selection of attributes given by the user, can be crucial for successfully solving a specific problem.

Search in rule-based clustering is performed using a large beam. It is guided by a heuristic which prefers to refine rules describing clusters with higher potential to form even more coherent subclusters (informed search). Rule-based clustering can also incorporate an on-the-fly feature construction. Only new features, that are possible on a current subset of examples, are tried.

The result of rule-based clustering is a model that lists a set of rules. The modeling done with rule-based clustering is both descriptive and predictive. Due to the symbolic language used to encode the conditional part of the rule, the user can gain new knowledge by studying the patterns found and presented with the inferred rules. By observing the number of rules found describing each cluster of genes, the user can decide which discovered clusters should be given more weight in subsequent steps of the analysis. We also showed how to evaluate those same rules for their predictive ability using the standard machine learning evaluation method of k-fold cross-validation.

Using proper visualization can augment the way one explores and studies the inferred models and underlying data, especially when a rich descriptive language is used for describing discovered patterns and inferring complex models. A good visualization of discovered patterns and clusters of examples may allow the user to discover higher-order structure or other properties of inferred patterns. For this, we propose three visualizations: example network, cluster network and feature network. We also propose a user interface that incorporates all three types of visualizations and allows for explorative data analysis.

Although designed with bioinformatics problems in mind, the method can be regarded as a general machine learning technique. The approach is able to discover clusters – groups of data items, or genes – that are described using a

symbolic assertion on the gene sequence (the conditional part), with genes in the group bearing similar phenotypes (the action part of the rule). We showed examples on *S. cerevisiae* and *D. discoideum* data where rule-based clustering was successfully used to combine sequence and gene expression information to find clusters of genes with similar gene expression profiles and sequence (*i.e.*, the structure of the regulatory region). In Chapter 6 we showed that the inferred models have a fairly good predictive ability. We defined a descriptive language for modeling the gene regulatory region. An evaluation of the various elements of the proposed descriptive language showed that distance of transcription factor binding site from ATG is the single most informative descriptive element, which should be used when predicting gene expression from the content of the regulatory regions.

Still, further research of the most appropriate descriptive language for modeling regulation of gene transcription is needed. Additional elements known from biological theory need to be included and their predictive value assessed. Additional descriptive elements should include data on chromatin structure, CpG islands [Katoh et al., 2006], predicted secondary structure, ncRNA, RNAi, *etc.* The main challenge here remains obtaining genome-wide data of good quality.

Further work on the rule-based clustering method includes using other types of gene characterizations (*e.g.*, mutant sensitivity profiles) for the identification of cluster of genes with similar regulatory structure (preliminary work, not shown in this Thesis, indicates that other types of gene characterization could be used to identify groups of co-regulated genes).

Rule-based clustering requires a number of parameters, *i.e.*, length of the beam, significance level when testing decrease in variance after refinement of a cluster, *etc.* Future work includes finding ways to minimize the number of parameters, or transforming them to make them more intuitive to the end user, and making the method more adaptive to different problem domains, without any user intervention or setting of parameters. For example, the distance threshold parameter D given by the user could be replaced by a parameter limiting the percentage of most similar groups discovered by the method that get reported to the user. As search progresses, the method could keep track of the intra-distance of all groups discovered so far, and automatically set parameter D to keep only an arbitrarily selected percentage (set by user) of all discovered clusters. Another, more technical aspect of future work, is to use a full con-

junctive and disjunctive language, eliminating current constraints of usage of disjunctively added terms.

Given its generality, rule-based clustering could be applied to many other problem domains, one being the inference of relations of structures of chemical compounds – small molecules and their corresponding phenotype effects [Lamb, 2007; Lamb et al., 2006]. Similar to the application for the analysis of gene regulatory regions, here measured phenotype (*i.e.*, gene expression) could be used to calculate distance between compounds, and a descriptive language for chemical formulae and other structural information on molecules could be used to describe clusters of similar responses to small molecules.

The proposed decomposition of gene expression profile signatures, presented in Chapter 7, proved useful when analyzing newly measured transcriptional data, and placing it into a biological context. The decomposition method tries to identify parts (or components) of the cell's response to the environment, condition, treatment or some other perturbation. We showed two examples on *S. cerevisiae* and *D. discoideum* data, where the approach was successfully applied. The decomposition method was implemented as a web-based tool.

Further work on this topic include a thorough evaluation of other functions that can be used to combine signature profiles, and devising good heuristics that would speed-up the search for decompositions. The main motivation for the development of the proposed method for the decomposition of signature profiles was to complement the analysis done with rule-based clustering, and to allow the researcher to easily compare his own measurements with other published data. Further work includes developing algorithms that automate the merging of relations inferred by rule-based clustering and relations found with the decomposition of gene signature profiles.

Bibliography

Akutsu, T., Miyano, S., and Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the boolean network model. *Pac Symp Biocomput*, pages 17–28.

Akutsu, T., Miyano, S., and Kuhara, S. (2000). Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–34.

Alberts, B., Bray, D., Lewis, J., Raff, M., and Roberts, KeithWatson, J. D. (1994). *Molecular Biology of the Cell*. Garland Publishing, New York/New York, third edition edition.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., and Staudt, L. M. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–11.

Alter, O., Brown, P. O., and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A*, 97(18):10101–6.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–10.

Ankerst, M., Keim, D., and H.-P., K. (1996). Circle segments: A technique for visually exploring large dimensional data sets. *In Proceedings of the IEEE Visualization Conference*.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald,

M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–9.

Avery, L. and Wasserman, S. (1992). Ordering gene function: the interpretation of epistasis in regulatory hierarchies. *Trends Genet*, 8(9):312–6.

Bailey, T. L. and Elkan, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol*, 2:28–36.

Bajic, V. B., Tan, S. L., Suzuki, Y., and Sugano, S. (2004). Promoter prediction analysis on the whole human genome. *Nat Biotechnol*, 22(11):1467–73.

Barabasi, A.-L. (2002). *Linked: the new science of networks*. Perseus Pub., Cambridge, Mass.

Barabasi, A. L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nat Rev Genet*, 5(2):101–13.

Batada, N. N., Reguly, T., Breitkreutz, A., Boucher, L., Breitkreutz, B. J., Hurst, L. D., and Tyers, M. (2007). Still stratus not altocumulus: further evidence against the date/party hub distinction. *PLoS Biol*, 5(6):e154.

Batagelj, V. and Mrvar, A. (2003). Pajek - analysis and visualization of large networks. In Jünger, M. and Mutzel, P., editors, *Graph Drawing Software*, pages 77–103. Springer, Berlin.

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

Bertin, N., Simonis, N., Dupuy, D., Cusick, M. E., Han, J. D., Fraser, H. B., Roth, F. P., and Vidal, M. (2007). Confirmation of organized modularity in the yeast interactome. *PLoS Biol*, 5(6):e153.

Birnbaum, K., Benfey, P. N., and Shasha, D. E. (2001). cis element/transcription factor analysis (cis/tf): a method for discovering transcription factor/cis element relationships. *Genome Res*, 11(9):1567–73.

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Ben-Dor, A., Sampas, N., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Carpten, J., Gillanders, E., Leja, D., Dietrich, K., Beaudry, C., Berens, M., Alberts, D., and

Sondak, V. (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406(6795):536–40.

Blockeel, H., De Raedt, L., and Ramon, J. (1998). Top-down induction of clustering trees. *Machine Learning*.

Booth, E. O., Van Driessche, N., Zhuchenko, O., Kuspa, A., and Shaulsky, G. (2005). Microarray phenotyping in dictyostelium reveals a regulon of chemotaxis genes. *Bioinformatics*, 21(24):4371–7.

Braga-Neto, U. M. and Dougherty, E. R. (2004). Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–80.

Bratko, I. (2001). *PROLOG Programming for Artificial Intelligence*. Addison-Wesley, third edition edition.

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., J., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci U S A*, 97(1):262–7.

Bussemaker, H. J., Li, H., and Siggia, E. D. (2001). Regulatory element detection using correlation with expression. *Nat Genet*, 27(2):167–71.

Carter, G. W., Prinz, S., Neou, C., Shelby, J. P., Marzolf, B., Thorsson, V., and Galitski, T. (2007). Prediction of phenotype and gene expression for combinations of mutations. *Mol Syst Biol*, 3:96.

Carter, G. W., Rupp, S., Fink, G. R., and Galitski, T. (2006). Disentangling information flow in the ras-camp signaling network. *Genome Res*, 16(4):520–6.

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

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell*, 2(1):65–73.

Choi, C. H., Kalosakas, G., Rasmussen, K. O., Hiromura, M., Bishop, A. R., and Usheva, A. (2004). Dna dynamically directs its own transcription initiation. *Nucleic Acids Res*, 32(4):1584–90.

Clark, P. and Nibblet, T. (1989). The cn2 induction algorithm. *Machine Learning*, 3(4):261–83.

Conlon, E. M., Liu, X. S., Lieb, J. D., and Liu, J. S. (2003). Integrating regulatory motif discovery and genome-wide expression analysis. *Proc Natl Acad Sci U S A*, 100(6):3339–44.

Curk, T., Demsar, J., Xu, Q., Leban, G., Petrovic, U., Bratko, I., Shaulsky, G., and Zupan, B. (2005a). Microarray data mining with visual programming. *Bioinformatics*, 21(3):396–8.

Curk, T., Petrovic, U., Shaulsky, G., and Zupan, B. (2005b). Mutant vs. gene expression profiles for function prediction. In *In Workshop Notes of IDAMAP-05*, pages 15–20.

Curk, T., Petrovic, U., Shaulsky, G., and Zupan, B. (2006a). Rule-based clustering for gene regulation pattern discovery. In *IDAMAP 2006: Intelligent Data Analysis in Biomedicine and Pharmacology*, pages 45–50, Department of Computer Science, University of Verona, Italy.

Curk, T., Shaulsky, G., and Zupan, B. (2006b). Discovering patterns of gene regulation in dictyostelium discoideum using rule-based clustering. In *International Dictyostelium Conference 2006*, page 127, Hotel La Fonda, Santa Fe, New Mexico, USA.

Curk, T., Zupan, B., Petrovic, U., Demsar, J., Shaulsky, G., Sacchi, L., Larizza, C., and Bellazzi, R. (2003). Machine learning for functional genomics: some experiments with supervised learning on microarray data set. In Abu-Hanna, A. and Hunter, J., editors, *Working notes of the joint workshop on intelligent data analysis in medicine and pharmacology and knowledge-based information management in anaesthesia and intensive care 2003*, pages 47–51, held at the ninth conference on Artificial Intelligence in Medicine Europe, Cyprus, 19-22 October, 2003.

Curk, T., Zupan, B., Petrovic, U., and Shaulsky, G. (2006c). Odkrivanje pravil uravnavanja izražanja genov z razvrščanjem na podlagi pravil = rule-based clustering for discovery of patterns in gene expression regulation. *Informatica medica slovenica*, 11(1):52–59.

De Jong, H., Geiselmann, J., Batt, G., Hernandez, C., and Page, M. (2004a). Qualitative simulation of the initiation of sporulation in bacillus subtilis. *Bull Math Biol*, 66(2):261–99.

De Jong, H., Gouze, J. L., Hernandez, C., Page, M., Sari, T., and Geiselmann, J. (2004b). Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull Math Biol*, 66(2):301–40.

Demsar, J., Zupan, B., and Leban, G. (2004a). Orange: From experimental machine learning to interactive data mining, white paper (www.ailab.si/orange).

Demsar, J., Zupan, B., Leban, G., and Curk, T. (2004b). Orange: from experimental machine learning to interactive data mining. In *European Conference of Machine Learning*, pages 537–539, Pisa, Italy. Springer Verlag.

Down, T. A., Bergman, C. M., Su, J., and Hubbard, T. J. (2007). Large-scale discovery of promoter motifs in drosophila melanogaster. *PLoS Comput Biol*, 3(1):e7.

Durbin, R., Haussler, D., Stein, L., Lewis, S., Krogh, A., and others, a. (2000). Gff (general feature format) specifications document.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331.

Eichinger, L., Pachebat, J. A., Glockner, G., Rajandream, M. A., Sucgang, R., Berriman, M., Song, J., Olsen, R., Szafranski, K., Xu, Q., Tunggal, B., Kummerfeld, S., Madera, M., Konfortov, B. A., Rivero, F., Bankier, A. T., Lehmann, R., Hamlin, N., Davies, R., Gaudet, P., Fey, P., Pilcher, K., Chen, G., Saunders, D., Sodergren, E., Davis, P., Kerhornou, A., Nie, X., Hall, N., Anjard, C., Hemphill, L., Bason, N., Farbrother, P., Desany, B., Just, E., Morio, T., Rost, R., Churcher, C., Cooper, J., Haydock, S., van Driessche, N., Cronin, A., Goodhead, I., Muzny, D., Mourier, T., Pain, A., Lu, M., Harper, D., Lindsay, R., Hauser, H., James, K., Quiles, M., Madan Babu, M., Saito, T., Buchrieser, C., Wardroper, A., Felder, M., Thangavelu, M., Johnson, D., Knights, A., Loulseged, H., Mungall, K., Oliver, K., Price, C., Quail, M. A., Urushihara, H., Hernandez, J., Rabbinowitsch, E., Steffen, D., Sanders, M., Ma, J., Kohara, Y., Sharp, S., Simmonds, M., Spiegler, S., Tivey, A., Sugano, S., White, B., Walker, D., Woodward, J., Winckler, T., Tanaka, Y., Shaulsky, G., Schleicher, M., Weinstock, G., Rosenthal, A., Cox, E. C., Chisholm, R. L., Gibbs, R., Loomis, W. F., Platzer, M., Kay, R. R., Williams, J., Dear, P. H., Noegel, A. A., Barrell, B., and Kuspa, A. (2005). The genome of the social amoeba dictyostelium discoideum. *Nature*, 435(7038):43–57.

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

Ženko, B. (2007). *Learning predictive clustering rules*. PhD thesis, University of Ljubljana.

Fawcett, T. (2003). Roc graphs: Notes and practical considerations for data mining researchers. Technical report, HP Laboratories.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a). Data mining and knowledge discovery in databases. *Communications of the ACM*, 39(11):24–26.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996b). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54.

Foster, I. (2006). 2020 computing: a two-way street to science's future. *Nature*, 440(7083):419.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *J Comput Biol*, 7(3-4):601–20.

Friend, S. H. and Stoughton, R. B. (2002). The magic of microarrays. *Scientific American Magazine*, 286(2):44–9.

Gasch, A. P. and Eisen, M. B. (2002). Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biol*, 3(11):RESEARCH0059.

Gasch, A. P. and Werner-Washburne, M. (2002). The genomics of yeast responses to environmental stress and starvation. *Funct Integr Genomics*, 2(4-5):181–92.

Goh, K. I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., and Barabasi, A. L. (2007). The human disease network. *Proc Natl Acad Sci U S A*, 104(21):8685–90.

GuhaThakurta, D. and Stormo, G. D. (2001). Identifying target sites for cooperatively binding factors. *Bioinformatics*, 17(7):608–21.

Hannenhalli, S. and Levy, S. (2002). Predicting transcription factor synergism. *Nucleic Acids Res*, 30(19):4278–84.

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J. B., Reynolds, D. B., Yoo, J., Jennings, E. G., Zeitlinger, J., Pokholok, D. K., Kellis, M., Rolfe, P. A., Takusagawa, K. T., Lander,

E. S., Gifford, D. K., Fraenkel, E., and Young, R. A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104.

Harrison, R. and DeLisi, C. (2002). Condition specific transcription factor binding site characterization in saccharomyces cerevisiae. *Bioinformatics*, 18(10):1289–96.

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., and Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pac Symp Biocomput*, pages 422–33.

Holloway, D., Kon, M., and DeLisi, C. (2006). Machine learning for regulatory analysis and transcription factor target prediction in yeast. *Syst Synth Biol*.

Hughes, T. R. (2005). Universal epistasis analysis. *Nat Genet*, 37(5):457–8.

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., Kidd, M. J., King, A. M., Meyer, M. R., Slade, D., Lum, P. Y., Stepaniants, S. B., Shoemaker, D. D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M., and Friend, S. H. (2000). Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–26.

Hulf, T., Bellosta, P., Furrer, M., Steiger, D., Svensson, D., Barbour, A., and Gallant, P. (2005). Whole-genome analysis reveals a strong positional bias of conserved dmyc-dependent e-boxes. *Mol Cell Biol*, 25(9):3401–10.

Hvidsten, T. R., Laegreid, A., and Komorowski, J. (2003). Learning rule-based models of biological process from gene expression time profiles using gene ontology. *Bioinformatics*, 19(9):1116–1123.

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

International, H. G. S. C. (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945.

Jolly, E. R., Chin, C. S., Herskowitz, I., and Li, H. (2005). Genome-wide identification of the regulatory targets of a transcription factor using biochemical characterization and computational genomic analysis. *BMC Bioinformatics*, 6:275.

Katoh, M., Curk, T., Xu, Q., Zupan, B., Kuspa, A., and Shaulsky, G. (2006). Developmentally regulated dna methylation in dictyostelium discoideum. *Eukaryot Cell*, 5(1):18–25.

Kimball, J. W. (2006). Kimball's biology pages. http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/.

Lamb, J. (2007). The connectivity map: a new tool for biomedical research. *Nat Rev Cancer*, 7(1):54–60.

Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J. P., Subramanian, A., Ross, K. N., Reich, M., Hieronymus, H., Wei, G., Armstrong, S. A., Haggarty, S. J., Clemons, P. A., Wei, R., Carr, S. A., Lander, E. S., and Golub, T. R. (2006). The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313(5795):1929–35.

Latchman, D. S. (1998). *Eukaryotic transcription factors (Third Edition)*. Academic Press, San Diego/California.

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

Liang, S., Fuhrman, S., and Somogyi, R. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput*, pages 18–29.

Luscombe, N. M., Greenbaum, D., and Gerstein, M. (2001). What is bioinformatics? a proposed definition and overview of the field. *Methods Inf Med*, 40(4):346–58.

Matys, V., Kel-Margoulis, O. V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A. E., and Wingender, E. (2006). Transfac and its module transcompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res*, 34(Database issue):D108–10.

McCarroll, S. A., Li, H., and Bargmann, C. I. (2005). Identification of transcriptional regulatory elements in chemosensory receptor genes by probabilistic segmentation. *Curr Biol*, 15(4):347–52.

McPherson, J. D., Marra, M., Hillier, L., Waterston, R. H., Chinwalla, A., Wallis, J., Sekhon, M., Wylie, K., Mardis, E. R., Wilson, R. K., Fulton, R., Kucaba, T. A., Wagner-McPherson, C., Barbazuk, W. B., Gregory, S. G., Humphray, S. J., French,

L., Evans, R. S., Bethel, G., Whittaker, A., Holden, J. L., McCann, O. T., Dunham, A., Soderlund, C., Scott, C. E., Bentley, D. R., Schuler, G., Chen, H. C., Jang, W., Green, E. D., Idol, J. R., Maduro, V. V., Montgomery, K. T., Lee, E., Miller, A., Emerling, S., Kucherlapati, Gibbs, R., Scherer, S., Gorrell, J. H., Sodergren, E., Clerc-Blankenburg, K., Tabor, P., Naylor, S., Garcia, D., de Jong, P. J., Catanese, J. J., Nowak, N., Osoegawa, K., Qin, S., Rowen, L., Madan, A., Dors, M., Hood, L., Trask, B., Friedman, C., Massa, H., Cheung, V. G., Kirsch, I. R., Reid, T., Yonescu, R., Weissenbach, J., Bruls, T., Heilig, R., Branscomb, E., Olsen, A., Doggett, N., Cheng, J. F., Hawkins, T., Myers, R. M., Shang, J., Ramirez, L., Schmutz, J., Velasquez, O., Dixon, K., Stone, N. E., Cox, D. R., Haussler, D., Kent, W. J., Furey, T., Rogic, S., Kennedy, S., Jones, S., Rosenthal, A., Wen, G., Schilhabel, M., Gloeckner, G., Nyakatura, G., Siebert, R., Schlegelberger, B., Korenberg, J., Chen, X. N., Fujiyama, A., Hattori, M., Toyoda, A., Yada, T., Park, H. S., Sakaki, Y., Shimizu, N., Asakawa, S., et al. (2001). A physical map of the human genome. *Nature*, 409(6822):934–41.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–7.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

Neznanov, N., Umezawa, A., and Oshima, R. G. (1997). A regulatory element within a coding exon modulates keratin 18 gene expression in transgenic mice. *J Biol Chem*, 272(44):27549–57.

Pavesi, G., Mereghetti, P., Mauri, G., and Pesole, G. (2004). Weeder web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res*, 32(Web Server issue):W199–203.

Pedersen, A. G., Baldi, P., Chauvin, Y., and Brunak, S. (1999). The biology of eukaryotic promoter prediction–a review. *Comput Chem*, 23(3-4):191–207.

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

Pilpel, Y., Sudarsanam, P., and Church, G. M. (2001). Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat Genet*, 29(2):153–9.

Pournara, I. and Wernisch, L. (2004). Reconstruction of gene networks using bayesian learning and manipulation experiments. *Bioinformatics*, 20(17):2934–42.

Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, 42:203–231.

Qiu, P. (2003). Recent advances in computational promoter analysis in understanding the transcriptional regulatory network. *Biochem Biophys Res Commun*, 309(3):495–501.

Quackenbush, J. (2002). Microarray data normalization and transformation. *Nat Genet*, 32 Suppl:496–501.

Roberts, C. J., Nelson, B., Marton, M. J., Stoughton, R., Meyer, M. R., Bennett, H. A., He, Y. D., Dai, H., Walker, W. L., Hughes, T. R., Tyers, M., Boone, C., and Friend, S. H. (2000). Signaling and circuitry of multiple mapk pathways revealed by a matrix of global gene expression profiles. *Science*, 287(5454):873–80.

Sacchi, L., Bellazzi, R., Larizza, C., Magni, P., Curk, T., Petrovic, U., and Zupan, B. (2005). Ta-clustering: cluster analysis of gene expression profiles through temporal abstractions. *Int J Med Inform*, 74(7-8):505–17.

Schlitt, T. and Brazma, A. (2005). Modelling gene networks at different organisational levels. *FEBS Lett*, 579(8):1859–66.

Schlitt, T. and Brazma, A. (2006). Modelling in molecular biology: describing transcription regulatory networks at different scales. *Philos Trans R Soc Lond B Biol Sci*, 361(1467):483–94.

Schmid, C. D., Perier, R., Praz, V., and Bucher, P. (2006). Epd in its twentieth year: towards complete promoter coverage of selected model organisms. *Nucleic Acids Res*, 34(Database issue):D82–5.

Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*, 18(20):6097–100.

Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thastrom, A., Field, Y., Moore, I. K., Wang, J. P., and Widom, J. (2006). A genomic code for nucleosome positioning. *Nature*, 442(7104):772–8.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003a). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet*, 34(2):166–76.

Segal, E., Taskar, B., Gasch, A., Friedman, N., and Koller, D. (2001). Rich probabilistic models for gene expression. *Bioinformatics*, 17 Suppl 1:S243–52.

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

Shrager, J., Langley, P., and Pohorille, A. (2002). Guiding revision of regulatory models with expression data. *Pac Symp Biocomput*, pages 486–97.

Smith, J. J., Marelli, M., Christmas, R. H., Vizeacoumar, F. J., Dilworth, D. J., Ideker, T., Galitski, T., Dimitrov, K., Rachubinski, R. A., and Aitchison, J. D. (2002). Transcriptome profiling to identify genes involved in peroxisome assembly and function. *J Cell Biol*, 158(2):259–71.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12):3273–97.

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

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

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., and Church, G. M. (1999). Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–5.

Terai, G. and Takagi, T. (2004). Predicting rules on organization of cis-regulatory elements, taking the order of elements into account. *Bioinformatics*, 20(7):1119–28.

The 2020 Science Group (2006). The 2020 science group, towards 2020 science. Microsoft Research, Cambridge, UK, July 12, 2006.

Thieffry, D., Huerta, A. M., Perez-Rueda, E., and Collado-Vides, J. (1998). From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in escherichia coli. *Bioessays*, 20(5):433–40.

Thieffry, D. and Thomas, R. (1998). Qualitative analysis of gene networks. *Pac Symp Biocomput*, pages 77–88.

Toh, H. and Horimoto, K. (2002). Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18(2):287–97.

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

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

Van Driessche, N. (2004). *Transcriptional profiling of Dictyostelium discoideum growth and development*. PhD thesis, Baylor College of Medicine.

Van Driessche, N., Demsar, J., Booth, E. O., Hill, P., Juvan, P., Zupan, B., Kuspa, A., and Shaulsky, G. (2005). Epistasis analysis with global transcriptional phenotypes. *Nat Genet*, 37(5):471–7.

Van Driessche, N., Shaw, C., Katoh, M., Morio, T., Sucgang, R., Ibarra, M., Kuwayama, H., Saito, T., Urushihara, H., Maeda, M., Takeuchi, I., Ochiai, H., Eaton, W., Tollett, J., Halter, J., Kuspa, A., Tanaka, Y., and Shaulsky, G. (2002). A transcriptional profile of multicellular development in dictyostelium discoideum. *Development*, 129(7):1543–52.

van Nimwegen, E., Zavolan, M., Rajewsky, N., and Siggia, E. D. (2002). Probabilistic clustering of sequences: inferring new bacterial regulons by comparative genomics. *Proc Natl Acad Sci U S A*, 99(11):7323–8.

Vuk, M. and Curk, T. (2006). Roc curve, lift chart and calibration plot. *Metodološki zvezki*, 3(1):89–108.

Wagner, A. (2001). How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. *Bioinformatics*, 17(12):1183–97.

Wahde, M. and Hertz, J. (2001). Modeling genetic regulatory dynamics in neural development. *J Comput Biol*, 8(4):429–42.

Wasserman, W. W. and Sandelin, A. (2004). Applied bioinformatics for the identification of regulatory elements. *Nat Rev Genet*, 5(4):276–87.

Wessels, L. F., van Someren, E. P., and Reinders, M. J. (2001). A comparison of genetic network models. *Pac Symp Biocomput*, pages 508–19.

Wingender, E., Dietze, P., Karas, H., and Knuppel, R. (1996). Transfac: a database on transcription factors and their dna binding sites. *Nucleic Acids Res*, 24(1):238–41.

Winzeler, E. A., Shoemaker, D. D., Astromoff, A., Liang, H., Anderson, K., Andre, B., Bangham, R., Benito, R., Boeke, J. D., Bussey, H., Chu, A. M., Connelly, C., Davis, K., Dietrich, F., Dow, S. W., El Bakkoury, M., Foury, F., Friend, S. H., Gentalen, E., Giaever, G., Hegemann, J. H., Jones, T., Laub, M., Liao, H., Liebundguth, N., Lockhart, D. J., Lucau-Danila, A., Lussier, M., M'Rabet, N., Menard, P., Mittmann, M., Pai, C., Rebischung, C., Revuelta, J. L., Riles, L., Roberts, C. J., Ross-MacDonald, P., Scherens, B., Snyder, M., Sookhai-Mahadeo, S., Storms, R. K., Veronneau, S., Voet, M., Volckaert, G., Ward, T. R., Wysocki, R., Yen, G. S., Yu, K., Zimmermann, K., Philippsen, P., Johnston, M., and Davis, R. W. (1999). Functional characterization of the s. cerevisiae genome by gene deletion and parallel analysis. *Science*, 285(5429):901–6.

Wuensche, A. (1998). Genomic regulation modeled as a network with basins of attraction. *Pac Symp Biocomput*, pages 89–102.

Yoo, C. and Cooper, G. F. (2002). Discovery of gene-regulation pathways using local causal search. *Proc AMIA Symp*, pages 914–8.

Yoo, C., Thorsson, V., and Cooper, G. F. (2002). Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational dna microarray data. *Pac Symp Biocomput*, pages 498–509.

Yu, H., Luscombe, N. M., Qian, J., and Gerstein, M. (2003). Genomic analysis of gene expression relationships in transcriptional regulatory networks. *Trends Genet*, 19(8):422–7.

Zhu, J. and Zhang, M. Q. (1999). Scpd: a promoter database of the yeast saccharomyces cerevisiae. *Bioinformatics*, 15(7-8):607–11.

Zhu, Z., Pilpel, Y., and Church, G. M. (2002). Computational identification of transcription factor binding sites via a transcription-factor-centric clustering (tfcc) algorithm. *J Mol Biol*, 318(1):71–81.

Zupan, B., Bratko, I., Demsar, J., Juvan, P., Curk, T., Borstnik, U., Beck, J. R., Halter, J., Kuspa, A., and Shaulsky, G. (2003a). Genepath: a system for inference of genetic networks and proposal of genetic experiments. *Artif Intell Med*, 29(1-2):107–30.

Zupan, B., Demsar, J., Bratko, I., Juvan, P., Halter, J. A., Kuspa, A., and Shaulsky, G. (2003b). Genepath: a system for automated construction of genetic networks from mutant data. *Bioinformatics*, 19(3):383–389.

Zupan, B., Demsar, J., Curk, T., Petrovic, U., and Shaulsky, G. (2006). Computational phenomics, with emphasis on gene expression data analysis. In *First International Conference on Computational Systems Biology*, pages 19–24, Shanghai, China. Shanghai: The Center for American Studies, Fudan University.

Dodatek A

Računski pristopi k odkrivanju genskih mrež

Razširjeni povzetek v slovenskem jeziku

A.1 Povzetek

V pričujočem delu predlagamo nabor računskih metod za gradnjo genskih mrež na podlagi različnih genomskih podatkov in obravnavamo pomembne probleme, ki nastopijo pri gradnji genskih mrež. Le-ti so: napovedovanje funkcije genov na podlagi različnih, tako imenovanih računskih fenotipov (profil izražanja genov, transkripcijski fenotip mutiranih sevov in kvantitativni rastni fenotip mutiranih sevov), analiza regulatornih regij genov, dekompozicija profilov izražanja genov.

Glavni prispevek pričujoče disertacije je metoda, ki hkrati obravnava genetski zapis DNA in informacijo o fenotipu, ter tako poišče skupine genov s podobnim fenotipom in zapisom v regulatornih regijah genov. Metoda je osnovana na podlagi novega pristopa k strojnemu učenju, imenovanem razvrščanje na podlagi pravil (ang. *rule-based clustering*), ki ga predlagamo v disertaciji. Pristop omogoča odkrivanje skupin primerov oziroma genov, katere člane lahko opišemo na simboličnim način, zakodiranim v pogojnem delu pravila. V zaključku pravila sledi opis fenotipa, ki je značilen za vse gene v skupini. Čeprav je bila metoda za razvrščanje na podlagi pravil prvotno razvita za odkrivanje pravil uravnavanje izražanja genov, je splošno uporabna tudi za reševanje drugih problemov. Metoda zahteva dva nabora atributov. Na podlagi prve skupine

atributov se izračuna razdalje med vsemi pari primerov, medtem ko se druga množica atributov uporablja za gradnjo simboličnih opisov odkritih skupin. Predlagani algoritem za odkrivanje pravil uporablja metodo iskanja v snopu (ang. *beam search*) ter statistični test za izbor pravil in za ustavitveni pogoj iskanja. Metoda omogoča odkrivanje prekrivajočih skupin. Za boljšo predstavitev in interpretacijo odkritih pravil predlagamo nabor vizualizacij odkritih skupin.

Eksperimentalno smo ovrednotili in pokazali uspešne primere uporabe metode za razvrščanje na podlagi pravil za odkrivanje uravnalnih pravil izražanja genov glive sluzavke *Dictyostelium discoideum* in kvasovke *Saccharomyces cerevisiae*, za kar smo uporabili podatke o izmerjenem izražanju genov z DNA mikromrežami in podatke o zapisu DNA regulatornih regij genov. Z uporabo metode razvrščanja na podlagi pravil smo poskusili odgovoriti tudi na nekaj pomembnejših bioloških vprašanj: kateri so najbolj pomembni opisni elementi regulatornih regij in kateri del zapisa DNA regulatorne regije nosi največ informacije o uravnavanju genov.

Ovrednotili smo zmožnost napovedovanja funkcije genov na podlagi različnih tipov podatkov o fenotipu. Rezultati kažejo, da univerzalno najboljši tip podatkov ne obstaja, temveč je potrebno uporabiti vse razpoložljive podatke.

V disertaciji obravnavamo in predlagamo metodo za dekompozicijo profilov izražanja genov. Za podani profil izražanja genov celotnega genoma, metoda poišče komponente (podskupine genov in njihovo izražanje pod različnimi pogoji ali za različne seve, podatki o tem so shranjeni v podatkovni bazi), ki združene skupaj v nov profil dobro aproksimirajo od uporabnika podan profil izražanja genov. Rezultat dekompozicije je torej mreža mutant in pogojev, ki umesti uporabnikovo meritev v neki biološki kontekst ter tudi omogoča odkrivanje genskih poti ter določanje v njih udeleženih genov.

Ključne besede

strojno učenje,
bioinformatika,
vizualizacija,
razvrščanje na podlagi pravil,
genske mreže, funkcijska genomika,
modeliranje uravnavanja izražanja genov,
dekompozicija profilov izražanja genov

A.2 Uvod

V disertaciji se ukvarjamo z razvojem in uporabo računskih metod za analizo genetskih podatkov. Končni cilj predlaganih metod je izgradnja in prikaz modela v formalnem, simboličnem zapisu, to je, v obliki genske mreže. Obsežnost in hitra rast količine in raznolikosti genetskih podatkov [Luscombe et al., 2001] zahteva razvoj specializiranih metod za analizo genetskih podatkov, kar predstavlja velik izziv in hkrati priložnost za napredek v razumevanju celičnih procesov in posledično izboljšanje zdravljenja bolezni [Schlitt and Brazma, 2005, 2006].

Glavni prispevki disertacije so:

- razvoj metode za razvrščanje na podlagi pravil (ang. *rule-based clustering*), nova metoda strojnega učenja za razvrščanje primerov, ki omogoča obravnavo kompleksnih in opisno bogatih problemskih domen,
- praktična implementacija metode razvrščanja na podlagi pravil, za reševanje problemov na področju bioinformatike. Formalizirali in ovrednotili smo opisni jezik za modeliranje strukture regulatorne regije genov,
- razvoj in implementacija računske metode za dekompozicijo profilov izražanja genov,
- poskusi na področju računske fenomike. Ocenjevali smo uspešnost napovedovanja funkcije genov na podlagi različnih tipov podatkov o fenotipu.

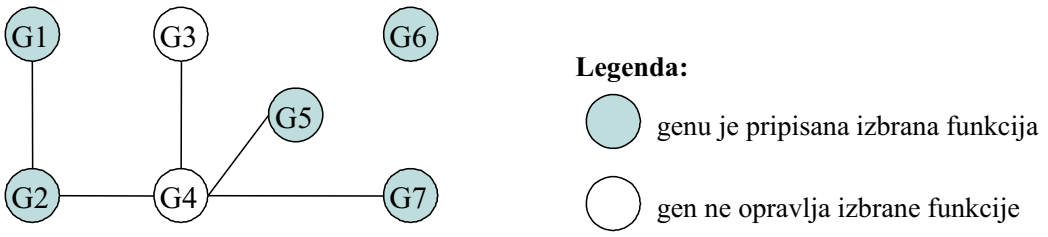
Predlagamo tudi vrsto vizualizacij za podporo odkrivanju zakonitosti v podatkih in tolmačenju model odkritih z metodo razvrščanja na podlagi pravil. Vse razvite metode smo uporabili za analizo podatkov dveh modelnih organizmov, in sicer kvasovke *Saccharomyces cerevisiae* in glive sluzavke *Dictyostelium discoideum*. Praktični prispevki disertacije vključujejo implementacijo metode razvrščanja v skupine na podlagi pravil v obliki skript za programski jezik Python, v oviru sistema za strojno učenje Orange [Demsar et al., 2004a], obsežno analizo regulatornih regij genov glive sluzavke *D. discoideum*, ki je dosegljiva na spletni strani http://bubble.fri.uni-lj.si/dictyBase, ter preko uradne spletne strani o organizmu, na naslovu http://dictybase.org [Eichinger et al., 2005]. Praktični prispevek je tudi implementacija metode dekompozicije profilov izražanja v obliki spletne aplikacije, dosegljive na naslovu http://bubble.fri.uni-lj.si/microCOMB.

A.3 Računska fenomika

V tem poglavju smo poskusili odgovoriti na vprašanje koliko informacije o funkciji genov nosijo različni tipi genomskih podatkov. Pri določanju funkcije posameznega gena, se genetiki zanašajo na "klasično" opazovanje morfološkega fenotipa. Le-ta pa je navadno nezadosten za obsežne celo-genomske študije, saj takšen fenotip nosi relativno malo informacije o celotnem stanju organizma, kar je pa potrebno pri sklepanju o funkciji (več) genov. Ovrednotili smo različne tipe genomskih podatkov, ki opisujejo vpliv oziroma odziv vseh genov genoma in za katere lahko upravičeno trdimo, da predstavljajo neki "globalni oziroma univeralni fenotip" celice. Ker lahko tovrstni podatki opisujejo tudi do nekaj deset tisoč genov hkrati in je njihova obravnava smiselna oziroma možna le z računalnikom, jih imenujemo računski fenotip.

Za metodo modeliranja funkcije genov smo uporabili metodo mrež so-izraženih genov (ang. *gene co-expression networks*), ki so jo razvili Stuart in sodelavci [Stuart et al., 2003]. Najprej se izračuna korelacijo (podobnost) profilov izražanja genov za vse pare genov. Nato se poveže tiste pare genov, ki so nad izbranim pragom podobnosti. Tako nastane mreža genov (glej sliko A.1). Vozlišča genov s pripisano izbrano funkcijo se pobarva. Za tako obarvano mrežo se izračuna dve meri uspešnosti napovedovanja funkcije na podlagi podobnosti genov (pokritost razreda in klasifikacijo točnost mreže za izbrani razred, ang. *network coverage and accuracy*). Pokritost je razmerje pobarvanih vozlišč, ki so povezani z vsaj še enim pobarvanim vozliščem, deljeno s številom vseh pobarvanih genov. Klasifikacijska točnost mreže je število povezav pobarvanih vozlišč, ki povezujejo ostala pobarvana vozlišča, deljeno s številom vseh povezav, ki izhajajo iz pobarvanih vozlišč. Meri uspešnosti, ki ju predlagajo Stuart in sodelavci, je težko uporabiti za primerjavo uspešnosti napovedovanja različnih funkcij genov ali primerjavo uspešnosti napovedovanja mrež so-izraženih genov zgrajenih na različnih računskih fenotipih. Zato v disertaciji predlagamo bolj ustrezno mero uspešnosti napovedovanja funkcije genov, ki je osnovana na analizi ROC krivulj [Provost and Fawcett, 2001; Fawcett, 2003].

V empiričnih poskusih smo uporabili različne genomske podatke oziroma računske fenotipe za izračun podobnosti genov (klasični profil izražanja genov izmerjen pod različnimi pogoji, transkripcijski fenotip mutantov, to je izražanje vseh genov nekega mutanta, ter kvantitativni, rastni fenotip mutantov v različnih pogojih). Za vsak tip podatkov smo zgradili mreže so-izraženih

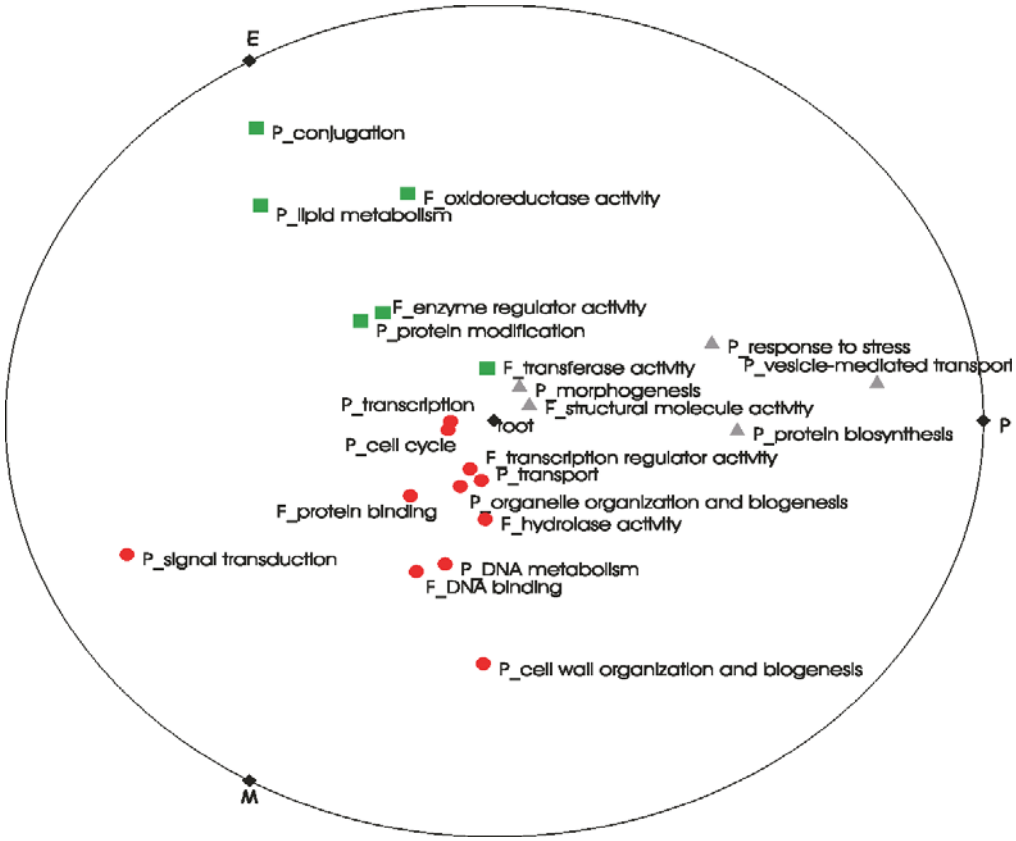


Slika A.1: Primer mreže so-izraženih genov, ki vključuje sedem genov (G1-G7). Povezani so le tisti pari genov, katerih korelacija v izmerjenem izražanju je nad izbranim pragom. Pokritost razreda je $2/5 = 0.4$, klasifikacijska točnost je $1/4 = 0.25$.

genov ter ovrednotili napovedno zmožnost mreže za napovedovanje različnih funkcij genov. Primerjali smo napovedne zmožnosti mrež za vsak tip podatkov in napovedano funkcijo genov. Absolutnega zmagovalca v teh primerjavah ni. Ugotovili smo, da so različni tipi genomskih podatkov bolj uspešni za napovedovanje določenih genskih funkcij. Tako smo pokazali, da je za uspešno analizo potrebno čim več različnih tipov podatkov. Slika A.2 prikazuje rezultate primerjave uspešnosti napovedovanja funkcije genov v obliki grafa RadViz. Iz slike je moč razbrati, katere genske funkcije se da bolje napovedovati na podlagi posameznega tipa podatkov. Točke bližje enemu izmed treh dimenzijskih sider (ang. *anchors*) na krožnici predstavljajo funkcije, ki se da bolje napovedovati z mrežo so-izraženih genov zgrajeno na podlagi računskega fenotipa, ki ga predstavlja sidro (E za profil izražanja genov, M za transkripcijski fenotip mutant, P za kvantitativni rastni fenotip mutant). Ocenili smo 28 različnih izbranih funkcijskih skupin (ang. *GO slim terms*) iz tako imenovane ontologije genov (ang. *Gene Ontology*). Uporabili smo skupine z vsaj desetimi pripisanimi geni [Ashburner et al., 2000].

A.4 Razvrščanje v skupine na podlagi pravil

Glavni prispevek disertacije je nov pristop k strojnemu učenju, tako imenovana metoda razvrščanja na podlagi pravil (ang. *rule-based clustering*). Odkrita pravila so oblike IF *simbolični opis primerov* THEN *profil primerov*. Pristop omogoča razvrščanje primerov oziroma genov v skupine, katerih člane lahko opišemo z nekim simboličnim zapisom, ki je zakodiran v pogojnem delu pravila,



Slika A.2: Uspešnost napovedovanja funkcije genov mrež so-izraženih genov, zgrajenih na podlagi različnih tipov podatkov (E – profil izražanja genov, M – transkripcijski fenotip mutant, P – kvantitativni rastni fenotip mutant). Uspešnost je merjena s površino pod krivuljo ROC. Barva točke prikazuje, kateri tip podatka najbolje napoveduje funkcijo, ki jo točka predstavlja (rdeč krog za M, zelen pravokotnik za E, siv trikotnik za P).

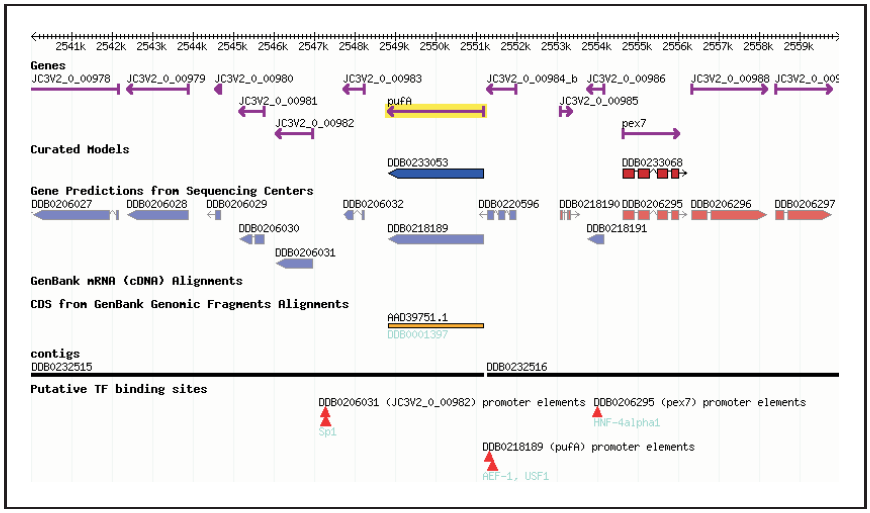
v zaključku pravila pa sledi opis fenotipa oziroma profila, ki je enak (oziroma podoben) za vse gene v skupini. Metoda zahteva dva nabora atributov. Na podlagi prve skupine atributov se izračuna razdalja med primeri. V naših analizah smo za razdaljo uporabili funkcijo Pearsonove korelacije. Drugo množico atributov se uporabi za gradnjo simboličnih opisov odkritih skupin. Predlagani algoritem za odkrivanje pravil uporablja metodo iskanja v snopu (ang. *beam search*) omejene velikosti L, podobno algoritmu CN2 [Clark and Nibblet, 1989]. Med iskanje je lahko v snopu shranjenih največ L pravil, ki opisujejo najbolj homogene trenutno odkrite skupine. Algoritem poskuša le-te dodatni izostriti. Metoda uporablja statistični F-test za izbor pravil in posredno za ustavitveni

pogoj. Začenši z enim, enostavnim pravilom 'True' v snopu, se nato vsako pravilo v snopu izostri z dodajanjem pogojev. Z uporabo F-testa se računa statistično značilnost spremembe variance med skupino pokrito s prvotnim pravilom in skupino pokrito z izostrenim pravilom. Algoritem teži k odkrivanju vedno bolj homogenih skupin, za kar smo se zgledovali po metodi Blockeel in sodelavcev [Blockeel et al., 1998], kjer pa gradijo drevesa za razvrščanje. V primeru značilne izostritve se pravilo doda v snop. V nasprotnem primeru se pravilo doda v končni seznam odkritih pravil, v katerem se hrani le K najboljših pravil, ki opisujejo najbolj homogene trenutno odkrite skupine. Pokriti primeri se ne odstranijo, kar omogoča odkrivanje prekrivajočih skupin; preiskovanje se ustavi, ko se snop izprazni. Za boljšo predstavitev in interpretacijo odkritih pravil predlagamo nabor vizualizacij. Prav tako predlagamo predstavitev rezultatov v obliki spletnih strani.

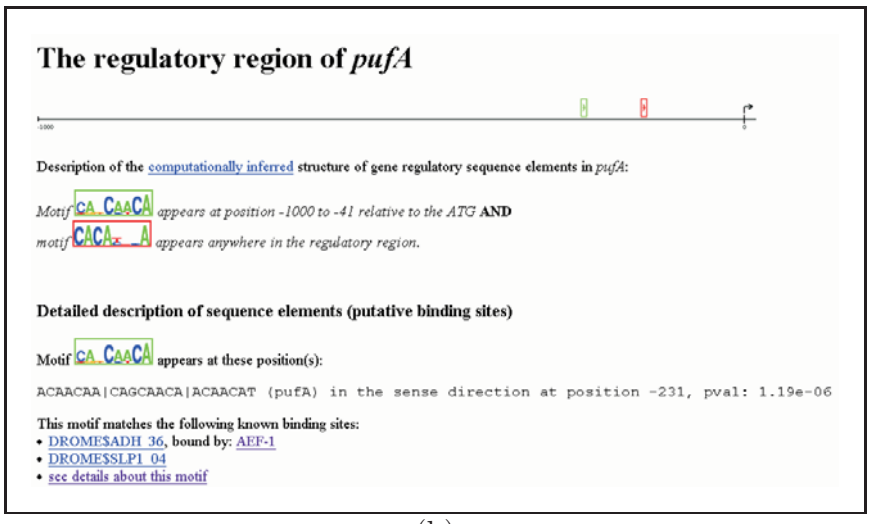
Metoda razvrščanja v skupine na podlagi pravil je bila prvotno razvita za analizo genetskih zapisov DNA regulatornih regij genov in podatkov o računskem fenotipu (n.pr., fenotipu mutanta, profilu izražanja genov, ipd.), z namenom iskanja skupin genov s podobnim fenotipom in strukturo regulatorne regije. Kljub temu je metoda splošno uporabna tudi za reševanje drugih, podobnih problemov.

Z uporabo standardnih pristopov strojnega učenja za ocenjevanje napovedne zmožnosti modelov smo eksperimentalno ovrednotili modele odkrite s predlagano metodo razvrščanja na podlagi pravil. Navajamo primere uporabe metode za razvrščanje na podlagi pravil za odkrivanje uravnalnih pravil izražanja genov glive sluzavke *Dictyostelium discoideum* in kvasovke *Saccharomyces cerevisiae*, za kar smo uporabili javno dostopne podatke o izmerjenem izražanju genov z DNA mikromrežami in podatke o zapisu DNA regulatornih regij genov. Z uporabo metode razvrščanja na podlagi pravil smo poskusili odgovoriti tudi na nekaj pomembnejših bioloških vprašanj: kateri so najbolj pomembni opisni elementi regulatornih regij in kateri del zapisa DNA regulatorne regije nosi največ informacije o uravnavanju genov. Rezultati pokažejo, da je najbolj informativen podatek o oddaljenosti veznih mest transkripcijskih faktorjev od mesta translacije genov. Najbolj informativna regija pa se razteza -900 do +600 baz relativno na začetek translacije gena (ATG), kar nakazuje, da je uravnalni program genov delno določen tudi v kodirajočem področju genov.

Metodo razvrščanja na podlagi pravil smo uporabili za obsežno analizo regu-



(a)



(b)

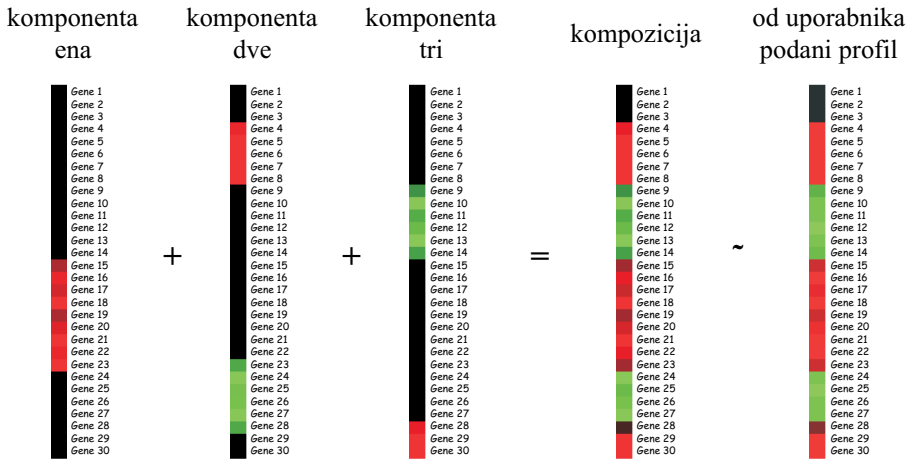
Slika A.3: a) Steza "Putative TF binding sites" v pregledovalniku genoma glive sluzavke *D. discoideum* (ang. *dictyBase Genome Browser*, na naslovu http://dictybase.org), s povezavami do b) spletnih strani s podrobnimi rezultati analize z metodo razvrščanja na podlagi pravil.

latornih regij genov glive sluzavke *D. discoideum.* Rezultati analize so dosegljivi širši raziskovalni skupnosti glive sluzavke na spletni strani http://bubble.fri.uni-lj.si/dictyBase ter preko uradne spletne strani o organizmu, na naslovu http://dictybase.org [Eichinger et al., 2005]. Primer strani je prikazan na Sliki A.3.

A.5 Dekompozicija profilov izražanja

Razvita metoda dekompozicije profilov izražanja genov omogoča modeliranje in sklepanje o genskih mrežah, ter tako nudi komplementaren pogled na rezultate dobljene z razvrščanjem na podlagi pravil. Uporabnik poda na vhod izmerjeni profil izražanja genov celotnega genoma, za kar metoda poišče komponente (podskupine genov in njihovo izražanje pod različnimi pogoji), ki združene skupaj v nov profil dobro opisujejo od uporabnika podan profil izražanja genov.

Algoritem za dekompozicijo potrebuje bazo izmerjenih profilov izražanja genov celotnega genoma (transkripcijskih profilov), izmerjene za različne mutante ali pod različnimi pogoji. Za iskanje najboljše dekompozicije algoritem izčrpno preišče vse kombinacije, ki vsebujejo od 1 do K komponent. Za neko kombinacijo se vsakemu genu določi pripadnost komponenti, v kateri je izražanje gena najbolj podobno podanemu izražanju gena (za primer glej Sliko A.4). To velja, če izvajamo dekompozicijo z uporabo funkcije "*min*," sicer pa je možno uporabiti poljubno funkcijo (vsota, utežena vsota, *itd.*). Ker lahko baza vključuje tudi več tisoč transkripcijskih profilov, algoritem za dekompozicijo izbere le N transkripcijskih profilov, ki so najbolj podobni od uporabnika podanemu profilu. Na izbrani množici potem izvede izčrpno preiskovanje kombinacij.



Slika A.4: Dekompozicija profilov izražanja genov. Po združitvi treh komponent ("ena," "dve" in "tri"), dobimo kompozicijo. Le-ta naj bo podobna od uporabnika podanemu profilu izražanja.

Rezultat dekompozicije je seznam kombinacij profilov v podani bazi, katerega je možno prikazati kot mrežo najbližjih mutantov in pogojev, iz katere lahko sklepamo o genskih poteh ter o v njih udeleženih genih. Algoritem za dekompozicijo je implementiran v oblik spletne aplikacije, dosegljive na naslovu http://bubble.fri.uni-lj.si/microCOMB. Navajamo tudi dva primera uspešne uporabe dekompozicije za odkrivanje genov, ki delujejo v paralelnih poteh genske mreže. Na podatkih glive sluzavke *D. discoideum* smo uspešno odkrili dekompozicijo izražanja mutanta gene *pkaC*-, izražanje genov katerega še najbolje opisuje izražanje genov mutant *yakA*- in *acaA*-, ki nastopata v vzporednih poteh, ter dokazano ključno vplivata na aktivnost gena *pkaC*. Drugi primer, za gensko mrežo MAPK gena *Hog* v kvasovki, prav tako kaže, da dekompozicija odkriva "biološki kontekst" (to je, ostale meritve iz baze, s katerimi je možno odpisati od uporabnika podano meritev) in tako omogoča sklepanje o genskih mrežah (za podrobnosti glej poglavje 7.3).

A.6 Zaključek in nadaljnje delo

Eksperimenti z mrežami so-izraženih genov, zgrajenimi na podlagi različnih (računskih) fenotipov, kažejo na to, da je za uspešno napovedovanje funkcije genov potrebno uporabiti (vse razpoložljive) različne tipe fenotipov. Nadaljnje delo na tem področju vključuje razvoj algoritmov, ki bodo zmožni učenja uporabe najustreznejšega fenotipa za napovedovanje posameznih funkcij genov.

Glavni prispevek disertacije je nova metoda strojnega učenja za razvrščanje v skupine na podlagi pravil. Metoda združuje preiskovanje v snopu metode CN2 [Clark and Nibblet, 1989], ter določene elemente gradnje prototipov in ocenjevanja homogenosti skupin metode dreves za razvrščanje [Blockeel et al., 1998]. Pokazali smo, da lahko metoda odkriva kompleksne, opisne in človeku razumljive modele, ki jih je možno uporabiti tudi za napovedovanje novih primerov. Obe lastnosti sta pomembni pri odkrivanju novega znanja iz podatkov. Iz navedenih primerov je tudi razvidno, da igra vizualizacija odkritih pravil pomembno vlogo pri razumevanju odkritih zakonitosti in pridobivanju novega znanja. Nadaljnje delo vključuje zmanjševanje števila parametrov in poenostavljanje zahtevanih parametrov algoritma. Za analizo uravnavanja izražanja genov, bi bilo zanimivo dodati in preizkusiti še druge elemente opisnega jezika, s katerimi bi bilo moč opisati strukturo kromatina, tako imenovane otoke CpG (ang. *CpG islands*),

sekundarno strukturo mRNA, ipd. Poleg tega, bi bilo v nadaljnjem delu potrebno preučiti kakšna je povezava med strukturo regulatorne regije genov ter drugih tipov računskih fenotipov (n.pr., transkripcijski profil mutantov).

Ker je metoda za razvrščanje na podlagi pravil dovolj splošna, bi jo bilo smiselno preizkusiti tudi na drugih domenah. Zelo aktualno področje je analiza različnih kemijskih učinkovin (zdravil). V tem primeru bi opisni jezik služil za opisovanje kemijske strukture učinkovin, profil primerov pa izmerjeno izražanje genov v celicah izpostavljenim učinkovini.

Predlagana dekompozicija profilov izražanja se je izkazala kot uporabna za dodatno odkrivanje relacij med geni ter postavitev uporabnikovega eksperimenta v neki biološki kontekst že znanih meritev izražanja genov. Metodo smo tudi implementirali kot spletno orodje. Nadaljnje delo na tem področju vključuje ovrednotenje različnih funkcij za dekompozicijo, ter razvoj časovno manj potratne hevristike za iskanje dobrih kombinacij komponent.

Izjava

Izjavljam, da sem doktorsko disertacijo z naslovom "Računski pristopi k odkrivanju genskih mrež" izdelal samostojno pod vodstvom mentorja prof. dr. Blaža Zupana. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Tomaž Curk