Contextual Analysis of Gene Expression Data

Florian Sohler

München 2006

Contextual Analysis of Gene Expression Data

Florian Sohler

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Florian Sohler
aus Dortmund

München, den 10.05.2006

Erstgutachter: Prof. Dr. Ralf Zimmer
Zweitgutachter: Prof. Dr. Martin Vingron
Tag der mündlichen Prüfung: 20. Juli 2006

Contents

List of Figures

List of Tables

Summary

As measurement of gene expression using microarrays has become a standard high throughput method in molecular biology, the analysis of gene expression data is still a very active area of research in bioinformatics and statistics. Despite some issues in quality and reproducibility of microarray and derived data (Michielis et al., 2005; Marshall, 2004), they are still considered as one of the most promising experimental techniques for the understanding of complex molecular mechanisms.

This work approaches the problem of expression data analysis using contextual information. While all analyses must be based on sound statistical data processing, it is also important to include biological knowledge to arrive at biologically interpretable results.

After giving an introduction and some biological background, in chapter 2 some standard methods for the analysis of microarray data including normalization, computation of differentially expressed genes, and clustering are reviewed. The first source of context information that is used to aid in the interpretation of the data, is functional annotation of genes. Such information is often represented using ontologies such as gene ontology (The Gene Ontology Consortium, 2001). GO annotations are provided by many gene and protein databases and have been used to find functional groups that are significantly enriched in differentially expressed, or otherwise conspicuous genes. In gene clustering approaches, functional annotations have been used to find enriched functional classes within each cluster. In chapter 3, a clustering method for the samples of an expression data set is described that uses GO annotations during the clustering process in order to find functional classes that imply a particularly strong separation of the samples. The resulting clusters can be interpreted more easily in terms of GO classes. The clustering method was developed in joint work with Henning Redestig.

More complex biological information that covers interactions between biological objects is contained in networks. Such networks can be obtained from public databases of metabolic pathways, signaling cascades, transcription factor binding sites, or high-throughput measurements for the detection of protein-protein interactions such as yeast two hybrid experiments. Furthermore, networks can be inferred using literature mining approaches or network inference from expression data. The information contained in such networks is very heterogenous with respect to the type, the quality and the completeness of the contained data. ToPNet, a software tool for the interactive analysis of networks and gene expression data has been developed in cooperation with Daniel Hanisch (Hanisch et al., 2004). The basic analysis and visualization methods as well as some important concepts

of this tool are described in chapter 4.

In order to access the heterogeneous data represented as networks with annotated experimental data and functions, it is important to provide advanced querying functionality. *Pathway queries* (Sohler et al., 2004; Sohler and Zimmer, 2005) allow the formulation of network templates that can include functional annotations as well as expression data. The *pathway search* algorithm finds all instances of the template in a given network. In order to do so, a special case of the well known subgraph isomorphism problem has to be solved. Although the algorithm has exponential running time in the worst case, some implementation tricks make it run fast enough for practical purposes. Often, a *pathway query* has many matching instances, and it is important to assess the statistical significance of the individual instances with respect to expression data or other criteria. In chapter 5 the *pathway query language* and the *pathway search* algorithm are described in detail and some theoretical properties are derived. Furthermore, some scoring methods that have been implemented are described. The possibility of combining different scoring schemes for different parts of the query result in very flexible scoring capabilities.

In chapter 6, some applications of the methods are described, using public data sets as well as data sets from research projects. On the basis of the well studied public data sets, it is demonstrated that the methods yield biologically meaningful results. The other analyses show how new hypotheses can be generated in more complex biological systems, but the validation of these hypotheses can only be provided by new experiments.

Finally, an outlook is given on how the presented methods can contribute to ongoing research efforts in the area of expression data analysis, their applicability to other types of data (such as proteomics data) and their possible extensions.

Zusammenfassung

Während die Messung von RNA-Konzentrationen mittels Microarrays eine Standardtechnik zur genomweiten Bestimmung von Genexpressionswerten geworden ist, ist die Analyse der dabei gewonnenen Daten immer noch ein Gebiet äußerst aktiver Forschung. Trotz einiger Probleme bezüglich der Reproduzierbarkeit von Microarray- und davon abgeleiteten Daten werden diese als eine der vielversprechendsten Technologien zur Aufklärung komplexer molekularer Mechanismen angesehen.

Diese Arbeit beschäftigt sich mit dem Problem der Expressionsdatenanalyse mit Hilfe von Kontextinformationen. Alle Analysen müssen auf solider Statistik beruhen, aber es ist außerdem wichtig, biologisches Wissen einzubeziehen, um biologisch interpretierbare Ergebnisse zu erhalten.

Nach einer Einleitung und einigem biologischen Hintergrund werden in Kapitel 2 einige Standardmethoden zur Analyse von Expressionsdaten vorgestellt, wie z.B. Normalisierung, Berechnung differenziell exprimierter Gene sowie Clustering. Die erste Quelle von Kontextinformationen, die zur besseren Interpretation der Daten herangezogen wird, ist funktionale Annotation von Genen. Solche Informationen werden oft mit Hilfe von Ontologien wie z.B. der Gene Ontology (The Gene Ontology Consortium, 2001) dargestellt. GO Annotationen werden von vielen Gen- und Proteindatenbanken zur Verfügung gestellt und werden unter anderem benutzt, um Funktionen zu finden, die signifikant angereichert sind an differenziell exprimierten oder aus anderen Gründen auffälligen Genen. Bei Clusteringmethoden werden funktionale Annotationen benutzt, um in den gefundenen Clustern angereicherte Funktionen zu identifizieren. In Kapitel 3 wird ein neues Clusterverfahren für Proben in Expressionsdatensätzen vorgestellt, das GO Annotationen während des Clustering benutzt, um Funktionen zu finden, anhand derer die Expressionsdaten besonders deutlich getrennt werden können. Die resultierenden Cluster können mit Hilfe der GO Annotationen leichter interpretiert werden. Die Clusteringmethode wurde in Zusammenarbeit mit Henning Redestig entwickelt.

Komplexere biologische Informationen, die auch die Interaktionen zwischen biologischen Objekten beinhaltet, sind in Netzwerken enthalten. Solche Netzwerke können aus öffentlichen Datenbanken von metabolischen Pfaden, Signalkaskaden, Bindestellen von Transkriptionsfaktoren, aber auch aus Hochdurchsatzexperimenten wie der Yeast Two Hybrid Methode gewonnen werden. Außerdem können Netzwerke durch die automatische Auswertung wissenschaftlicher Literatur oder Inferenz aus Expressionsdaten gewonnen werden. Die Information, die in solchen Netzwerken enthalten ist, ist sehr verschieden in Bezug

auf die Art, die Qualität und die Vollständigkeit der Daten. ToPNet, ein Computerprogramm zur interaktiven Analyse von Netzwerken und Genexpressionsdaten, wurde gemeinsam mit Daniel Hanisch entwickelt (Hanisch et al., 2004). Die grundlegenden Analyse- und Visualisierungsmethoden sowie einige wichtige Konzepte dieses Programms werden in Kapitel 4 beschrieben.

Um auf die verschiedenartigen Daten zugreifen zu können, die durch Netzwerke mit funktionalen Annotationen sowie Expressionsdaten repräsentiert werden, ist es wichtig, flexible und mächtige Anfragefunktionalität zur Verfügung zu stellen. *Pathway queries* (Sohler et al., 2004; Sohler and Zimmer, 2005) erlauben die Beschreibung von Netzwerkmustern, die funktionale Annotationen sowie Expressionsdaten enthalten. Der *pathway search* Algorithmus findet alle Instanzen des Musters in einem gegebenen Netzwerk. Dazu muss ein Spezialfall des bekannten Subgraph-Isomorphie-Problems gelöst werden. Obwohl der Algorithmus im schlechtesten Fall exponentielle Laufzeit in der Größe des Musters hat, läuft er durch einige Implementationstricks schnell genug für praktische Anwendungen. Oft hat eine *pathway query* viele Instanzen, so dass es wichtig ist, die statistische Signifikanz der einzelnen Instanzen in Hinblick auf Expressionsdaten oder andere Kriterien zu bestimmen. In Kapitel 5 werden die Anfragesprache *pathway query language* sowie der *pathway search* Algorithmus im Detail vorgestellt und einige theoretische Eigenschaften gezeigt. Außerdem werden einige implementierte Scoring-Methoden beschrieben. Die Möglichkeit, verschiedene Teile der Anfrage mit verschiedenen Scoring-Methoden zu bewerten und zu einem Gesamtscore zusammenzufassen, erlaubt äußerst flexible Bewertungen der Instanzen.

In Kapitel 6 werden einige Anwendungen der vorgestellten Methoden beschrieben, die auf öffentlichen Datensätzen sowie Datensätzen aus Forschungsprojekten beruhen. Mit Hilfe der gut untersuchten öffentlichen Datensätze wird gezeigt, dass die Methoden biologisch sinnvolle Ergebnisse liefern. Die anderen Analysen zeigen, wie neue Hypothesen in komplexeren biologischen Systemen generiert werden können, die jedoch nur mit Hilfe von weiteren biologischen Experimenten validiert werden könnten.

Schließlich wird ein Ausblick gegeben, was die vorgestellten Methoden zur laufenden Forschung im Bereich der Expressionsdatenanalyse beitragen können, wie sie auf andere Daten angewendet werden können und welche Erweiterungen denkbar und wünschenswert sind.

Chapter 1

Introduction and Concepts

1.1 Introduction

The recent advent of new high-throughput experimental techniques has changed research in molecular biology in many aspects. Whole-genome sequence data for many organisms, and most importantly for human, have become available as well as a mass of data on gene expression and protein-protein interactions. Other high-throughput methods, for instance for the detection of protein-DNA interactions or silencing of genes via RNA interference are being established.

These new technologies have been connected with many promises regarding human health. A better understanding of complex diseases as well as improved diagnostic possibilities and individualized treatments have been envisioned. But so far, most of these promises could not be fulfilled as the underlying biological systems have eluded a comprehensive analysis due to the prohibitive complexity of these systems and their dynamics.

The analysis of the new wealth of data poses new challenges to biological and pharmaceutical researchers. While the results of traditional small-scale experiments can be analyzed manually by biological experts, large-scale experiments require at least some automatic preprocessing to extract and highlight important aspects of the resulting data. The manual evaluation of experimental data has the advantage that the same researchers who designed and conducted the experiment can perform the analysis of the results. This ensures that no false assumptions about the experimental set-up are made and that detailed background knowledge about the relevant biology is available for the analysis. Computer programs carrying out at least part of the analysis usually cannot take advantage of such background knowledge. The basic assumption of this work is that in order to gain new insights from biological experiments, background knowledge about the investigated biological questions is as important as the collected data themselves. Thus, the goal is to bring biological data and background knowledge together in computational approaches. The first questions that have to be answered include:

- How can biological knowledge be represented to make it accessible for computational analyses?

- What sources for biological knowledge are available?
- How can different sources of biological knowledge and data be integrated?
- How can a researcher include his own knowledge and expectation in an automatic analysis?

Knowledge representation is an important topic in other fields of computer science like databases, data mining and artificial intelligence. Knowledge bases, as they are used in logic programming, are composed of facts and rules. Using different principles of logical reasoning, new facts can be derived, or explained by the given rules. As it is difficult to deal with the uncertainties and incompleteness of biological knowledge using such classical knowledge bases, their use in molecular biology has been restricted to rather small examples so far (Tran et al., 2005; Baral et al., 2004; Zupan et al., 2003).

The so-called semantic web is another field of computer science that deals with knowledge representation. The goal of the semantic web is to annotate documents in the world wide web with semantic information and provide tools for the processing of this information by computers. The knowledge representation language proposed for the semantic web is the resource definition framework (RDF). Many of the techniques and tools developed for the semantic web could have applications in bioinformatics.

Ontologies constitute another important means to represent knowledge; they formalize a certain domain by describing all relevant entities and relationships in that domain as well as rules on these entities and relationships, therefore providing a controlled vocabulary as well as the possibility to reason about terms from the ontology. Gene Ontology (GO) is widely used in computational biology (The Gene Ontology Consortium, 2001). It contains descriptions of the function of a gene product and connects these functions using two different relationships, is-a and part-of. GO has three different parts: biological process, molecular function and cellular component, therefore these three different aspects of a gene product's function can be described using terms from GO.

But functional annotations cover only a small part of the available biological knowledge. Relationships between genes or proteins such as protein-protein interactions also constitute relevant information. Such information can be represented in networks of genes or proteins, for instance as metabolic networks, signaling pathways or protein interaction networks. In order to describe the relationships between biological entities within a network, again ontologies or at least some controlled vocabularies are needed.

After we have seen some possible representations of biological knowledge, the next question is, if such knowledge is already available and which formats are actually used by biologists and bioinformaticians. Many gene and protein databases like Swissprot[1] (Boeckmann et al., 2003) or Entrez Gene[2] (Maglott et al., 2005) contain GO annotations for the objects described in those databases. And there are also databases of pathways and networks that contain interaction information for genes and proteins and describe

[1] `http://www.expasy.ch/sprot/`

[2] `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene`

these interactions in various levels of detail. Network formats range from tabular files with simple controlled vocabularies to complex description languages, allowing to specify complex issues like the conditions for an interaction to occur, the experiment that was used to find that interaction, or kinetic parameters.

Another important source of information is the collected scientific literature. The Medline database contains abstracts of millions of scientific publications in the area of medical and biological research. If it was possible to automatically extract information from these abstracts, this would constitute an extremely rich and detailed source of biological knowledge. Therefore text mining has become an important discipline in bioinformatics as can be seen at international conferences where separate sessions on text mining are often employed. Text mining basically addresses the problem of generating formal representations of knowledge that is given in free text form, thus making it accessible for automatic analysis systems. Specific problems range from the identification of biological entities in free text (Hanisch et al., 2003; Krauthammer et al., 2000) over the extraction of interactions and relationships between such entities (Daraselia et al., 2004) to the extraction of correlations that are not explicit in a single document but can be inferred from many documents (Smalheiser and Swanson, 1998; Srinivasan and Libbus, 2004). Results of text mining tools are often represented as networks, especially when the goal was the identification of interactions; but networks can also be generated from occurrences of biological entities in texts. In that case an edge is introduced between two objects if they occur together in a sentence or an abstract of a publication. We will call such networks co-occurrence networks.

In practice, biological background knowledge is usually available for computational purposes in gene, protein or network databases and is represented using various standard and non-standard formats. These formats apply ontologies, controlled vocabularies or even free text annotations. Such background knowledge in its currently available representation is the basis of this thesis. The integration of different sources of background knowledge as biological context for the interpretation of experimental data is the main goal.

On the one hand, the integration task poses technical problems like cross-linking databases or identifying redundancies in different databases. For some of these problems approaches were developed and implemented for this thesis, but this will not be discussed in detail. On the other hand, there are also interesting problems concerning the analysis of the resulting heterogeneous data sets. The main emphasis of this work lies on this second type of problems.

It is also assumed that data analysis depends strongly on what the researcher is actually interested in. Therefore, fully automatic algorithms can only deliver solutions for rather special problems. Semi-automatic tools are needed where the user can direct the analyses performed into a direction of interest.

In this thesis, the data under investigation are restricted to gene expression data; network data (from curated databases as well as high-throughput experiments and text mining) and functional annotations are considered as context information. Gene expression data play a special role in molecular biology because they can be collected cheaply using microarrays and provide a genome-wide snapshot of a cell population under a certain experimental condition. Genome-scale measurements of protein levels are still infeasible,

therefore microarrays provide a unique look into the state of a cell.

Early prominent applications of microarray technologies include Spellman et al. (1998) who identified cell-cycle regulated genes by expression profiling or DeRisi et al. (1997) who analyzed the regulation of the yeast metabolism during the diauxic shift, i.e. during the shift from fermentation to respiration that occurs when glucose becomes unavailable to the yeast. Both works could provide hints for the function of many previously uncharacterized genes. These early successes increased the popularity of microarrays significantly. Today, microarrays are used routinely in biological and pharmaceutical research in order to measure gene expression levels on a genomic scale under different experimental conditions.

The analysis of gene expression data is a very active area of research in bioinformatics and statistics. While some basic analyses, like normalization and calculation of fold changes and differentially expressed genes, can be performed without background knowledge, more detailed analyses require the inclusion of such knowledge. After some basic analysis steps, the resulting lists of statistically relevant features is often returned to the researcher who then tries to answer biological questions using these processed data. Such questions could cover the biological mechanism that causes the observed regulation as well as the effect that it has on certain biological processes like the metabolism of the organism under study. For the first question, it would be useful to investigate the data in context of a model for gene regulation. If, for instance, pathway models along with their target genes are known, it is possible to come up with hypotheses about the pathways involved in an observed pattern of gene regulation. For the second question a model of the metabolism should provide valuable insight. For instance, it is possible to identify metabolites whose synthesis or degradation could be defective by looking at the expression levels of enzymes needed for the corresponding reactions.

In summary, this thesis is concerned with the analysis of gene expression data under the following two assumptions:

1. **Context knowledge must be included in an advanced analysis of gene expression data.**

2. **The researcher must be able to direct the computational analysis toward specific questions about the data and possible conclusions.**

We consider context in the form of functional annotations to genes and proteins and in the form of networks that specify interactions between genes and proteins. The second assumption requires interactivity, therefore most methods have been implemented within an interactive tool for network analysis, called ToPNet – Toolbox for protein networks. Furthermore, *pathway queries* are developed as a method specifically designed to let the user look at certain aspects of the data by specifying network templates that represent relevant context for the research question under investigation. *Pathway queries* are also implemented within ToPNet, thus allowing further exploration of the results. With these approaches we address the fact that different questions require different views on the measured data. If in pharmaceutical research the goal is to find a possible target gene for novel disease-modifying drugs, it will be of high interest to understand the mechanism

that causes the observed altered regulation in a disease. This goal requires looking at regulatory molecules and mechanisms like transcription factors, signaling molecules and pathways. In a more fundamental investigation on the metabolism of a model organism, it could be interesting to understand changes in the metabolism that result from the observed regulation. Here, knowledge about enzymes and metabolic pathways will be very helpful. These examples demonstrate that the incorporation of background knowledge is crucial for the specific problems arising in the analysis of expression data.

1.2 Biological background

In order to motivate and explain many methods in this work, a basic understanding of some concepts of molecular biology is required. First, the most important biological entities are introduced, then the central dogma of molecular biology is explained, and finally a few examples of biological regulation are discussed that should give an impression of the complexity of the biological systems under consideration.

1.2.1 Biological Entities

Deoxyribonucleic acid (DNA) is a molecule that comprises a long chain of nucleotides which in turn are composed of a nucleobase, the sugar deoxyribose and a phosphate. The nucleobases can be cytosine, thymine, adenine and guanine, the two former called pyrimidines and the latter purines. Usually, DNA is found in a double-stranded form where each nucleotide binds its complementary nucleotide according to the pairing rule: adenine binds thymine and guanine binds cytosine and vice versa. The DNA molecule forms the well-known double-helix structure as discovered by Watson and Crick (1953). The complete genetic information of an organism is stored in every cell nucleus in the form of double-stranded DNA that is coiled up to build the chromosomes. For computational purposes, DNA can be viewed as a long string from the alphabet $\{A, C, G, T\}$, denoting the four nucleobases.

Ribonucleic acid (RNA) is similar to DNA, but it contains ribose as sugar and usually exists in single-stranded form. Furthermore the nucleobase thymine is replaced by uracil. RNA can fulfill different functions; we will be mostly concerned with messenger RNA (mRNA) which transports genetic information within the cell. The function of mRNA will be explained in detail in section 1.2.2.

Genes still elude a general definition. We will consider a gene a piece of DNA in the genome that is transcribed to build a functional product. Such a product can be functional RNA or a protein. The definition of genes becomes quite problematic when start and end of a gene should be defined because there can be different gene products starting and ending at different positions in the genome.

Proteins are another type of chain molecules, the constituting elements being the 20 naturally occurring amino acids. Proteins form complex and versatile three dimensional structures, giving rise to many functions. They can for instance function as structural constituents of the cell, enzymes or signaling molecules. Proteins can interact with other proteins, with RNA, DNA or metabolites. They constitute more than half of a cell's dry weight and carry out most cellular functions.

Pathways comprise molecules that work together in a defined spatio-temporal order to carry out a particular function. Metabolic pathways contain enzymes and their substrates and products and describe how metabolites are consumed and produced. Regulatory pathways contain signaling proteins that transduce a signal to turn on or off certain cellular functions, for instance the transcription of particular genes.

The cell is the basic functional unit of higher organisms. It contains all necessary parts to produce proteins, be metabolically active, and communicate with other cells. A cell is separated from its environment by the plasma membrane, which is a lipid bilayer membrane. The membrane is the communication channel for a cell, it contains proteins that take up or release metabolites from or into the extracellular space, and other proteins that recognize signaling molecules from other cells. That way, cells can react to changes in their environment. Inside of the membrane, the cell contains various organelles serving different purposes.

In principle, every cell is built up from the same ingredients. All information necessary for any cellular function is encoded in the genome that is located in the nucleus of each cell. But still, cells differentiate irreversibly in order to serve special purposes. In fact, cells differ in morphology, metabolism, gene expression and protein production. In terms of regulation and dynamics, two cell types from one mammal can be as different as two unrelated bacteria.

1.2.2 The Central Dogma of molecular biology

The *central dogma of molecular biology* states that DNA is transcribed to RNA which is in turn translated to form proteins. Figure 1.1 gives a schematic overview of the central dogma and, additionally, the DNA replication, which is central for cell division. In the first step of the protein production, DNA is transcribed multiple times by a multi-protein complex known as RNA-polymerase. The resulting mRNA is translated into a chain of amino acids at the ribosome, a cellular structure that is formed by ribosomal RNA and proteins. Translation obeys the genetic code, which is universal across all species and assigns each triplet of nucleotides one amino acid. On a molecular level, the genetic code is realized with tRNAs, RNA molecules that can recognize nucleotide triplets and bring the corresponding amino acid to the ribosome where it is attached to the growing protein chain.

The three most well known *-omics* disciplines in bioinformatics, namely genomics, transcriptomics and proteomics are related to the three stages mentioned by the central

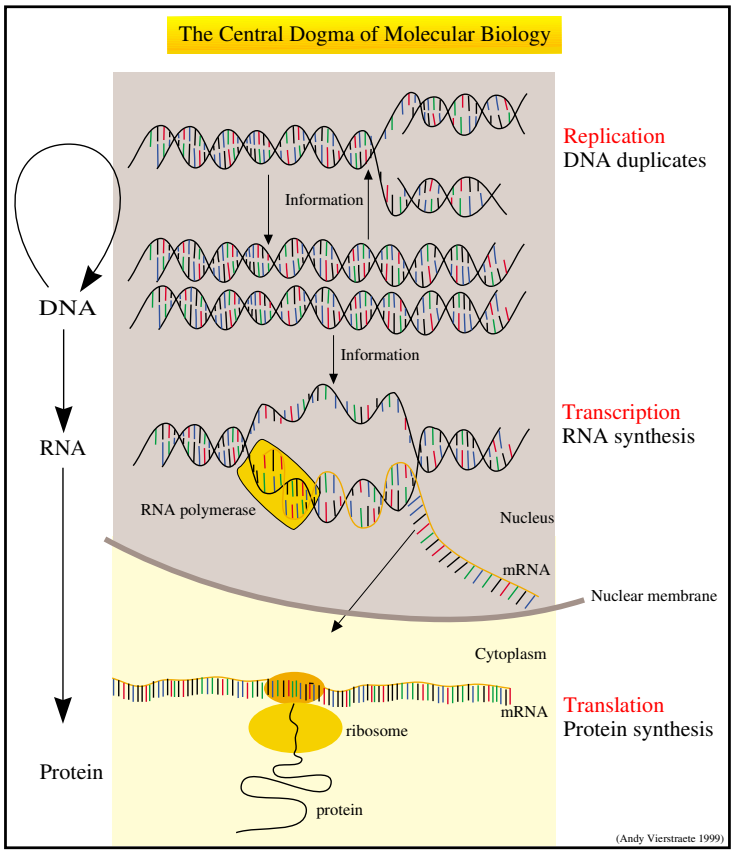

Figure 1.1: The central dogma of molecular biology.

dogma. Genomics is concerned with the sequence and structure of the genome, i.e. the DNA, transcriptomics covers the investigation of transcript data (mRNA), and finally, proteomics deals with the analysis of the proteome, the static description and dynamic behavior of proteins that can be found in an organism.

As proteins are the main active players in most cellular processes, it is their working and regulation that we need to understand ultimately. If the translation of mRNA to protein was one-to-one, proteomics would be much easier. The number of mRNA molecules of a particular gene would be a precise indicator of the number of proteins after translation (assuming also that there is no regulation of protein degradation). Unfortunately, there are several mechanisms for the regulation of protein concentration and activity before and after the transcription step. Thus, all *-omics* have to play together to gain a deep understanding of the regulation of cellular functions. In order to demonstrate that necessity, we give some examples of known regulation mechanisms to demonstrate the complexity of the cellular system.

1.2.3 Regulation mechanisms

DNA methylation: DNA methylation describes the 'tagging' of nucleotides with a methyl group. In mammals, these methyl tags are usually attached to cytosine nucleotides. Methylated DNA can silence transcription in at least two ways: By direct interference with transcription factors whose binding sites contain cytosines or indirectly by recruiting methyl-binding proteins that also act as transcriptional repressors (Scarano et al., 2005).

Chromatin structure and modification: Throughout most of the cell cycle, DNA is organized in tightly packed chromatin. Chromatin is made up of nucleosomes, consisting of a DNA stretch of 147 base pairs wrapped around a core of eight histone proteins. These nucleosomes bind each other, linker histones, or DNA, to form fibers which again fold into higher order structures. Histones are modified by different proteins, e.g. histone acetyltransferases. This modification influences the structural properties of the chromatin and can lead to the unfolding of chromatin fibers, making genes and their promoter regions accessible to the transcriptional machinery of the cell. Horn and Peterson (2002) review the current knowledge on chromatin structure and its influence on transcription. Another review on chromatin remodeling and relations to transcription factor binding sites is given in Urnov (2003).

Transcription factors: The regulation of transcription is controlled by cis-acting DNA elements and trans-acting factors, of particular importance are transcription factors. The definition of a transcription factor given in the TRANSFAC database (Wingender et al., 2000) is as follows: *A transcription factor is a protein that regulates transcription (after nuclear translocation) by sequence-specific interaction with DNA or by stoichiometric interaction with a protein that can be assembled into a sequence-specific DNA-protein complex.* The RNA-polymerase, which constitutes

the general transcription machinery, binds rather unspecifically and with low affinity to DNA. Transcription factors are required to locate the RNA polymerase to the promoter regions of genes that should be transcribed. Transcription factors can also act as repressors by blocking promoter sites for the RNA polymerase or other activating transcription factors. Often, many transcription factors, both activating and repressing, regulate the transcription of a gene coordinately. An example are the ternary complex transcription factors (TCFs) (Buchwalter et al., 2004). These proteins can form a complex with the serum response factor (SRF) over the serum response element (SRE), a sequence motif that can be found in the c-fos promoter. Conformational changes of the TCFs induced by phosphorylation as well as interactions with other proteins can induce activation or shut-down of target genes. Transcription factors not only bind DNA at the promoter region in the proximity of the transcription start site, but also at sites far away from the gene, the so-called enhancers. Enhancers usually contain many binding sites for different transcription factors. Important for the detailed regulation by enhancers are also the boundary elements which can block the influence of enhancers on promoter sites. A review on enhancers and boundary elements can be found in Blackwood and Kadonaga (1998) and Bell et al. (2001), respectively.

Alternative splicing: After transcription, the pre-mRNA is processed to remove non-coding parts, the so-called introns. This process is called splicing. In some cases not all of the coding regions, the exons, are kept in the transcript. They may be removed or even reordered. If there is more than one possible final product, this is called alternative splicing, which is also subject to regulation.

MicroRNAs: Recently, so-called microRNAs have received some attention as another type of regulatory molecule. MicroRNAs are transcribed from DNA that does not code for proteins, and regulate protein expression through inhibition of translation or degradation of mRNAs. It is speculated that they play a considerable role for the regulation of higher organisms (Bartel, 2004). For instance, it has been shown that they are essential in the development of Drosophila (Leaman et al., 2005).

Translation: A regulation mechanism that is not very well known is the regulation of the translation which can be accomplished by proteins binding to transcripts of other genes. This can be found for instance in the Drosophila embryo, where Bicoid and Nanos, two proteins encoded by maternally provided transcripts, bind mRNA from caudal and hunchback, respectively, and thus, regulate the spatial distribution of those proteins.

Protein modification: There are several forms of protein modification such as phosphorylation, ubiquitination or cleavage. Phosphorylation of proteins often leads to their activation, while ubiquitination (the binding of ubiquitin) serves as a signal that starts the degradation process of the marked protein.

Protein localization: Some proteins are only functional at certain cellular locations. E.g. the transcription factor NF-κB (prototypically consisting of the two subunits p50, encoded by NFKB1, and p65/RELA) in its inactive form is located in the cytoplasm of a cell and bound by IκBα. When IκBα is phosphorylated, it gets ubiquitinated and degraded, thus releasing the NF-κB complex which is then translocated to the nucleus where it can transcribe various target genes (Chen and Greene, 2004).

Signaling: Signaling pathways are regulatory units that make use of many of the mechanisms already discussed. Figure 1.2 shows the IL1 receptor signaling pathway. This pathway contains instances of phosphorylation, protein cleavage, translocation, and finally DNA-binding of transcription factor complexes.

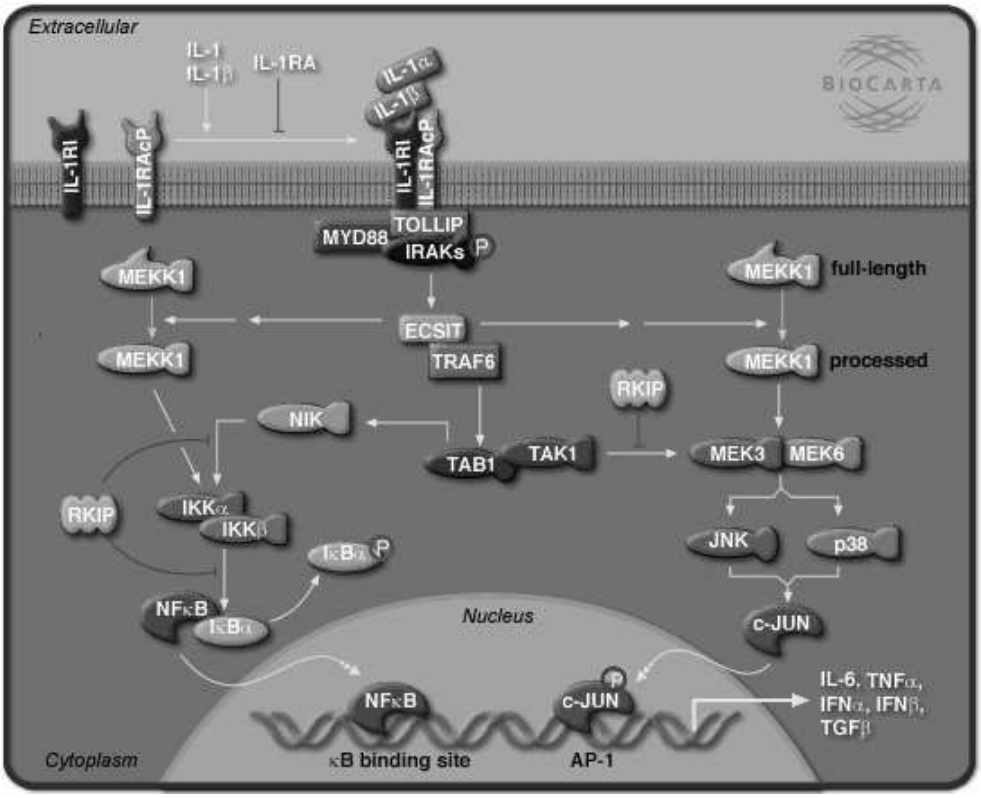

Figure 1.2: The IL1 receptor pathway as found at Biocarta (`http://www.biocarta.com`).

The regulatory mechanisms described here are interconnected in many ways. The first four examples all influence transcription, the others are more concerned with the translation or the activity of proteins, but this activity can often result in transcriptional effects again, e.g. by producing metabolites which in turn induce a transcriptional response. Signaling of course influences transcription directly again, as the activated proteins are often transcription factors that activate or inhibit the transcription of target genes.

1.3 Data used in this thesis

As explained in section 1.1, this thesis deals with the analysis of gene expression data in the context of functional annotations and biological networks. These two data types will be introduced in this section.

1.3.1 Gene expression data

The term gene expression data refers to the measured abundances of mRNA of a subset or all genes in the genome of an organism. Such measurements are usually performed using microarrays, a description of the experimental techniques will be given in chapter 2. As it was discussed in section 1.2.3, mRNA abundances only reflect one out of many mechanisms that are important in the regulation of biological processes in a living cell. Through the advances of the microarray technology, collecting gene expression data on a genome-wide scale has become affordable, providing a unique possibility to gain insight into a cell's state.

For computational purposes, expression data can be viewed as a simple matrix, where rows correspond to genes and columns to samples. A matrix element contains the measured expression value or a derived statistic for the corresponding gene and sample.

1.3.2 Biological networks and annotations

We consider any collection of relationships between biological entities a biological network. Most common examples are protein-protein interaction networks, metabolic networks or genetic regulatory networks. While the first two types describe physical relationships, a regulatory network can also contain more abstract interactions. A relation like 'gene A regulates gene B' does not necessarily imply any physical interaction between the genes or their products. Even more abstract networks are co-occurrence networks derived from the literature where an interaction between two biological objects means that the names of these objects have been found together in a scientific publication.

Metabolic networks describe the metabolism of an organism in terms of metabolic reactions and participating enzymes and metabolites. Metabolic networks are usually divided into pathways, which contain all reactions required for a certain biological process, such as the biosynthesis of an amino acid or the production of fatty acids.

Protein-protein interaction networks contain information on physical interactions between proteins. Such networks can be assembled manually from results of small-scale experiments, as they are presented in the scientific literature, in which case the quality of the data is usually very high. Otherwise, the information may come from one of the different high-throughput methods that are available today. The probably most prominent of such methods is the yeast two-hybrid system (Fields and Song, 1989; Fields and Sternglanz, 1994). This method uses a transcription factor (usually GAL4) that is composed of a binding domain (BD) and an activation domain (AD) to construct two fusion proteins as depicted in Figure 1.3. If two proteins X and Y

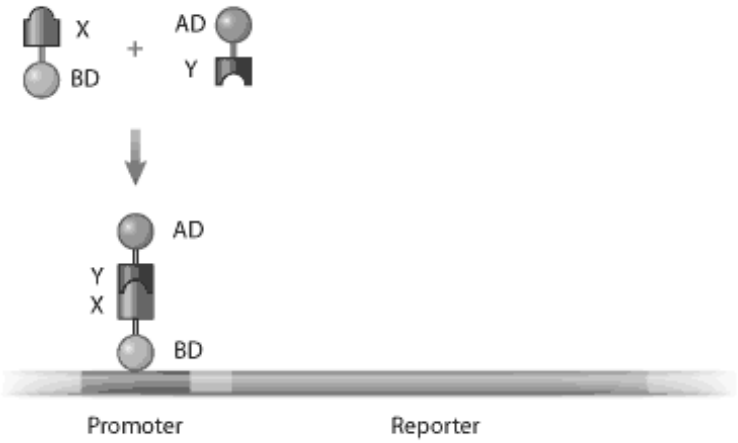

Figure 1.3: The yeast two-hybrid system. Only if the two proteins X and Y bind to each other, a functional transcriptional activator containing the binding domain (BD) and the activation domain (AD) is formed and can transcribe the reporter gene.

should be tested for an interaction, in the first fusion protein X is attached to BD and in the second Y is attached to AD. This construct ensures that BD and AD can interact and induce transcription if X and Y bind to each other. Thus, the reporter gene for the transcription factor is only transcribed if there is an interaction between X and Y.

Another important high-throughput technique for the identification of protein-protein interactions, and especially complex formation, is the tandem affinity purification (TAP). TAP was introduced by Rigaut et al. (1999) and is combined with mass spectrometry in order to purify and identify protein complexes at natural expression levels.

Protein-DNA interaction networks capture information on gene regulation mediated by the binding of regulatory proteins to DNA. Such networks can be derived from so-called genome-wide location analysis or ChIP on chip data. ChIP on chip means Chromatin Immunoprecipitation (Orlando, 2000) on microarrays (DNA chips) and was first described in Ren et al. (2000). The idea is to induce fixation of bindings of proteins to DNA using formaldehyde cross-linking and extract protein-DNA complexes after sonification by immunoprecipitation with antibodies directed against the proteins of interest. Next, the crosslinks are released and the DNA is amplified, labeled and hybridized on a microarray together with DNA that was not enriched by immunoprecipitation. That way, spots containing sequences that are often bound by the proteins of interest will appear overrepresented on the microarray.

1.4 Definitions and notation

1.4.1 Statistical tests and significance

For many applications in computational molecular biology, statistical tests are used, and their significance must be computed. Usually, such a significance is given in the form of a p-value which denotes the probability of an observed event under a null model. For instance, a p-value for differential expression of a gene under two different conditions is usually defined as the probability of observing some measured data in the two conditions, given that the gene is expressed at the same level under both conditions. In order to compute such a probability, a probabilistic null model must be defined that captures fluctuations of the measurement process in a reasonable way. A low p-value casts doubt on the null hypothesis. In our example, this could mean that the two genes are indeed differentially expressed, i.e. have different levels of expression in the two conditions. On the other hand, it could also mean that the null model does not appropriately reflect the experimental conditions. There could be an experimental bias that has not been accounted for.

Following Casella and Berger (2002), p-values can be defined mathematically as a certain kind of test statistic:

Definition 1.4.1. Two complementary hypotheses in a hypothesis testing problem are called the *null hypothesis* and the *alternative hypothesis*. They are denoted as H_0 and H_1, respectively.

Let X be a test statistic that is distributed with P_0 under the null hypothesis H_0. A *p-value* $p(X)$ is a test statistic satisfying $0 \leq p(x) \leq 1$ for every sample point x. Small values of $p(X)$ give evidence that H_1 is true. A p-value is *valid* if, under H_0 for every $0 \leq \alpha \leq 1$,

$$P_0(p(X) \leq \alpha) \leq \alpha. \tag{1.1}$$

Two tests that will be used several times in this work, are Fisher's exact test and the Mann-Whitney-Wilcoxon test. Both can be used to test the enrichment of certain features within a subset of observations. A third test that is commonly used, e.g. for the computation of differentially expressed genes, is the student's t-test or a variation thereof.

Fisher's exact test

Fisher's exact test (FET) is most often used in the context of 2x2 contingency tables, i.e. with two random variables which are both two-valued. The test is applied to identify dependencies between the two variables. The method was first proposed by Fisher (1932) and has been used in a wide range of settings.

For our purposes, FET can be nicely explained using an urn model. Consider an urn containing k red and $N - k$ black balls. Now, n balls are drawn from the urn without replacement. In the null model, the drawing is done without 'looking', i.e. without knowing anything about the color of the drawn balls. Therefore, the number S of red balls drawn

should be distributed hypergeometrically:

$$P(S = s) = \frac{\binom{N-k}{n-s}\binom{k}{s}}{\binom{N}{n}} \tag{1.2}$$

The numerator gives the number of possibilities to draw $n-s$ black balls times the number of possibilities to draw s red balls. The denominator counts the total number of possibilities to draw n from N balls.

Large values of S cast doubt on the null hypothesis, as it is unlikely to draw many red balls without knowing anything about the color in advance. A one-sided p-value connected to FET is the probability that S is greater or equal the observed value:

$$p(s) = P(S \geq s) = \sum_{i=s}^{min\{n,k\}} P(S = i) \tag{1.3}$$

In many cases, it is more intuitive to consider FET as a test that assesses the size of the overlap of two subsets. Given a set M and two subsets T_1 and T_2, the significance of the size of their intersection can be computed using FET. If both subsets are drawn randomly and independently, the size of their intersection is hypergeometrically distributed.

In this thesis, FET is only used for the case described above. It can be generalized to bigger contingency tables with variables having more than two possible values.

Mann-Whitney-Wilcoxon test

The Mann-Whitney-Wilcoxon test (MWWT) can be used to assess the enrichment of a subset of objects with respect to a numerical feature. Such a feature is used to sort all objects and compute a statistic based on the ranks of the subset of interest. We will state the formal description of the test similarly to Rohatgi and Saleh (2001). Consider $m + n$ observations $X_1, X_2, \ldots, X_m$ and $Y_1, Y_2, \ldots, Y_n$ from two continuous distribution functions, f_X and f_Y, respectively. The Wilcoxon statistic is given by

$$W = \sum_{i=1}^{n} Q_i, \tag{1.4}$$

where Q_i denotes the rank (in ascending order) of observation Y_i among all $m+n$ observations. The null hypothesis is that $f_X = f_Y$, the alternative could be one- or two-sided. If for instance the alternative is that $f_X \geq f_Y$ then large values of W support the alternative. The corresponding p-value is defined as

$$p(w) = P(W \geq w), \tag{1.5}$$

where the right-hand side denotes the probability under the null model. In order to compute that probability for small values of m and n a difference equation can be used. For large values, the following normal approximation is available:

$$\frac{(W - \frac{n(n+1)}{2})/(mn) - \frac{1}{2}}{\sqrt{(m+n+1)/(12mn)}} \xrightarrow{d} \mathcal{N}(0,1) \tag{1.6}$$

This approximation works well for $m, n > 8$. If both $m, n \leq 8$ we can easily compute the exact probability by enumerating all possible permutations or using a difference equation. If only n is small and m is large (or vice versa) the computation is more difficult. So-called network algorithms can still do the exact computation in reasonable time while brute force algorithms are too slow.

Student's t-test

The t-test is one of the most commonly used tests in statistics and exists in a number of variations. Given a sample from a normal distribution, it tests if the expectation value of the underlying normal distribution is equal to (or less or greater than) a given value (or the mean of the underlying normal distribution of another sample). For instance, the t-test can be used to test two samples against the null hypothesis that the underlying normal distributions have the same expectation value.

Multiple testing

In many applications, we will use one of the described tests many times in order to identify certain objects that are significant with respect to the test out of many such objects. Let us assume that n tests are executed independently, and that the null hypothesis is true for all tests. Then the probability to get some p-value $p < \hat{p}$ in at least one of the tests is $1 - (1 - \hat{p})^n$, as p-values are uniformly distributed under the null hypothesis. For instance with $\hat{p} = 0.05$ and $n = 100$ this probability is 0.9941, i.e. even if the null hypothesis holds, it is almost certain to find a 'significant' p-values (less than 0.05) during 100 tests. Obviously, the p-value is not a very objective measure, if the number of independent tests is not taken into account. Several procedures have been developed to compute p-values or other statistics that are independent of the number of tests performed. One possibility is to use the formula given above, which computes a corrected p-value $\bar{p}$ as

$$\bar{p}(p, n) = 1 - (1 - p)^n. \tag{1.7}$$

An approximation for that formula is the well-known Bonferroni correction

$$\bar{p}(p, n) = np. \tag{1.8}$$

This approximation is reasonably good for very small p. For instance, if $p = 0.001$ and $n = 50$, the exact formula computes $\bar{p} = 0.0488$, whereas the Bonferroni procedure results in $\bar{p} = 0.05$. But the Bonferroni correction is more than an approximation to equation 1.7. It results in a valid p-value even if there are dependencies between the tests. To prove that, let us denote with E_i the event that test i passes given the null hypothesis. Now, the probability that any of the n tests passes is $P(\cup_{i=1}^{n} E_i)$. Following Bonferroni's inequalities, we have $P(\cup_{i=1}^{n} E_i) \leq \sum_{i=1}^{n} P(E_i)$. Therefore, if we choose our tests (or the thresholds on the tests) such that $P(E_i) = p = \frac{\bar{p}}{n}$, the probability to get at least one passing result is $P(\cup_{i=1}^{n} E_i) \leq \sum_{i=1}^{n} P(E_i) \leq np = \bar{p}$. Bonferroni's method is usually considered too conservative. Less stringent methods were developed subsequently for instance in Holm (1979); Hochberg (1988) and Hommel (1988).

Computation of the hypergeometric distribution

In order to compute p-values for FET, the hypergeometric distribution has to be computed. If the calculations are performed naïvely, e.g. as provided by the COLT[3] library that we use for other numerical computations, the occurring binomial coefficients can become very large, leading to numerical problems. Starting from the hypergeometric distribution

$$P(S=s) = \frac{\binom{N-k}{n-s}\binom{k}{s}}{\binom{N}{n}}, \tag{1.9}$$

we get the following equations:

$$P(S=s) = \frac{(N-k)!k!n!(N-n)!}{(N-k-n+s)!(n-s)!s!(k-s)!N!} \tag{1.10}$$

$$= \frac{(N-k)!k!n!(N-n)!}{(N-k-n+s)!(n-s)!s!(k-s)!N!} \tag{1.11}$$

$$= \prod_{j=0}^{s-1} \frac{(n-j)(k-j)}{j} \frac{\prod_{j=0}^{k-s-1} N-n-j}{\prod_{j=0}^{k-1} N-j} \tag{1.12}$$

$$= \prod_{j=0}^{s-1} \frac{(n-j)(k-j)}{j(N-j)} \prod_{j=0}^{k-s-1} \frac{N-n-j}{N-s-j} \tag{1.13}$$

This can be easily translated into two loops that calculate results with good precision even for quite large parameter values.

1.4.2 Graphs and Petri nets

As most parts of this work deal with biological networks, a computational representation is needed. **Graphs** are a well-studied data structure in computer science and are well suited to represent biological networks. A graph consists of vertices and edges; in many applications vertices represent some kind of objects, and edges describe relationships among those objects.

In our case, vertices will most of the time correspond to biological entities, mainly genes and proteins, while edges represent interactions between those entities. In some cases, a different representation is advantageous, where we have two different sets of vertices, one of which describes the biological entities and the other one their interactions. Edges in that kind of graph can only occur between the two types of vertices, defining which biological entities participate in which interaction. This can be useful for instance in metabolic networks where interactions correspond to chemical reactions with several participating metabolites and enzymes. A graph with two sets of vertices and edges only between those sets is called a bipartite graph.

The formal definition of a graph is as follows.

[3] http://dsd.lbl.gov/~hoschek/colt/index.html

Definition 1.4.2. A graph G is a tuple of vertices and edges, $G = (V, E)$. In a directed graph, the edges are ordered pairs of vertices $E \subset V \times V$; in an undirected graph, the edges are unordered pairs $E \subset \{\{u, v\} : u, v \in V\}$.

A **subgraph** G' of G is a graph $G' = (V', E')$ with $V' \subset V$ and $E' \subset E$. An induced subgraph is defined by selecting only a subset of vertices, while the edges are the same as in the original graph, but restricted to the selected set of edges: $V' \subset V$ and $\forall v, w \in V'$: $(v, w) \in E'$ if and only if $(v, w) \in E$.

In chapter 5, an algorithm will be introduced that is related to the **subgraph isomorphism (SI)** problem, which is defined as the following decision problem: Given two graphs G_1 and G_2, is there a (not necessarily induced) subgraph G_1' of G_1 that is isomorphic to G_2? A graph $G = (V_G, E_G)$ is isomorphic to another graph $H = (V_H, E_H)$ if and only if there is a one-to-one map f between V_G and V_H such that for all pairs of vertices $v, w \in V_G$ the following condition is fulfilled:

$$(v, w) \in E_G \Leftrightarrow (f(v), f(w)) \in E_H \tag{1.14}$$

Sometimes a variant of this problem is considered which is called the **induced subgraph isomorphism** problem. In this variant, the problem is to decide if G_2 is isomorphic to an induced subgraph of G_1.

A **Petri net** is a bipartite graph with one set of vertices called places and the other set of vertices called transitions. Petri nets are used to study the dynamics of complex systems such as metabolic networks using so-called tokens that are used to describe for instance the concentrations of metabolites. A function giving the number of tokens for each place is then called a marking of the Petri net. Although we do not investigate the dynamics of networks in this work, we will adopt the Petri net terminology of places and transitions, when we deal with bipartite graphs.

Chapter 2

Expression Data Analysis

The term gene expression data refers to the abundances of messenger RNA transcribed from different genes that can be found within a cell under a certain experimentally controlled condition. Methods for measuring these abundances include serial analysis of gene expression (SAGE) (Velculescu et al., 1995), cDNA library sequencing (Adams et al., 1991), cDNA subtraction (Bautz and Reilly, 1966; Muerhoff et al., 1997), quantitative real time polymerase chain reaction (RT-PCR) (Bustin, 2000), and northern blotting (Parker and Barnes, 1999). Today, gene expression data are usually collected using DNA microarrays which are capable of measuring tens of thousands of mRNAs simultaneously. As microarray measurements are often impaired by strong noise, classical small-scale methods are still important for the validation of results from microarray analysis or for a more exact quantification of expression levels of single genes. Of high interest for the application of microarrays are also statistical and computational methods for the analysis of the raw microarray data. This chapter first reviews the microarray technology and then the problems arising from the data analysis task together with some algorithms that have been developed for these problems.

2.1 Areas of application

Gene expression measurement using DNA microarrays has many applications in biology and medicine. An overview of applications and technology can be found in Stoughton (2005). Emphasis on medical application or drug discovery is put in Petricoin III et al. (2002) or Stoll et al. (2005), respectively.

The most common set-up for microarray experiments is the comparison of two biological conditions, for instance comparing diseased with healthy tissue. The goal is to find out more about the disease on a molecular level, or – more concretely – to find potential drug targets among genes that are up- or down-regulated in the disease sample. Even if the regulated genes are no suitable drug targets, they could still be useful as diagnostic markers.

Time series measurements are used to gain an understanding of the dynamics of cel-

lular processes. In an early study that uses a time series of microarray measurements Spellman et al. (1998) investigate changes of gene expression during the yeast cell cycle.

Although the dominant application of microarrays today is the measurement of mRNA concentration, microarrays are not limited to expression profiling. When the sequences deposited on the microarray (the probes) correspond to other genetic features, they can also be utilized for genotyping and help identifying novel genes, transcription factor binding sites, alternative splicing, and exon structure.

2.2 Microarray technology

A microarray is a small slide that is designed as a sensor for many nucleotide sequences. Each microarray contains many spots organized in a two-dimensional grid, where each position corresponds to one specific nucleotide sequence. There are several microarray technologies, the two prevalent types today are cDNA spotted arrays and oligonucleotide arrays. Reviews of microarray technologies can be found for instance in Venkatasubbarao (2004), Southern (2001), Schena (2000), Coe (2003), Duggan et al. (1999) or Hardiman (2002). All microarrays rely on the ability of complementary DNA strands to bind to each other, as cDNA from a labeled sample hybridizes to the complementary strand deposited on the array; the amount of the hybridized molecules is measured using a microarray scanner. The amount of hybridized cDNA is supposed to be roughly proportional to the amount of cDNA in the sample, as the DNA on the array is available in much greater abundance, so first-order kinetics can be assumed (Duggan et al., 1999). The measured signal should therefore be roughly proportional to the amount of cDNA in the sample as well.

Oligonucleotide and cDNA arrays differ mainly in the DNA that is deposited on the array and how it was attached. Spotted cDNA arrays are constructed with cDNA samples from clone libraries that are amplified using PCR and then spotted on the array using spotting robots most of which employ contact printing techniques. For the construction of oligonucleotide arrays, DNA oligonucleotides of length 25-75 base pairs are synthesized directly onto the array using photolithography. The latter approach results in spots with good hybridization targets, as the short chains are in a defined state and accessible to the probes at every nucleotide. Furthermore, oligonucleotide arrays show very little variation between the slides. DNA spotted on cDNA arrays, instead, is in an ill-defined state. Only part of it is single-stranded, it contains intra-strand cross-links and multiple contacts to the slide material (Duggan et al., 1999). On the other hand, spots on cDNA arrays are more specific due to the greater length of spotted sequences; cDNA arrays are more affordable and can be customized more easily.

Due to the greater spot variability, hybridizations from different slides of a cDNA microarray cannot be compared directly. Therefore, cDNA that was reverse transcribed from mRNA from two different biological sources and labeled with two different dyes, is hybridized on a single array. The array is then scanned by a laser scanner to measure dye fluorescence. The resulting intensities of the two different colors can be compared, as fluctuations in the spotting procedure apply to both colors in the same way.

For oligonucleotide arrays on the other hand, cDNA from only one source is hybridized against the oligonucleotides on the slide. The spot variability is low enough that data from two different slides can be compared directly. Variations occurring due to differences in sample preparation, amplification, labeling, or the external conditions during hybridization apply to both types of arrays and must be dealt by appropriate normalization methods.

The market for oligonucleotide microarrays is dominated by Affymetrix GeneChips. Affymetrix uses oligonucleotides of length 25, which are organized in probe cell pairs consisting of a Perfect Match (PM) and a Mismatch (MM) cell. The PM cell contains a specific sub-sequence of its target, in the MM cell one nucleotide of that sequence is exchanged. With the help of the MM cells, the amount of unspecific binding can be estimated. The current GeneChip for the human genome (U133 plus 2.0) contains more than 54,000 probe sets with each probe set containing 10-20 probe cell pairs.

2.3 Analysis methods

As microarray measurements have become a standard procedure in many areas of biomedical research, analysis methods for the ever-growing amounts of data become more and more important. Quackenbush (2002) provides a review of normalization methods while Slonim (2002) covers analysis techniques that succeed the normalization of the raw data. Both reviews are from a Nature Genetics supplement issue that contains a lot of interesting information on analysis of microarray data and applications. The problems under investigation in microarray data analysis include the following:

Image analysis: The data that are directly measured are images produced by scanners for the microarrays. These images can be quite noisy, the exact shape and intensity of the spots has to be determined and extracted from the background. Therefore, powerful image analysis software is required to extract good quality data from the images. Figure 2.1 shows background images extracted by an image analysis software. Some artifacts can easily be determined. Spots in the corresponding regions should be analyzed very carefully. Mostly, such spots are flagged by the image analysis software, and flagged values are ignored by later analysis programs.

Normalization of the raw data: The next step after the image analysis is the normalization of the data. The goal of normalization is to remove from the data systematic effects that are caused by the technical procedure rather than biological differences. In general, normalization consists of different steps: background correction, probe-level analysis (for oligonucleotide arrays), within-array normalization (for cDNA arrays), and between-array normalization. A review on microarray data normalization can be found in Quackenbush (2002).

One effect that is often observed with two-color cDNA arrays is a dependency of the ratio between the two channels from the average intensity. A second problem for normalization is the high variance that is often observed for ratios of spots in the low-intensity area. Figure 2.2 shows an M vs A plot of expression data from a

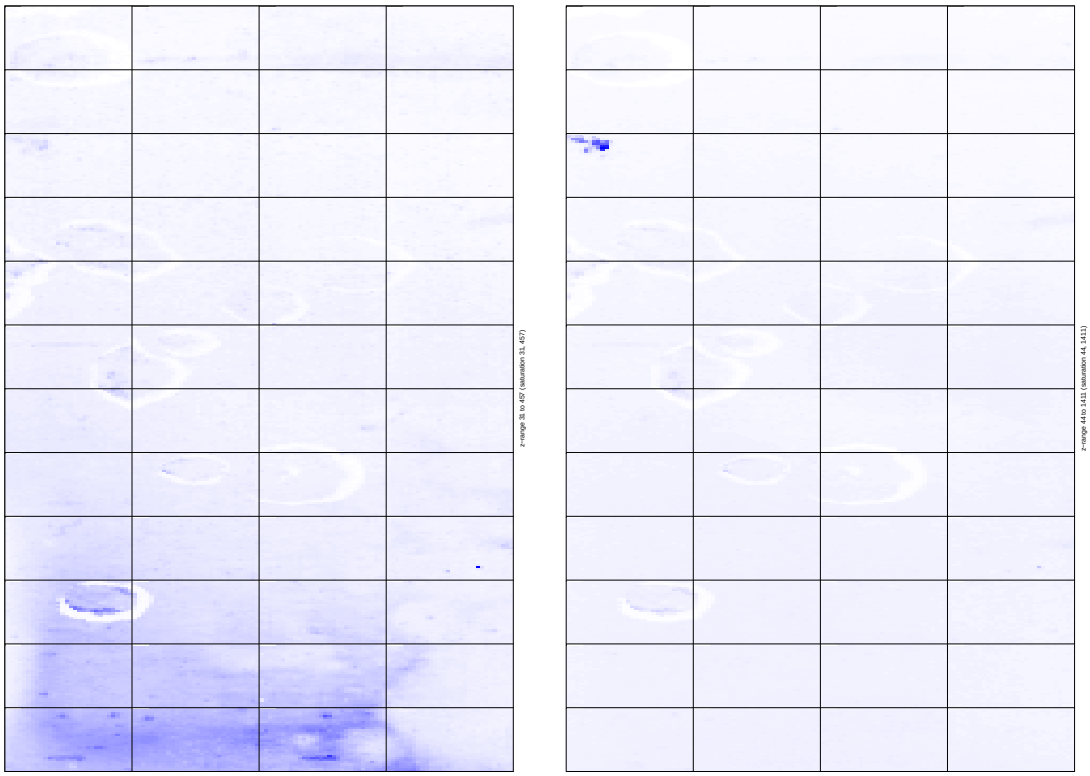

Figure 2.1: Image of the red (left) and green (right) background of a cDNA microarray as determined by an image analysis software. Some artifacts can easily be determined by visual inspection. Image analysis software and the experimenter have to make sure that such artifacts are corrected or at least marked in the data (flagged). Data are taken from a publicly available study on prostate cancer (Lapointe et al., 2004).

cancer study (Lapointe et al., 2004) before and after (within-array) normalization. M denotes the logarithm of the expression ratio of the two channels while A denotes the logarithm of the average intensity.

Computation of differentially expressed genes: For many research questions, it is of interest to find differentially expressed genes between two states. For instance, in pharmaceutical research expression levels for diseased and normal tissue samples are measured. The first apparent question is, which genes show different expression behavior between the sample groups. This problem boils down to distinguish random noise in the measurements from real biological differences. Furthermore, when the data are not or inappropriately normalized, systematic differences between arrays can arise. Thus, the computation of differentially expressed genes depends on the normalization step.

Clustering of genes or samples: Since the famous work by Eisen et al. (1998), clustering of genes or samples is another standard procedure for the analysis of microarray data. Generally, clustering of objects can provide a visualization of distances in high-dimensional spaces, e.g. as a dendrogram, and exhibit subgroups (clusters) where objects within subgroups are more similar to each other than objects from different subgroups. Clustering of genes organizes genes into groups with similar expression profiles. Often, these genes are functionally related (Eisen et al., 1998). Furthermore, clusters with profiles showing changes in regulation contain genes that are important in the experimental conditions under investigation. Clustering of samples can for instance exhibit disease subgroups with clinical relevance.

Sample classification: Sample classification provides a method for diagnosis of diseases. Often, samples are classified using differentially expressed signature genes, therefore providing hints to the biological processes connected to the classification. Popular techniques for classification include support vector machines and decision trees. Sample classification for disease diagnosis is of particular interest when it can be performed in early disease stages and when samples can be obtained by non-invasive methods such as blood samples.

Enrichment analysis: Enrichment analysis provides a simple and powerful method to exploit context information in the form of predefined gene sets. Such a set could represent all proteins in a pathway or all genes that play a role in a certain biological process. The goal of the analysis is to identify gene sets that are enriched in genes that show an interesting expression pattern, for instance genes that are differentially expressed. This is usually accomplished by statistical tests, most often by Fisher's exact test or a χ^2-test.

For all of these problems a wealth of methods has been proposed in journals or at conferences of the computational biology and statistics community. In the following, some of these methods, which are particularly popular or of importance for later chapters, will be presented.

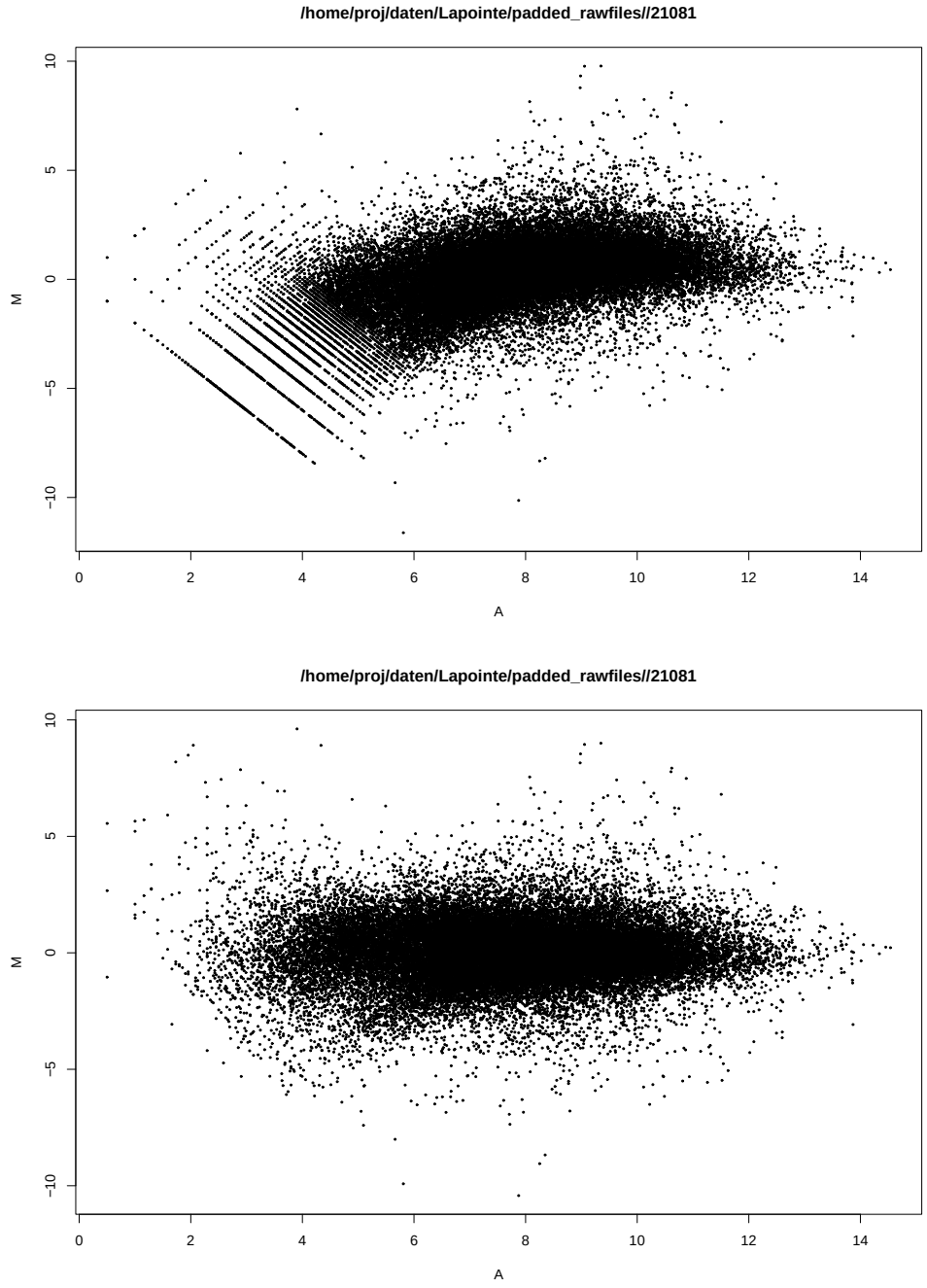

Figure 2.2: MA-plot of expression data from the same study as the data in Figure 2.1 before (left) and after (right) normalization. In the unnormalized data, spots with low intensities tend to have a lower ratio as well. In the normalized data, this bias has been removed.

2.3.1 Image analysis

The two most popular image analysis programs for microarray data, apart from the Affymetrix GeneChip software, are probably Spot, distributed by CSIRO Mathematical and Information Sciences, Australia, and GenePix, by Axon Instruments, which also sells microarray scanners and other related hard- and software. GenePix includes not only the image analysis but also software for the control of the scanning process and has more or less become an industry standard. While GenePix is a stand-alone software that runs only on Windows platforms, Spot is a package for the R statistics environment (R Development Core Team, 2004). This allows an easy integration of results from the Spot software into other statistical methods implemented in R. There are many more software packages for the analysis of microarray images. A short comparison of such packages can by found on Y. F. Leung's functional genomics website[1].

The main tasks for the image analysis software is to correctly estimate the shape and intensity of the spots as well as the background intensity. A review on image analysis methods can be found in Yang et al. (2001).

2.3.2 Normalization

Background correction

The image analysis software usually returns foreground- and background values for each spot. It is assumed that the foreground consists of two components, the signal from the bound cDNA molecules and noise from natural fluorescence, unspecific binding and the imaging process. The noise can be estimated from the background where no DNA was spotted. The simplest and most commonly used method for background correction is subtraction of the background intensities from the foreground intensities. Unfortunately, this leads to an increase of variance (fore- and background measurements both have variance, the sum has a greater variance) and many negative intensity values that can not be handled by some subsequent methods. Therefore, other methods try to minimize the number of negative values and use more sophisticated methods for background correction. Kooperberg et al. (2002) for instance use a Bayesian method, while Edwards (2003) use background subtraction if the difference between foreground and background intensities is large and uses a smooth monotonic and positive function when the difference is small or negative.

Alternatively, the background correction step can simply be skipped which can of course lead to biased ratios in cDNA arrays if the background differs in the two channels.

Lo(w)ess normalization

The *lowess* or *loess* method can be employed to remove intensity-dependent biases from two-channel microarray data. The name is derived from the term "**lo**cally **we**ighted

[1] http://ihome.cuhk.edu.hk/~b400559/arraysoft_image.html

scatterplot **s**moothing", as the method uses locally weighted regression to smooth data. A short introduction to smoothing that covers some additional approaches besides local regression can be found in Gentle et al. (2004). The *lowess* method was proposed by Cleveland (1979) and can be used for many different kinds of data that come in the form of one or more predictor variables and a response variable. It is based on the following principles:

- For each data point (the focal point), a polynomial fit on the local neighborhood is computed.
- The neighboring points are weighted according to their distance from the focal point for the fit.
- The value of the fitted polynomial is computed for the focal point as the "smoothed" value.
- Points with high residuals are iteratively down-weighted, as outliers should not affect the fit to a large degree.

The points used for the local fit can be specified by an absolute distance to the focal point or by the fraction of points that should be used. In the first case, the number of points can vary with the focal point, in the latter case, the specified number of nearest neighbors is used.

The points are weighted with a tri-cubic function:

$$w_i = (1 - (d_i/\text{maxdist})^3)^3,$$

where d_i denotes the distance of the i-th point from the focal point and maxdist is the maximal allowed distance. Next, the polynomial is fitted using a least-squares approach. With that approach, outliers can have a large influence on the fit. In order to achieve a more robust fit, a re-weighting procedure was suggested, down-weighting points with large residuals. With these additional weights, the fit is re-computed and the procedure is iterated a specified number of times. The effect of down-weighting and choice of polynomial degree is demonstrated in Figure 2.3.

The use of *lowess* for the normalization of microarray data was proposed by Yang et al. (2002).

Oligonucleotide array normalization

For the normalization of oligonucleotide arrays like the Affymetrix GeneChips, it is necessary to estimate the expression level for the probe sets from the measurements of the probe cell pairs. Rajagopalan (2003) compares three different statistical methods to estimate these levels and the expression ratios between different experiments. Among those methods is the MAS5 algorithm that is implemented in the Affymetrix software. The most basic feature of these algorithms is the present-call for each probe set giving an estimate

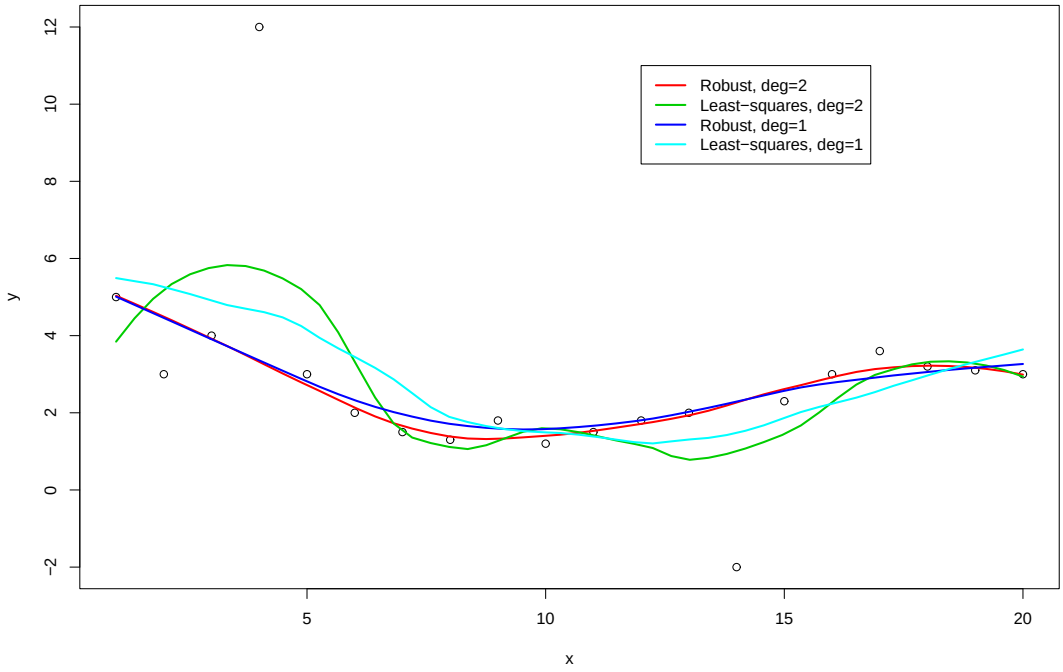

Figure 2.3: Robust and least squares lowess fit with polynomial degrees 1 and 2. Obviously the least squares fit without iterative down-weighting is much more vulnerable with respect to outliers.

whether the corresponding RNA was present in the sample or not. In MAS5, this present-call is based on a p-value computed by a Wilcoxon rank test that is based on the difference of the PM and MM cells.

Between-array normalization

Variations between arrays are usually not only due to the different conditions under study, but also due to differences in sample preparation, hybridization, and chip manufacturing. Between-array normalization is necessary in order to eliminate these effects as far as possible. In principle, the same methods can be used for oligonucleotide and cDNA arrays, but usually between-array normalization for cDNA arrays tries to retain the distribution of expression values for each array as far as possible and adjust only a few values like the median and the deviation using for instance a scale normalization (Yang et al., 2002). The reason for this difference is that many biases are already removed during within-array analysis.

For oligonucleotide arrays, it has to be decided, if the normalization is carried out on the cell intensities or on the summarized probe set values. The simplest between-array normalization is a global scaling to a common target mean value. This global normalization on the probe set values is the standard procedure in the Affymetrix software.

Bolstad et al. (2003) compare different between-array normalization methods for oligonucleotide arrays. They suggest several new methods, some of which are based on an MA-plot for pairs of arrays, i.e. treating two different arrays like the two channels of a cDNA array. One of these methods is the cyclic lowess normalization, which computes a lowess fit for all pairwise comparisons, adjusts the expression values according to a combination of the resulting lowess curves and iteratively repeats the process until the changes are very small.

In general, between-array normalization methods adjust the data such that some statistics become similar or equal for all arrays. As shown by Fundel et al. (2005), the effects of these normalizations can be drastic and should be checked manually or using automatic methods to check the stability of the final results for example by sub-sampling from the available replications.

2.3.3 Differentially expressed genes

One of the most common tasks in the analysis of microarray data is the identification of differentially expressed genes between different biological conditions, each of which is represented by one or more samples. In the first studies using microarrays this was usually done by computing a fold change, i.e. the ratio of expression levels in two samples, from a single two-channel microarray, as a replication of the microarray experiments was too costly (DeRisi et al., 1997; Chu et al., 1998). Genes with a fold change above a certain threshold were then considered differentially expressed. Today, replicate measurements are in most cases available, and differentially expressed genes are normally computed using statistical tests like the t-test or the Wilcoxon rank test. The most popular test is probably the t-test. More recently, special tests for microarray data have been developed. For instance,

significance analysis of microarrays (SAM) is a method based on a modified t-statistic and computes p-values using a re-sampling algorithm (Tusher et al., 2001). Classical linear models combined with empirical Bayes analysis represent a method that can be used to compute differentially expressed genes in a wide range of experimental settings (Smyth, 2004).

Statistical tests

In order to determine differentially expressed genes with statistical tests, one computes a gene-wise test statistic and considers genes differentially expressed if the value of the test statistic exceeds a threshold value. The test statistics that are used take into account the variation of the expression values over some control experiments or replicates. Usually, a p-value can be computed for the test statistic, giving the probability of observing an equally or more extreme value of the test statistic, given that the gene is not differentially expressed.

In most cases, the experimental set-up aims at a comparison of two biological conditions represented by two groups of replicated measurements. The question of interest is if a gene is differentially expressed between the two conditions. The multivariate case with more than two conditions is usually addressed with methods based on analysis of variance (ANOVA). Here, only the case of two conditions is covered.

The earliest and probably still the most popular test is the t-test or one of its variations, as described in 1.4.1, testing if the means of the two sample groups are different. The problem with this test is that it assumes a normal distribution of the expression values of each sample group, which often seems inappropriate or at least hard to justify. There are several approaches to remedy that problem. One is to compute p-values by a re-sampling strategy like it is done in the SAM, another one is to use distribution-free tests like the Mann-Whitney-Wilcoxon test described in 1.4.1. This test results in a significance value for the difference of the medians of the two sample groups. As it uses only the ranks of the expression data and not the values themselves it is independent of a concrete distribution. On the other hand, using only ranks can also be a drawback. For instance, if the expression values of some disease samples are for one gene just a little higher than those of the normal samples, the resulting p-value will be the same as for another gene with a much larger difference between the groups. Thus, the p-value only reflects the consistency of the difference between the sample groups but not the extend of the difference.

Significance analysis of microarrays (SAM)

Significance analysis of microarrays, introduced by Tusher et al. (2001), is a popular method to detect differentially expressed genes. The method is based on a new statistic, the relative difference, which is related to the t-statistics. The relative difference basically denotes the difference of the expression means for the two groups divided by the empirical deviation within the groups. The difference to the usual t-statistic is that a small constant term is added to the denominator (the variance). Therefore, genes with small fold change

and small variance do not appear as significant as with a t-test. Furthermore, the null distribution of the statistic is empirically determined. An expected value for this statistic is computed by gene-wise averaging over the relative differences calculated with permuted group labels. A threshold on the difference between the relative difference and its expected value is then introduced to identify differentially expressed. The same thresholds are then applied to the permuted data sets in order to estimate the number of false positive calls that can be expected. The method appears to be quite robust due to the heavy use of re-sampled data sets. Furthermore, there are nice ways to visualize the results that provide some possibility for plausibility checks.

2.3.4 Clustering and visualization

Since the paper of Eisen et al. (1998), clustering has become a standard analysis method for expression data analysis. Often, clustering is done only to provide a visualization of the measured data. A large matrix of expression data can be displayed as a so-called heat map, where the genes, and sometimes the samples as well, have been ordered according to a dendrogram that results from an agglomerative clustering. This heat map can sometimes exhibit interesting features that would not have been apparent if the matrix had not been re-ordered.

Especially in the case of time series experiments it can also be interesting to display the prototypic time course for each cluster.

Principle component analysis

Principal component analysis (PCA) is a dimension reduction technique. Given a set of points from a high-dimensional space, PCA projects these points onto a lower-dimensional subspace such that a maximal amount of variance is retained. If the main source of variance is the signal of interest, this approach can help reducing noise. Furthermore, when the dimension of the subspace is chosen to be three or less, visualization of the data is possible.

PCA is also mentioned in this section, as it is used for visualization purposes and one of the most important applications is to visually identify outliers in the experiments. Sometimes, also clusters are identified by simple visual inspection of a PCA plot.

Agglomerative clustering

In agglomerative clustering, a dendrogram is built in a bottom-up fashion. Given a set of objects and a distance matrix, an agglomerative clustering algorithms starts by joining the two closest objects. Then, in each step, the two closest clusters are joined to form a new cluster. Agglomerative clustering methods mostly differ in their definition of distance between clusters. In single linkage clustering, the distance of two clusters is the minimum distance of two objects within the clusters, in average linking, it is the average distance of object pairs, and in complete linkage it is the maximum distance. These definitions all rely

on the distance matrix between objects, which for gene expression data is usually based on Pearson or rank correlation, or euclidean distance.

K-means clustering

The k-means algorithm iteratively calculates cluster means, given the cluster assignments of all objects, and then re-assigns the objects to the cluster with the closest mean. While k-means usually gives nice and stable clusters, the problem is that the number of clusters must be determined in advance. Alternatively, the clustering algorithm can be executed for different values of k, and the best value is chosen afterward using some quality measure for the resulting clustering.

Self-organizing maps

Self-organizing maps (SOMs) have been suggested very early for the analysis of gene expression data (Tamayo et al., 1999; Törönen et al., 1999) and still constitute a popular tool. Originally proposed by Kohonen (1988), they can identify clusters of genes with similar expression profiles when applied to gene expression data. SOMs impose a topology on the identified clusters, which is probably the method's biggest strength. Neighboring clusters in that topology are similar with respect to their expression profile.

Bi-clustering

Bi-clustering looks for subsets of genes and conditions in a matrix of expression data, such that the expression data in that sub-matrix are coherent according to some measure. The reasoning behind this approach is that genes might be co-regulated under some conditions while under other conditions they might be regulated independently. A nice review of bi-clustering algorithms can be found in Madeira and Oliveira (2004).

2.3.5 Sample classification

Classification of tissue samples using gene expression data is considered a promising new method for diagnosis of complex diseases, especially cancer and cancer subtypes. A classifier can be trained from labeled data, i.e. using gene expression data from samples with known diseases; afterward, the classifier can be applied to new data, e.g. for diagnosis.

Three classifiers that have been applied to microarray data will be described here.

Decision trees

One important aspect of classifiers is their interpretability. While the main goal of a classifier is the correct labeling of samples, it is also important to know on which basis the label is selected. If the classification is based on a single gene or a small number of genes, this has an immediate biological interpretation. Decision trees have this property of interpretability which is one of their major strengths.

Decision trees classify data according to rules that are attached to the nodes of the tree. As they only depend on one feature at a time decision trees become interpretable, a desirable property that sometimes lets them appear advantageous even when other methods yield superior classification performance (Krishnan and Westhead, 2003). Decision trees represent a supervised classifier, i.e. labeled training data are necessary. In order to learn decision tree rules, the training data is repeatedly split into subsets in a way that optimally separates the known classes.

Support vector machines

Support vector machines are among the most popular machine learning tools today. In their simplest version, support vector machines learn the separation of two classes from labeled training data. Support vector machines represent linear classifiers that maximize the margin of the hyperplane that separates the two classes. Such classifiers were suggested by Boser et al. (1992). The theory of support vector machines is laid out in detail in Vapnik (1998).

Support vector machines have been used for the analysis of expression data early on (Brown et al., 2000; Furey et al., 2000), for the classification of genes as well as for samples.

StAM

Structured analysis of microarrays (StAM) was proposed by Lottaz and Spang (2005) and aims at providing focused classifiers that are based on relevant biological processes only. The idea is that there may be disease subgroups that can be classified according to different gene sets. Biologically, this corresponds to different molecular mechanisms that appear in disease subtypes. Therefore, classifiers should be derived that classify accurately at least a subgroup of the disease samples and are based on a focused set of genes belonging to one biological process.

The method uses GO annotations in order to define the biological processes each gene participates in. Classifiers are built for all leaf nodes of the GO hierarchy, using only genes annotated with the respective GO term. Next, these classifiers are propagated to higher nodes in the GO hierarchy using some combination rules. A scoring method is applied to highlight GO terms with associated classifiers that can identify at least a sample subgroup with high accuracy. An overall classifier is available at the root node.

2.3.6 Enrichment analysis

Given a predefined list of gene sets and expression data, the goal of enrichment analyses is to identify sets that contain many genes that show an interesting expression pattern. This approach can be used to identify pathways or functional classes of genes that are significantly regulated. If, for instance, a library of metabolic pathways is used to define the gene sets, it is possible to find pathways that are up- or down-regulated in the experimental conditions under study.

Very commonly, Fisher's exact test is used to calculate p-values for the over-representation of significantly regulated genes in the predefined gene sets. First of all, significantly regulated genes have to be detected, usually by defining a threshold on a previously computed p-value for differential expression. Then the resulting set of regulated genes is compared with each of the predefined gene sets. As described in 1.4.1, Fisher's exact test computes the significance of the overlap of the two sets. For large sample sizes, a χ^2-test is often used. Of course, it is possible to look at other features than differential regulation, and other tests can be used as well, for instance the Wilcoxon-Mann-Whitney rank test or the t-test.

A review on tools for enrichment analysis, focusing on Gene Ontology, can be found in Khatri and Draghici (2005). The authors list several tools and describe the statistical models used. Unfortunately, they describe Fisher's exact test and the hypergeometric model as different methods, which is a common misconception in many publications in bioinformatics journals. In an earlier work, Draghici et al. (2003) describe the methods in detail, giving mathematical formulas. Here, a little calculation shows that the models are indeed mathematically equal.

Recently, a new method for the identification of significant gene sets in expression profiling studies has been proposed by Tian et al. (2005). Their approach is based on a rank test, but they take the correlation structure of genes into account by determining the null distribution through a sampling approach. Furthermore, they compute q-values to correct for multiple testing.

Chapter 3

Unsupervised Decision Trees

3.1 Introduction

Clustering of microarray gene expression data is performed routinely, for genes as well as for samples. Clustering of genes can exhibit functional relationships between genes; clustering of samples on the other hand is important for finding e.g. disease subtypes, relevant patient groups or related treatments. Unfortunately, most sample-wise clustering methods do not facilitate the biological interpretation of the results. In this chapter a novel approach for clustering samples from gene expression data sets is proposed that makes use of available functional annotations for genes. The method computes dendrograms with Gene Ontology terms annotated to each splitting node. These dendrograms resemble decision trees with simple rules at each node which can help to find biologically meaningful differences between the sample groups. We have applied our method to a public gene expression data set from a study of prostate cancer. The original clustering which contains clinically relevant features is well reproduced, but in addition the decision tree rules give hints for a biological explanation of the clusters. In particular, the root node, which separates tumor from non-tumor samples are based on Gene Ontology terms such as *reproduction, monosaccharide metabolism, lipid biosynthesis* and *transmission of nerve impulse*. The described method can be useful for discovering gene classes that are particularly good at dividing a set of microarray samples into a hierarchical structure. These gene classes can give valuable indications about biological processes that diverge between sample groups.

3.2 Background

Clustering of genes or samples is a standard procedure performed for the analysis of gene expression data. While gene-wise clustering can bring functionally related genes together, clustering of samples can provide insight into disease subtypes or patient groups. This is of particular importance when subgroups can be linked to clinically relevant features such as recurrence or severity of disease.

Many different clustering algorithms have been applied to microarray data, including

agglomerative hierarchical clustering with distances based on Pearson correlation (Eisen et al., 1998) as well as more sophisticated methods such as self-organizing maps (Tamayo et al., 1999). These methods are usually applied on all measured genes or those satisfying some filtering criteria in order to obtain a grouping of the samples, often in the shape of a dendrogram. This study only covers clustering of samples, but as the formal question of grouping similar objects remains the same, most clustering methods can be applied on samples and genes alike.

In general, the resulting groups or dendrograms do not provide any hints about biological processes in which the samples were particularly different between clusters, an obvious question to ask from a biological point of view. This issue is commonly addressed by identifying significantly differentially expressed (DE) genes using statistical tests (Tusher et al., 2001; Gentleman et al., 2005; Lönnstedt and Speed, 2002) and then detect over/under-representations of positive genes within pre-defined functional classes (Palenchar et al., 2004; Al-Shahrour et al., 2004). Other approaches include gene-list based methods such as iterative group analysis (iGA) (Breitling et al., 2004), LACK (Kim and Falkow, 2003) and various resampling based methods that find e.g. significantly high pairwise gene correlations, high learnability (Pavlidis et al., 2002), or conspicuousness (Zien et al., 2000).

An aspect shared by all the methods based on DE genes is that they are uni-variate, i.e. they cannot address dependencies between genes as they work one gene at a time. For each gene, the association of the expression measurement with the sample label is assessed, and afterwards the distribution of the outcome is investigated. Since genes work together and are highly dependent of each other, this approach does not reflect the complexity of real biological systems. The resampling based methods do better in this aspect but as they do not consider all genes simultaneously they cannot discover more general tendencies in the data.

Using a standard clustering approach and looking for DE genes associated with a second variable, e.g. disease severity, is thus based on two disparate concepts. The clustering usually works on all genes simultaneously, mixing information from a large and heterogenous set of biological processes; identification of DE genes on the other hand does not take dependencies between genes into account at all.

Our approach instead includes information on predefined gene classes that represent different biological processes, in this case a mapping of genes to Gene Ontology (GO) (The Gene Ontology Consortium, 2000), directly in the clustering procedure. For each such gene class, a single feature is computed using principal component analysis (PCA). Then, the features that support a clear grouping of the samples are selected. These should give an indication on the biological processes in which the sample groups differ the most. Thus, we identify differentially expressed gene classes instead of differentially expressed genes and simultaneously compute a clustering that makes use of these known gene classes.

Similar to our approach, Lottaz and Spang (2005) take advantage of functional gene classes, introducing a classification method that combines expression values of genes that are a priori known to be related. Our goal is instead to use such information in clustering methods.

Using models based on biological processes, we propose a new clustering method and

call it GO unsupervised decision trees (GO-UDTs), as it results in decision tree-like structures. On a publicly available expression dataset from a prostate cancer study, our method exhibits similar clusters as were shown in the original publication by Lapointe et al. (2004); in addition it provides valuable indications for a biological interpretation.

3.2.1 Unsupervised Decision Trees

Our goal is to remedy one major drawback of current clustering methods – their lack of interpretability. In order to reach that goal, we borrow from methods known from classification theory (Quinlan, 1993) and adapt decision trees to our clustering problem.

Conventional decision trees need to be trained on labeled data which is not available in an unsupervised setting like clustering. Having no labeled data, an algorithm that determines splits and corresponding rules in an unsupervised way is needed. Such algorithms have been developed previously in the form of UDTs and used for clustering (Karakos et al., 2005; Basak and Krishnapuram, 2005; Bellot and El-Bèze, 2000). UDTs are constructed like conventional decision trees by splitting the data into subsets, selecting a simple feature and a cut-off as a rule for each split. Instead of using labeled training data to determine a split, UDTs make use of an objective function that measures the quality of the resulting clustering. To our knowledge UDTs have not been used for the analysis of biological data before.

3.2.2 GO-UDTs

In order to make UDTs applicable to the clustering of samples from gene expression data, we have designed two new objective functions using features based on functional gene classes. The intuition behind these functions is to score the quality of a split using a measure of the separation of the resulting groups. Each feature used for splitting the data is based on genes from a single functional class. The GO-UDT algorithm computes a dendrogram in a top-down manner, in each step dividing the samples into subsets that exhibit large differences at least in some gene classes. Given a subset of the samples, the following steps are performed to determine the gene classes that imply the optimal split at a tree node.

1. For each gene class, compute a single feature by
   - selecting all genes belonging to that class,
   - summarizing the data matrix built up with the genes from that gene class using PCA and
   - selecting the first principal component (PC).
2. Cluster the samples according to each gene class using the computed feature.
3. Score gene classes according to an objective function which measures the quality of the separation of the resulting clusters.

4. Select a set of high scoring gene classes that imply a similar clustering.

5. Recursively re-partition the resulting sample subsets until some stopping condition is fulfilled.

At each inner node of the tree, the samples are split into two groups. Using only the first PC of a gene class, a global optimum for the clustering can be achieved using algorithms like K-means or partitioning around medoids (PAM) (Kaufman and Rouosseeuw, 1990).

Two different objective functions, described in Section 3.3.3, were used for evaluating the obtained clusters. The first is called the *model comparison* (MC) score and is based on the comparison of a uni-modal to a bimodal model. The second, *weighted silhouette* (WS), is based on a well known characteristic of clusterings; the Silhouette index (Rousseeuw, 1987). Using any of these, the best clustering is chosen and the node is annotated with the gene class that was used for the split.

This way, functional classes are identified which most strongly exhibit natural partitionings. Furthermore, the labels of these classes are expected to carry relevant information about the reason for that partitioning.

3.3 Methods

3.3.1 Data

The investigated expression data set contains measurements from 41 normal prostate specimens, 62 primary prostate tumors and nine lymph node metastases and has therefore a sufficiently large amount of samples. Importantly, the original publication showed that relevant sample groups could be discovered using hierarchical clustering.

The first identified subgroup consists of samples from healthy patients. Tumor subgroup I was identified as the clinically least aggressive whereas subgroups II and III contain samples from more ill-prognosed tumors. Furthermore, subgroup III was found to be similar to the lymph node metastases and contain most of eight of them. These are the subgroups we expected to rediscover with the addition of attaching a set of relevant labels of biological processes at each split in the data.

The used cDNA microarrays had ∼38,000 spots representing ∼24,000 unique UniGene cluster identifiers (genes) and were cross hybridized with a common reference of pooled mRNA from eleven established human cell-lines.

Normalization, pre-processing

The raw data was downloaded from SMD and processed with *R* (R Development Core Team, 2004) using the *LIMMA* library (Smyth, 2004) which is part of the *Bioconductor* project (Gentleman et al., 2004). Arrays were print-tip-loess normalized (Yang et al., 2002) and gene-wise median centered within each of the three different print runs to minimize the otherwise strong experimental bias. The arrays were finally normalized between each other

using the scaling technique also described by Yang et al. (2002). Missing values coerced from background correction were imputed using the K-nearest neighbor algorithm provided by the R library *impute* (Troyanskaya et al., 2001). Expression values of each gene were mean-centered and scaled to unit-variance.

Proceeding only with genes associated with a Locuslink identifier and median replacement of duplicate spots, 13,365 genes were left for further examination. Since no analysis was made using all genes simultaneously, no filters were applied as these will inevitably remove *some* information and the vulnerability to the noising effect of many genes with low intensity is controlled by *a priori* dimension reduction instead.

3.3.2 Gene class models

A well covering Locuslink to GO mapping was obtained from ErmineJ (Pavlidis, 2005). Using that mapping, all GO terms that are annotated to at least ten and at most three-hundred genes in our dataset were selected, as we hypothesize that bigger classes are likely to resemble a random sampling of all genes, and smaller classes are vulnerable to noise in the data. Genes with a GO term in common, either by direct annotation or by implication from the GO tree structure, comprise the gene class corresponding to that term.

For each of the classes a PCA was performed using the singular value decomposition based function *prcomp* in R and the first PC was selected as the feature representing that gene class. Using only this feature, the samples were divided into two groups and a quality score was attached to the split as described in the next section.

3.3.3 Scoring and clustering

Two different measures for assessing the separation of clusters were applied: the model comparison (MC) score and the weighted silhouette (WS) score. When computing the MC score, the clusters are assigned implicitly; for the Silhouette score, clusters are computed using PAM. In both cases, we allow only two clusters.

Let $X = \{x_1, \ldots, x_n\}$ be the set of all sample points, projected onto the first PC of a gene class. The two scores are then defined as follows:

1. **Model comparison:** The idea behind the model comparison scoring scheme is to measure the bimodality of the data. Two models are fitted: a single Gaussian denoted by p_1, and a mixture of two Gaussians denoted by $p_2 = qg_1 + (1-q)g_2$. For the fit of the mixture model, the Mclust algorithm from the *MCLUST* package (Fraley and Raftery, 2002) was deployed, which uses an EM algorithm to fit the different models. The score is then defined as the average log-likelihood ratio of the data:
$$S_{\mathrm{MC}} = \frac{1}{n}\sum_{i=1}^{n} \log \frac{p_2(x_i)}{p_1(x_i)}. \tag{3.1}$$

2. **Silhouette Index:** Let C_1, C_2 be a partition of X, as computed by PAM. Silhouette values are defined for every point $x \in X$ of a cluster as the difference of the average

distance between x and all points from the closest of all other clusters, and the average distance between x and all other points from the same cluster. In the two-cluster case, the Silhouette value for point x in Cluster 1 is defined as follows:

$$a(x) = \frac{1}{|C_1| - 1} \sum_{x' \in C_1 \setminus \{x\}} d(x, x') \tag{3.2}$$

$$b(x) = \frac{1}{|C_2|} \sum_{x' \in C_2} d(x, x') \tag{3.3}$$

$$s(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}. \tag{3.4}$$

The silhouette value for points in Cluster 2 is defined analogously.

In order to compute a single value for the clustering, the silhouette values of all points are averaged.

$$S_{\text{silhouette}} = \frac{1}{n} \sum_{x \in X} s(x) \tag{3.5}$$

This score is somewhat sensitive to outliers, as it is based on the relative distance between the observed clusters. Therefore, often clusterings with unbalanced cluster sizes are picked. This is avoided by weighting the score with the entropy of the clustering, as that gives a moderate bias towards balanced clusters. Thus the final weighted silhouette score is defined as

$$S_{\text{ws}} = -\left(\frac{|C_1|}{n} \log \frac{|C_1|}{n} + \frac{|C_2|}{n} \log \frac{|C_2|}{n} \right) S_{\text{silhouette}}. \tag{3.6}$$

3.3.4 Selecting a good split

Obviously, one gene class will have the best score and could be the one finally chosen to split the data, but since small changes in the clustering vector will have large impact further down the tree, a higher amount of stability on the chosen gene class was desired. Therefore a heuristic was applied, choosing a gene class, only if the implied split is supported by other high scoring classes as well.

More precisely, a split for gene class c_1 can only be made if there are at least A different gene classes among the T highest scoring classes that imply similar clusters to a minimum threshold of S, where all of these gene classes must satisfy a maximal dependence level $d(c_1, \cdot) < D$. The similarity of clusterings $C^{(1)}$ and $C^{(2)}$ and the dependency of gene classes c_1 and c_2 are defined as follows:

$$s(C^{(1)}, C^{(2)}) = \frac{1}{2} \left(\frac{|C_1^{(1)} \cap C_1^{(2)}|}{|C_1^{(1)} \cup C_1^{(2)}|} + \frac{|C_2^{(1)} \cap C_2^{(2)}|}{|C_2^{(2)} \cup C_2^{(2)}|} \right) \tag{3.7}$$

and

$$d(c_1, c_2) = \frac{|c_1 \cap c_2|}{|c_1|}. \tag{3.8}$$

Still, the top scoring gene class fulfilling these conditions is not necessarily chosen; instead, we require the A supporting gene classes to have minimum average rank.

For visualization, the supporting terms were also added to the node, as they could be biologically relevant as well. A maximum of ten terms were added to each node.

If no A supporting gene classes can be found, the expansion of the tree is discontinued. Since (3.1) is only relevant for fairly big cluster sizes, another stopping condition was added, stating that nodes with less than five members should not be created.

Both the generated trees in this study was computed setting $A = 2$, $D = 0.5$, $S = 0.75$ and $T = 30$. The graph drawing program *dot* from the *graphviz* package (Gansner et al., 2002) was used for the visualization.

3.4 Results

In the publication of Lapointe et al. (2004) three subgroups of cancer samples were identified which are linked to relevant clinical features. Although the discovered grouping is not necessarily the best possible, we will consider it as a basis for comparison with our results.

See Section 3.3.1 for a detailed description of the data, data pre-processing, and cancer subgroups.

GO-UDTs of the expression data were generated using both the proposed goodness measures; the result is shown in Figure 3.1 and 3.2.

Both generated trees reconstruct the subgroups identified in the original paper well, but on slightly different terms. The MC based score groups five tumors with the non-tumors which is three more than in the original publication. The WS score performs slightly worse, grouping eight tumors erroneously. The first node is very similar between the two measures; seven of the ten best gene classes are the same.

Both methods find *lipid metabolism* as particularly differential between tumor and non-tumor samples, this was also pointed out in the original publication by looking at key-genes such as acetyl-coenzyme A carboxylase α and α-methylacyl-CoA racemase. The same way, metabolic activity was found to be higher in subgroup III by looking at ATP5D, DCI and DECR2. The first node in our trees indicates that these are features that also discriminate well generally between tumor and non-tumor as it also contains terms such as *monosaccharide metabolism* and *one-carbon compound metabolism*. Notably, also *transmission of nerve impulse* is good for this separation supporting the well-known tight interaction between prostate tumors and nerve cells (Ayala et al., 2004).

After the first node, the two trees diverge from each other. The WS score isolates subgroup II mixing it with three members of other subgroups, whereas the MC score separates thirteen subgroup III members to a pure leaf with six of the nine lymph node metastases. The WS score later creates a nearly identical leaf sharing *actin filament based process* with the MC score as motivation for doing so. A member of this class is actinin-4

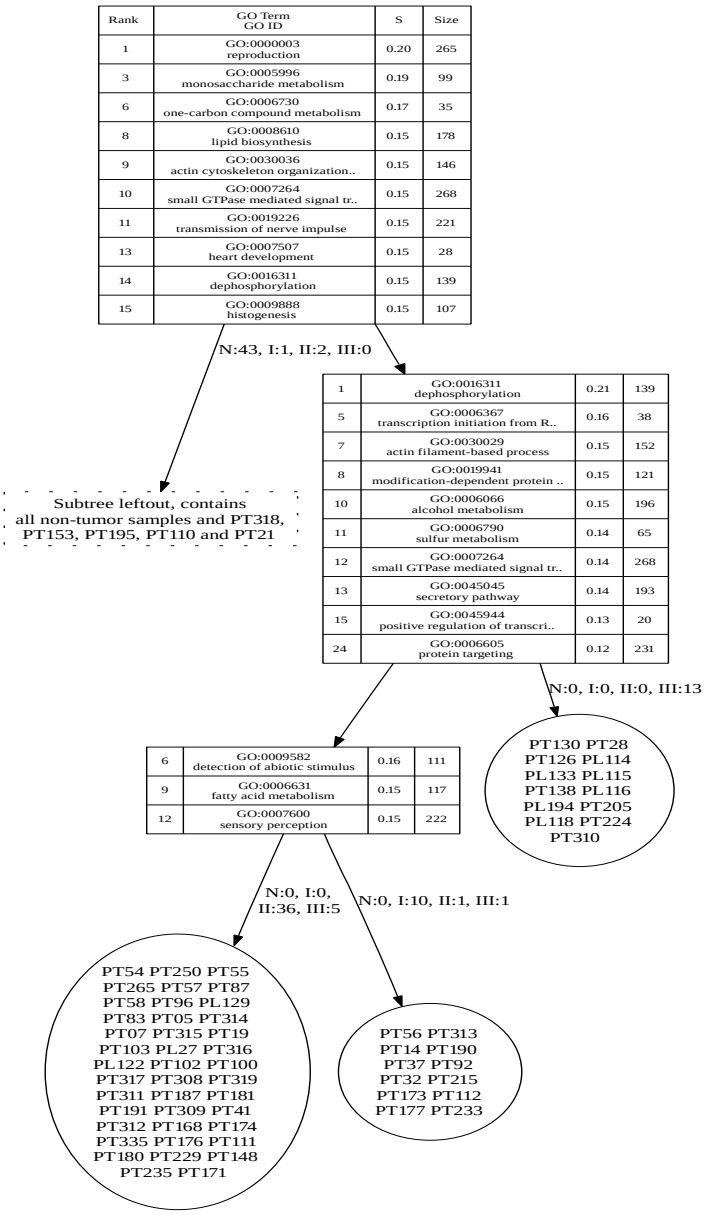

Figure 3.1: An unsupervised decision tree of the Lapointe prostate cancer dataset computed using the model comparison scoring method. GO terms are ranked (column *Rank*) according to the score of their induced split (column S) and then a set of GO terms with similar splits and minimum average rank is selected at each node. The column *Size* denotes the number of genes in each gene class. The root node indicates that genes associated with reproduction and metabolism are good at separating the non-tumor from tumor samples. *Dephosphorylation* separates thirteen of the nineteen samples from subgroup II from the others, and *detection of abiotic stimulus* and *fatty acid metabolism* gather nearly all samples from subgroup II to a mixed leaf slightly enriched with advanced grade tumors. PT153 and PT110 were grouped with the non-tumor samples in the original publication as well.

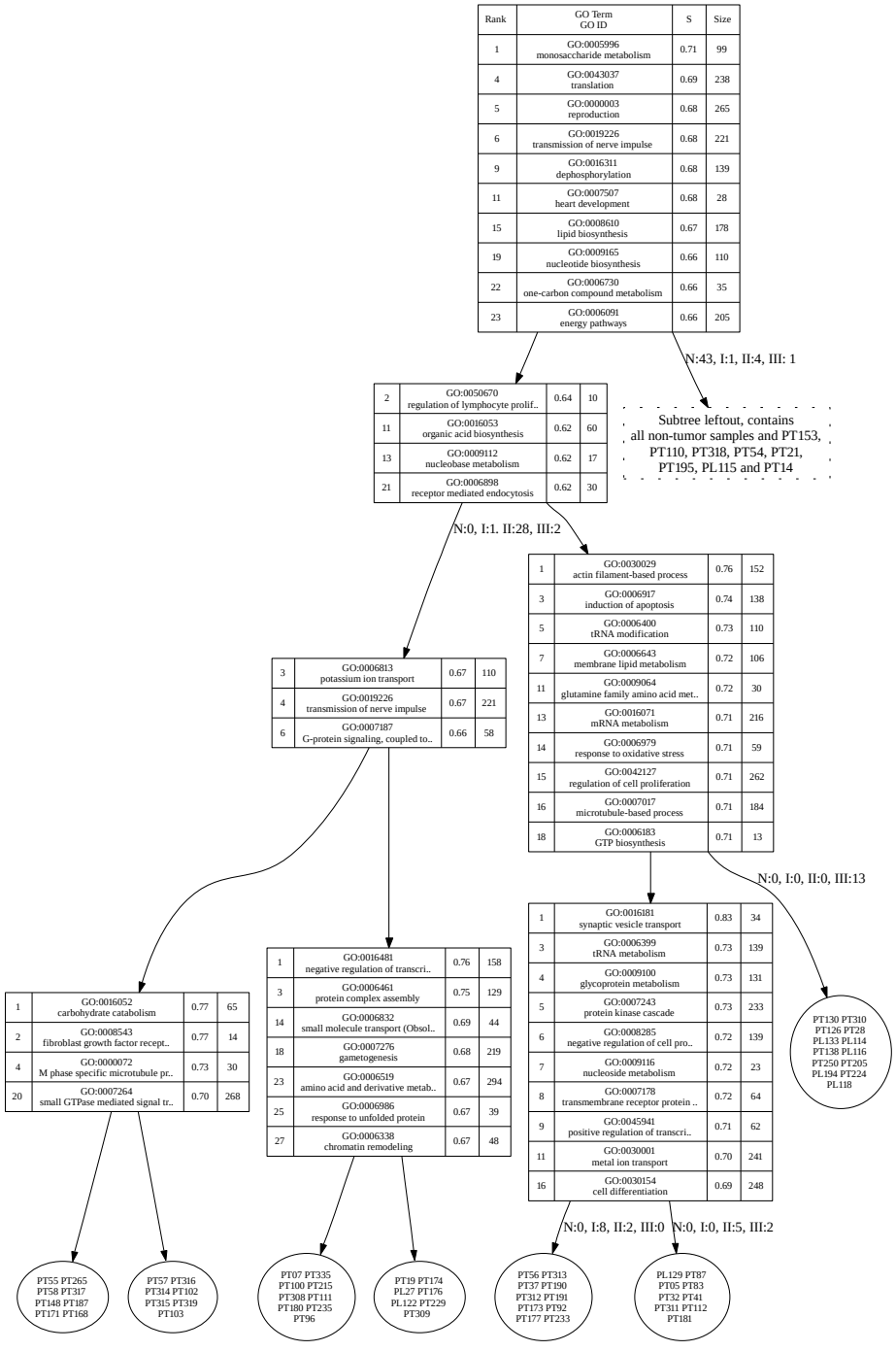

Figure 3.2: An unsupervised decision tree of the Lapointe prostate cancer dataset computed using the weighted Silhouette scoring method. The first node splits the data using *monosaccharide metabolism* resulting in eight tumors being grouped with non-tumor samples. *Regulation of lymphocyte proliferation* is good at enriching lymph node metastases, and *synaptic vesicle transport* distinguishes the less aggressive tumors from subgroup I from a mixture of samples from subgroups II and III.

which is a biomarker for cancer invasion and metastatic capability (Honda et al., 2005). Other terms found by the WS score is the highly relevant *induction of apoptosis* and *regulation of cell proliferation*. *tRNA modification* indicates differential protein synthesis in III versus I and II, something the original publication also pointed out but by instead looking at the key-genes RPL13, RPS15 and RPS9.

The MC score separates II from I well using *detection of abiotic stimulus* and the more relevant term *fatty acid metabolism* (Rossi et al., 2003). The resulting leaf containing subgroup II is also slightly enriched with advanced grade tumors ($p = 0.04$, Fisher's exact test).

The non-tumor samples were also separated into a series of subgroups. Even if these subgroups did correspond to some relevant variable, e.g. age, that subtree was left out as this is only speculative. Furthermore, GO-UDTs represent a kind of clustering technique and as such it can find groupings in the data regardless if they are meaningful or not.

Looking at the data displayed by the first two principal components using only genes from *reproduction*, two clear clusters are visible separating tumor from non-tumor samples. This motivates the simplified method of only considering the first PC, as the second provides little extra information for that split. Instead the second PC separates subgroup II from III but this direction changes expectedly to the first in the second node when all non-tumor samples have been removed, see Figures 3.3 and 3.4.

3.4.1 Over-representation of DE genes

As comparison to more well-known ways of testing functional classes for importance, we performed an over-representation analysis. First, DE genes between tumor and non-tumor/metastasis samples were identified using LIMMA to resemble the first split in the MC based tree. Dichotomizing the data on $q = 0.05$, with q-values defined as in Storey and Tibshirani (2003), the classes with the unlikeliest rate of DE genes were identified by computing Fisher's exact test p-values. The best class, *nerve ensheathment* had ten out of twelve genes identified as DE, $p = 0.0005$. *Transmission of nerve impulse* which was identified by both of the proposed scoring methods was also significant, $p = 0.01$ along with *actin filament-based process*, $p = 0.04$. None of the other mentioned terms had significant over-representation on $\alpha = 0.05$. Analogously, the most over-represented classes were not highly scored by our measures, the reason is clearly seen in Figure 3.5 showing the confounded score plot on *nerve ensheathment*. Examining one gene at a time can not reveal any synergistic effects; an argument that does not speak against over-representation analysis, it just indicates that other measures also should be considered.

3.5 Conclusions

Clustering via GO-UDTs is an interesting and novel approach for constricting clustering to be done on a one-feature-at-a-time basis in order to increase the interpretability of the final output. In this study, we investigated if GO-UDTs are useful for sample-wise clustering

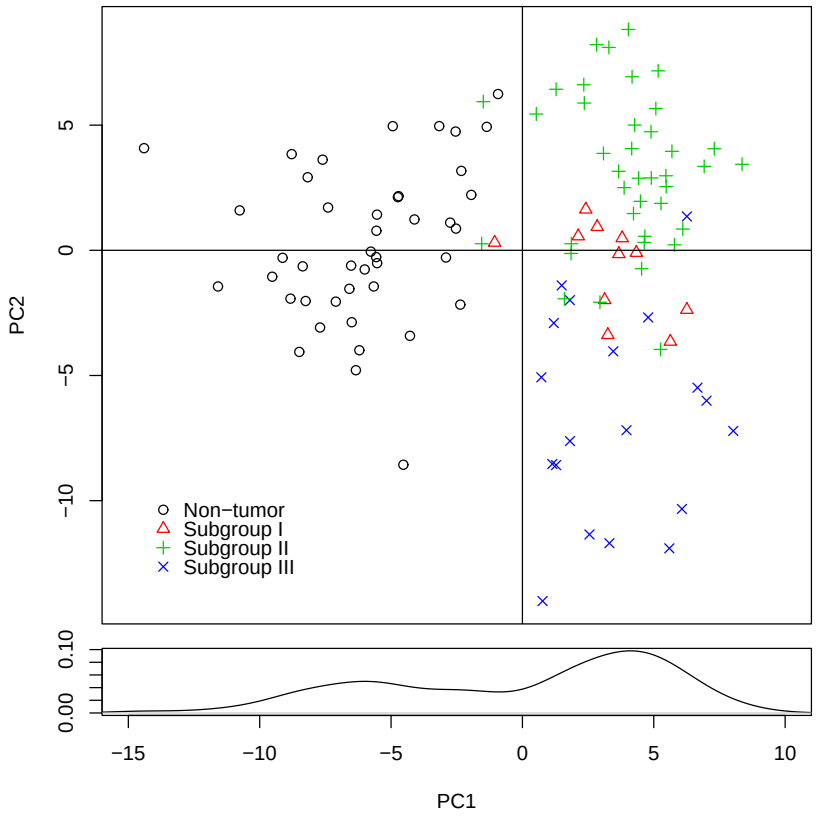

Figure 3.3: The score plot of the first two PCs (19% of the total variance) of the expression data matrix built up using genes from the class *reproduction*. This was the highest scoring gene class according to the model compairson measure. The first component allows separation between tumor and non-tumor samples, the second finds the subgroups within the tumor samples. The density plot at the bottom indicates that the data shows strong bimodality on the first PC.

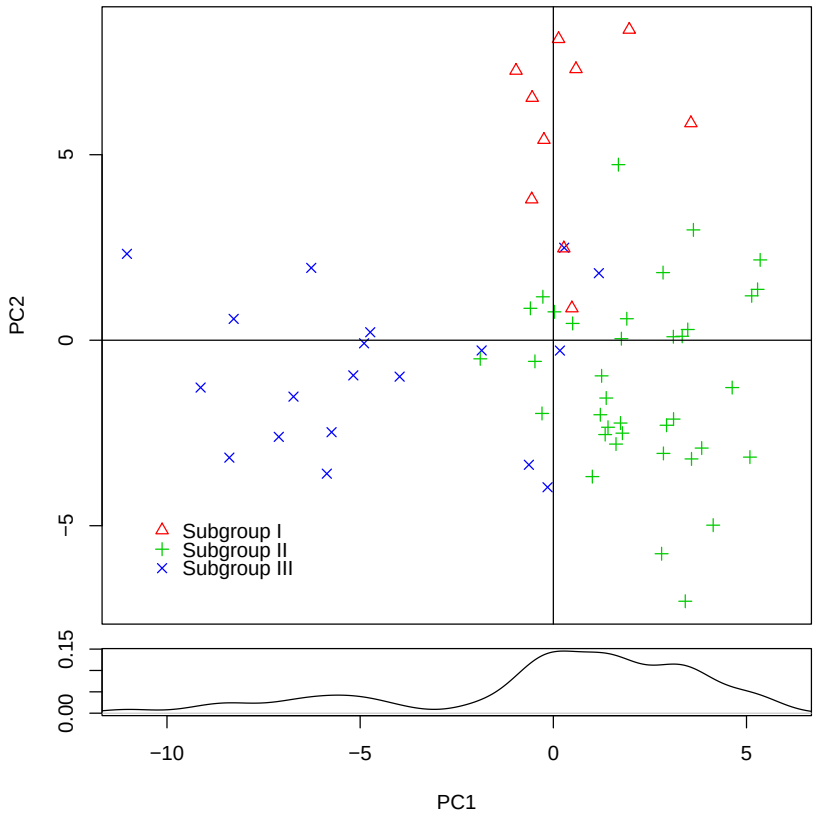

Figure 3.4: The score plot of the first two PCs (19% of the total variance) from the gene class *dephosphorylation*. All non-tumor samples have been removed along with tumor samples PT318, PT153, PT195, PT110 and PT21. This class received the highest score according to the model comparison score. The first PC shows a strong separation of subgroup III from I and II.

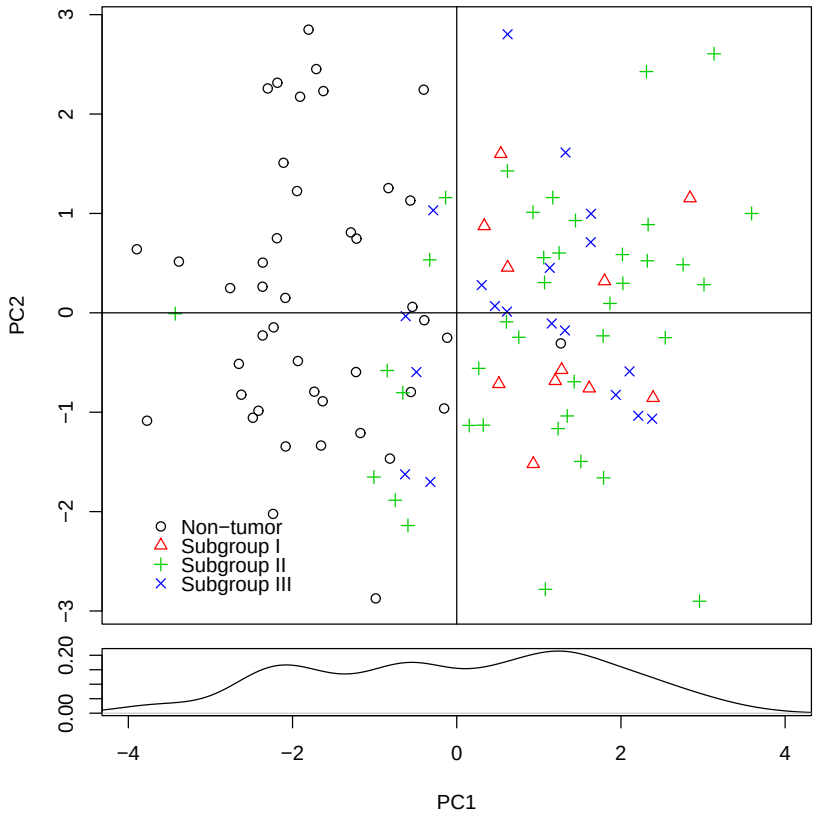

Figure 3.5: The score plot of the gene class with the largest fraction of differently expressed genes between tumor and non-tumor samples – *nerve ensheathment*. Even though the enrichment of differentially expressed genes in this class is highly significant, the sample groups are mixed on the first principal component and any clustering attempt would result in highly heterogeneous clusters.

of microarray experiments as a method for discovering the reasons for divergence between sub-groups of patients.

We restricted our trees to binary splits for simplicity and stability and tried two different ways of scoring these splits. The two propoposed scores seemed to perform equally well, overlapping strongly in the classes identied as relevant in the root node. Further down the trees the two methods started to diverge as differences get magnified rapidly.

Our main objective for the GO-UDT approach was to increase interpretability. We were not looking for better clusters but for a better biological explanation for clusters. This goal would have been well-fulfilled if it was possible to draw conclusions such as "protein biosynthesis was higher in the samples going left compared to those going right". Such inference can not be made using our trees. This does however merely reflect the complexity of gene expression data and biology as such. Single gene expression patterns, and possibly even short well-defined pathways, can be observed to be up or down regulated. Gene classes on the other hand are still quite heterogeneous and since they usually do not describe a well-defined biochemical event, the direction a specific process can go is not limited to up or down but is better viewed as being *divergent* or *differential* between samples. GO-UDTs can identify such divergent gene classes using expression data. Our analysis shows that these gene classes are indeed biologically relevant and can point to possible refined analyses and further experiments.

Chapter 4

Interactive Exploration of heterogeneous Data with ToPNet

ToPNet is a software tool for the analysis of biological data in form of networks and associated annotations. It was developed in joint work with Daniel Hanisch and published by Hanisch et al. (2004). Over the last years, the software company BioSolveIT GmbH, St. Augustin, has joined the development with one developer, Sabine Trochim, and now distributes ToPNet. Further contributions come from two students at the Ludwig-Maximilians-Universität at Munich, Maria Piskarev and Theresa Niederberger.

ToPNet provides methods for exploring biological networks with heterogenous annotations. The paradigm that guided the development of ToPNet was that the analysis of complex biological systems cannot be done completely automatically. Besides high quality networks and annotations, a human expert is necessary to direct the analysis. Therefore, besides the development of new algorithms for the analysis of biological data, visualization methods for networks and annotated data as well as an appropriate user interface were desired.

4.1 ToPNet Concepts

ToPNet was designed to handle several *networks* from multiple sources which can be restricted according to user-specified criteria and manipulated using a simple graph editor. Properties of the networks (e.g. color, size and hyperlinks) can be associated with *annotation data* (e.g. expression data or functional annotations) via *mappings* of names or identifiers. Figure 4.1 shows how these components work together to provide a useful visualization of networks and annotation data.

4.1.1 Representation of networks

Networks are implemented in the form of bipartite graphs in which one set of nodes (termed places) represents molecules (e.g. proteins or metabolites) and the other set of nodes

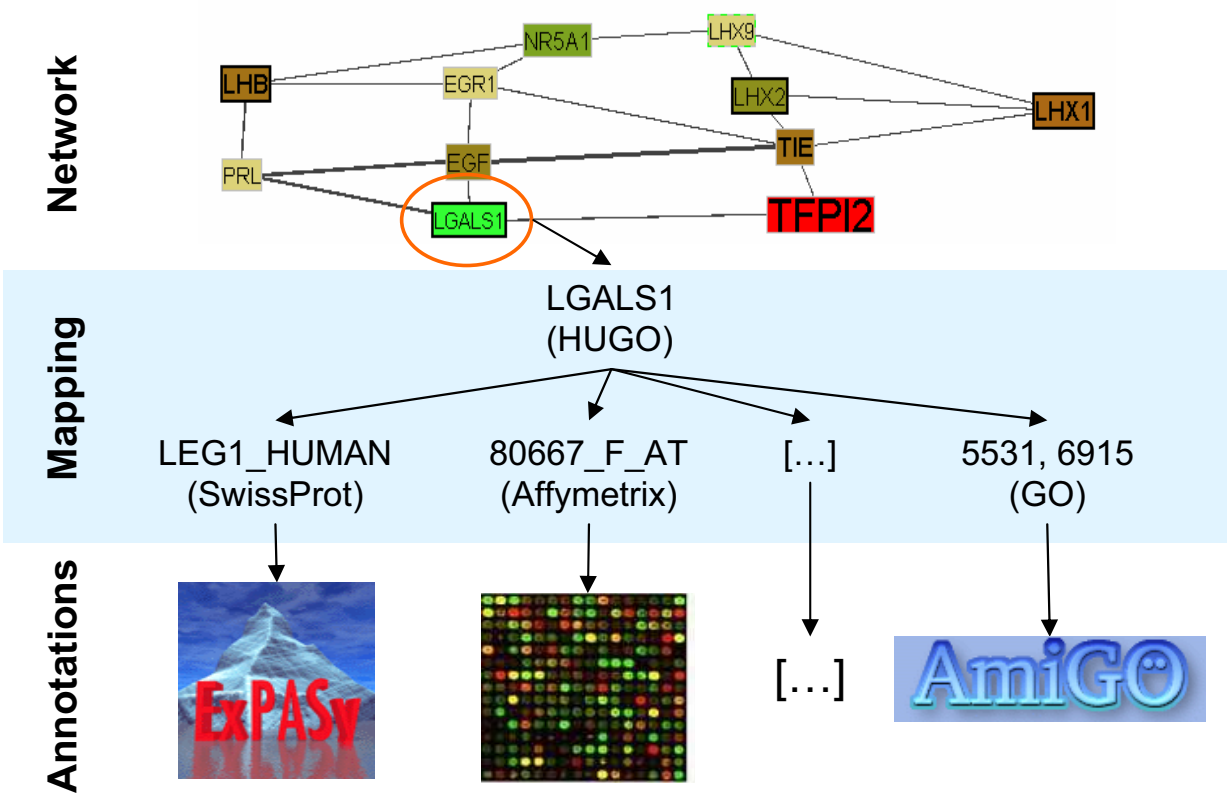

Figure 4.1: Visualization of a network with annotations. Each node (corresponding to a gene) is mapped to several identifiers linking it to relevant annotations. Annotations are displayed as properties of the network (e.g. size and color of nodes).

(called transitions) defines relationships among these molecules. This representation is valuable when complex reactions are considered. For instance, in metabolic networks places represent metabolites and enzymes. One metabolic reaction is then represented by a transition vertex which is connected via edges to the participating molecule places (Figure 4.2). Other network types include regulatory, protein-protein interaction and literature networks.

4.1.2 Network sources

Databases with information on biological networks become more and more abundant. At the moment, networks from KEGG (Kanehisa, 1996), TRANSFAC (Wingender et al., 2000), TRANSPATH (Schacherer et al., 2001), and DIP (Xenarios et al., 2000) can be imported. Furthermore, automatically generated text-mining networks are provided, as described in Hanisch (2004).

For KEGG, a graphical interface is provided that allows the user to download and

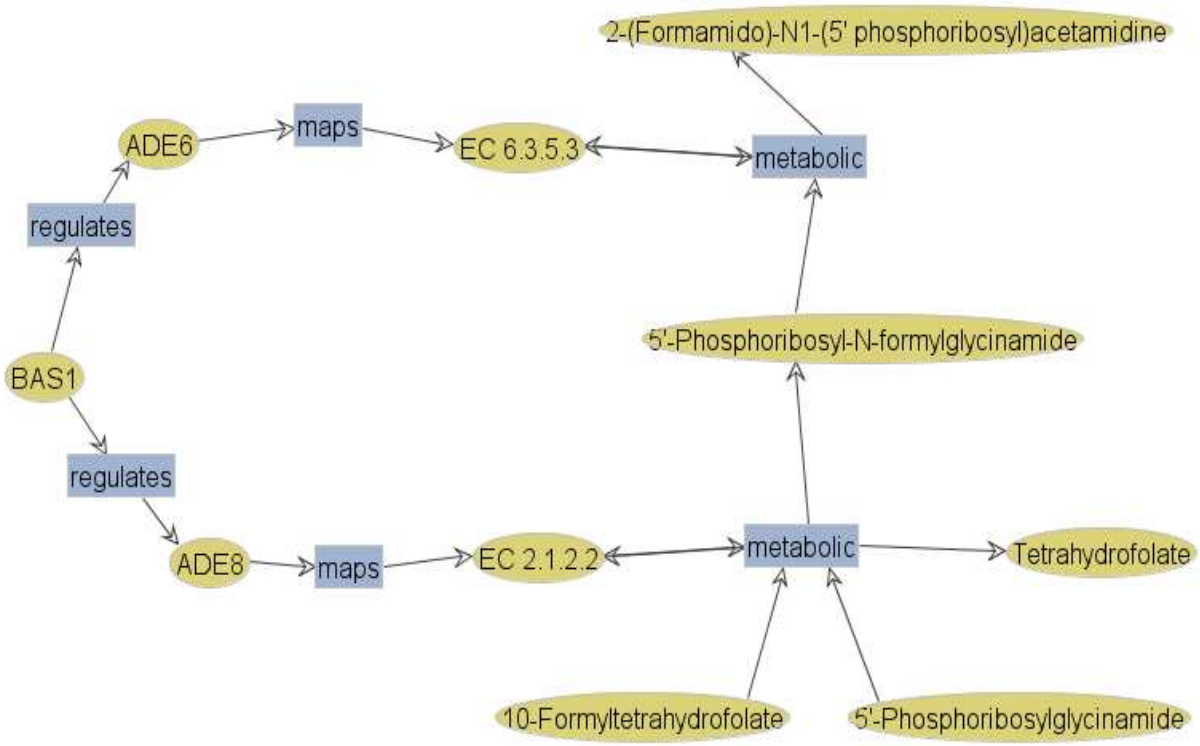

Figure 4.2: Petri Net visualization of regulatory and metabolic reaction where places are displayed as ellipses and transitions as rectangles. In yeast, the transcription factor BAS1 regulates ADE6 and ADE8 which participate as enzymes in two reactions of the purine metabolism pathway.

update pathways and appropriate mappings for any organism provided by the database. KEGG pathways can then be displayed in the original layout, as the necessary information is encoded in the KEGG markup language (KGML) files that are provided.

4.1.3 Visualization of networks

Visualization of networks is important to present biological models and algorithmic results in a concise way. A good visualization of a biological network has to serve several purposes:

- Provide a good overview of the molecules and their relationships in the network.
- Show important data and annotations.
- Provide a context and facilitate interpretation.

Network layout

In order to attain the first goal, the network has to be laid out clearly with few intersecting edges and overlapping nodes. In ToPNet, this is achieved using layout algorithms like the Fruchtermann-Rheingold spring embedding algorithm (Fruchtermann and Reingold, 1991). Spring embedding algorithms treat edges as springs and iteratively optimize the layout of a network by applying a force to each pair of connected nodes that is proportional to the deviation from the optimal distance. Additionally, a repulsive force is introduced for each pair of nodes. In each iteration, all forces affecting a node are summed up and the node is displaced in the direction of the resulting force by an amount proportional to the magnitude of the force.

Data visualization

First of all, the biological objects represented by the nodes have to be recognizable for the user, i.e. annotated with names that are known to the user. In ToPNet, the user can choose between several available database identifiers or a common name to be attributed to a node.

All additional data and annotations are visualized via a standardized procedure that will be described in the next section.

Cellular compartments as context

One suitable context to display networks in, is the natural layout of the cell. We have developed a graph layout that uses the GO annotations for the cellular component to split the graph into several compartments. In each compartment, all proteins from the corresponding cellular component are drawn. The layout is then optimized using the spring embedding algorithm subject to the constraint that nodes cannot leave their compartment. We use the compartments *extracellular*, *plasma membrane*, *other*, *nucleus*, and *unknown*. The compartment *other* contains proteins that are present in the cell, but not within the nucleus or the membrane. This is mostly the cytoplasm or organelles like the Golgi apparatus or the ribosome. The compartments *extracellular*, *plasma membrane*, and *nucleus* contain proteins that are annotated with the respective GO terms or any term below these in the GO hierarchy. If a protein's annotation does not provide a unique assignment, it is put into the compartment that contains most neighbors. If that is still not unique, or if proteins have no annotation at all, they are assigned to the *unknown* compartment. A typical result of the cellular layout algorithm is displayed in Figure 4.3. The *extracellular* compartment contains for instance several collagens, the *nucleus* compartment contains mainly cell cycle-related genes in this case. Depending on how the network was generated, a signal flow from the ligands to the transcription factors could be hypothesized.

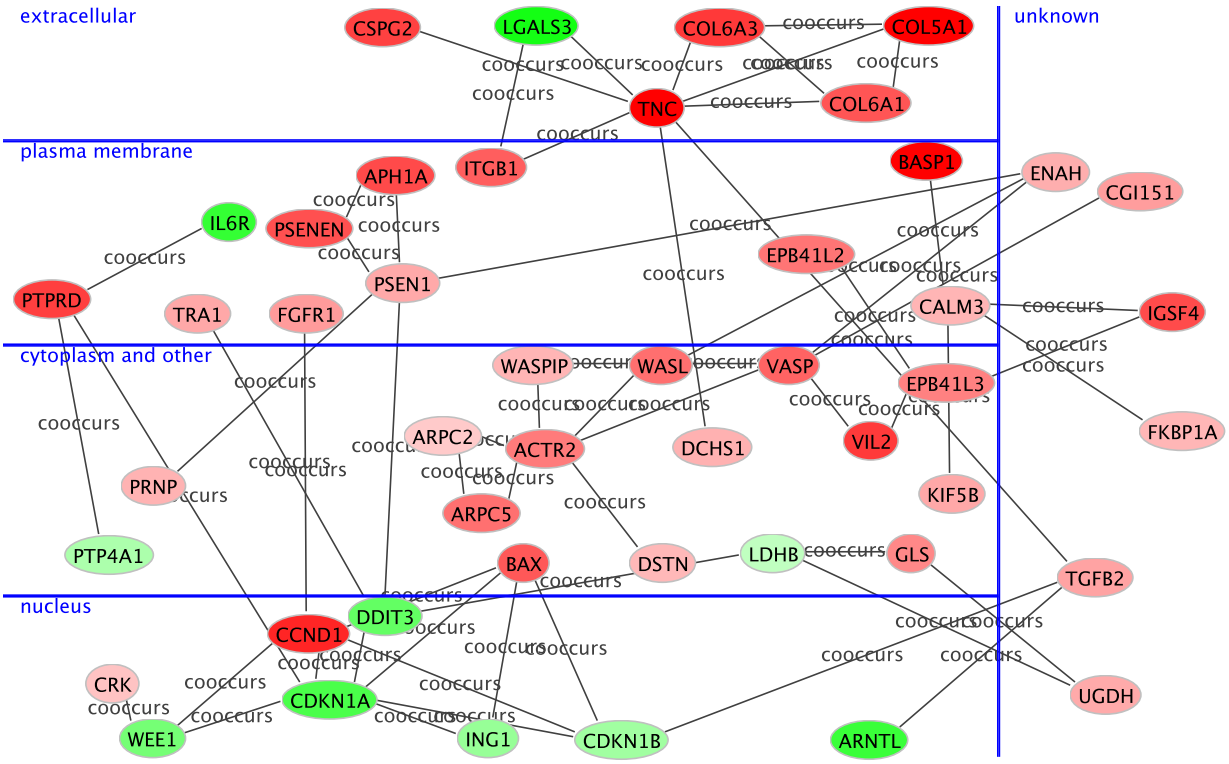

Figure 4.3: The cellular layout algorithm. The display is divided into four different areas representing cellular localizations. Gene ontology annotations are used to assign each displayed protein to one of the fields. If no annotations are available, the protein is assigned to a special area representing the *unknown* class.

4.1.4 Annotations for networks: data maps

Data maps handle annotation data in ToPNet by providing standardized information about their content. Data from *data maps* can be displayed as a textual description in the node labels or tool tips, as node size, as color, and as hyperlinked web sites. For example, expression data is often available in tabular format where rows represent genes and columns correspond to specific experimental conditions. The table itself contains e.g. probability values or fold changes quantifying differential expression. A *data map* then provides information for a gradient color coding of corresponding genes on a linear or logarithmic scale, for the size property of displayed molecules and annotation with the corresponding value as a tool-tip. As another example, terms from GO can be treated as a *data map* in ToPNet, thereby associating a set of GO terms with each gene. This data map provides, besides direct annotation of places with terms, the useful possibility to directly link to the corresponding entries in the GO hierarchy via a web browser. Table 4.1 shows some available general data types in ToPNet and their visualization.

Data type	Color	Size	Label/Tool-tip	Hyperlink
Expression data p-values	color gradient	size gradient	value	-
Expression data ratios	color gradient	size gradient	value	-
GO annotations	colored if available	-	GO term	Amigo GO browser
Database IDs	colored if available	-	Database ID	Corresponding database web site
Pubmed IDs	colored if available	-	Pubmed ID	Articles in Pubmed

Table 4.1: Some available data types and their visualization with the corresponding *data maps* in ToPNet.

4.1.5 Providing the link: mappings

To connect annotation data to network properties, a *mapping* is essential. As several major gene and protein databases exist and a general nomenclature for protein and gene names is still missing, ToPNet is able to handle sets of mappings for different sets of identifiers. Such mappings are defined using a simple tabular file format, allowing easy generation of custom mappings. They can be imported interactively and are then available through a special user interface. This interface allows to activate or inactivate mappings, such that there is a very flexible user control over the set of mappings that is used for a certain task. ToPNet builds a mapping graph (Figure 4.4) from the active mappings, and searches for paths in this graph if a mapping is required that is not directly available. Thus, the set of possible mappings is given by the transitive closure of the mapping graph.

As most gene and protein databases provide links to some other database (usually a well known standard database) but not to all databases that might be of interest to the user, a combination of different mappings is often necessary. For instance, it might be desired to map expression data collected from a disease model in mice to a human protein-protein interaction network. Such a task requires mapping of the identifiers of the respective microarray to a mouse gene database and then to orthologous human genes and finally to a human protein database. In a bachelor's thesis, Travis Holton implemented a database and user interfaces that are capable of computing transitive mappings according to different rules and user requirements (Holton, 2003). This database was used to generate specialized mappings that can then be imported into ToPNet. In recent years, the situation has improved a lot due to standardization efforts by large institutions like the National Center for Biotechnology Information (NCBI) of the United States National Institutes of Health (NIH). They provide mapping information between many heavily used databases. Furthermore, microarray manufacturers usually provide mappings of their spot identifiers to databases from NCBI or other standard databases. Still, mappings can contain errors, for instance if they are generated automatically using sequence alignments or other approaches. For specialized tasks, like mappings covering ortholog relationships or common names, transitive mappings are still necessary.

A similar approach to that of Holton (2003) was taken by Iragne et al. (2004). Their alias server is available via a web interface[1] and a SOAP API. The server takes a database identifier of a protein and a species code (taxonomic identifier) and returns identifiers of the same protein from other databases. We have implemented a mapping class that uses the SOAP API of the alias server. This class can be used like any other mapping (that relies on tables on the file system), but it is not used for transitive mappings. Thus, we have access to an externally maintained server for mappings and all aliases available from that server. In principle that could significantly reduce the manual efforts necessary to generate new mappings and keep old ones up-to-date. Unfortunately, a single call to the SOAP API takes approximately one second, rendering it useless for anything but very small graphs.

Because of the multitude of different possible links, such as orthology, genes encoding proteins, etc., and the huge number of possible combinations of such mappings, a database for mapping information remains useful, but the design in Holton (2003) and Iragne et al. (2004) has to be improved in several ways. Most importantly, different mapping types must be supported. The simplest type of mapping is the mapping of different identifiers for the same object. Other types could include mappings for genes encoding proteins, homology, orthology, splice variants etc. It should be useful to develop an ontology covering possible mapping relationships between genes and proteins. Such an ontology should contain identity, similarity, and transformation relationships and information about the method that was employed to infer the links. It could be incorporated in a general ontology of gene and protein relationships, unifying network and mapping information in a graph view of the data. Specialized transitive mappings could then be expressed as paths between objects

[1] http://cbi.labri.fr/outils/alias/

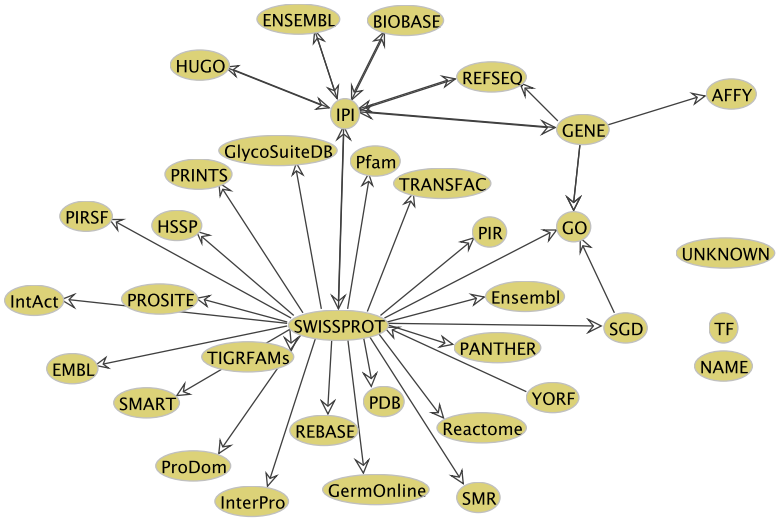

Figure 4.4: An example of a mapping graph. Here, mappings were imported connecting Swissprot and IPI with several other databases.

connected by certain mapping information. Such a design could be utilized immediately by special *pathway queries*, which will be described in chapter 5.

4.1.6 Network exploration

For interactive exploration of the data, gene sets can be selected according to user-defined criteria. These criteria are specified via boolean functions defined on data maps. For example, given that probability values and GO annotations are available, the following expression would select all apoptosis-related genes with a significant p-value: *GO biological process like apoptosis & pValue* $\leq$ *0.05*.
We call this kind of query *basic query*, they will be discussed in detail in chapter 5. Selected gene sets can be visualized as a network or further extended by graph operations. These operations include computing hulls around nodes, i.e. exploring the neighborhood of biological objects, or computing all shortest paths among selected nodes. In conjunction, the selection, manipulation and visualization options provide the basis for efficient interactive exploration of the gene expression data in the context of the given network.

4.1.7 Data Integration

One important goal in the development of ToPNet was to provide a tool for the integration of different data and annotation types in biological networks. In the preceding sections, the mechanisms used in ToPNet have been explained. Taken together, these mechanisms provide a powerful environment that allow algorithm developers as well as users to easily access heterogeneous data. For instance, it is possible for a user to visualize a graph using

several different annotations like expression values to color the nodes, GO annotations as tool-tips and hyperlinks to a protein database like Swissprot.

A developer can implement a new data type and provide the methods necessary for its visualization. The new data type can then be visualized by all users and it can also be used for custom algorithms that rely on special properties of that data type.

Figure 4.5 illustrates how ToPNet handles requests for a certain data type for a node in a graph: The user requests a customized visualization or an analysis algorithm which needs data from a certain *data map* for a graph node. The *data map* specifies for what kind of database identifier it contains the required information. The graph node itself contains a set of identifiers from different databases. Thus, a mapping from any of the available database identifiers to the database required by the *data map* is necessary. If such a mapping exists, it is used, otherwise it is attempted to build a transitive mapping by searching shortest paths in the mapping graph. If no such path exists, the mapping procedure fails, and the requested data cannot be provided. When a suitable mapping was found or generated, the identifiers from the graph node are mapped to the database that is required by the *data map*. The *data map* then provides the annotations according to the mapped identifiers. If there is more than one resulting annotation, e.g. if the mapping is not unique, a *data map* can apply a specific aggregation function. Such an aggregation function can simply result in all annotations being displayed or in some real aggregation, like the computation of a mean value for numerical data. The generated annotation is then passed on to the vizualisation or analysis algorithm, which requested it.

4.1.8 Scripting

ToPNet is mainly intended to serve as an interactive tool, but it was also desired to include some scripting capabilities in order to automate certain processes and add functionality without much programming effort. Therefore, we have provided an interface to the Bean Shell[2], a Java source interpreter, allowing a user to write scripts interactively using Java syntax and the full ToPNet API. Scripts can also be stored in a special directory and then become available via the scripting menu in the ToPNet main window.

A special application of these scripting capabilities are the scriptable *data maps*, which allow customization of *data maps* using scripts. For instance, it is possible to create a *data map* that behaves exactly like the standard GO *data maps*, but in addition displays all nodes belonging to a certain GO class in a specified color.

4.2 Algorithms

The goal of ToPNet is to facilitate the generation and verification of biological hypotheses. Besides the visualization of annotations, ToPNet provides several means to restrict networks to interesting regions with respect to experimental data.

[2]http://www.beanshell.org

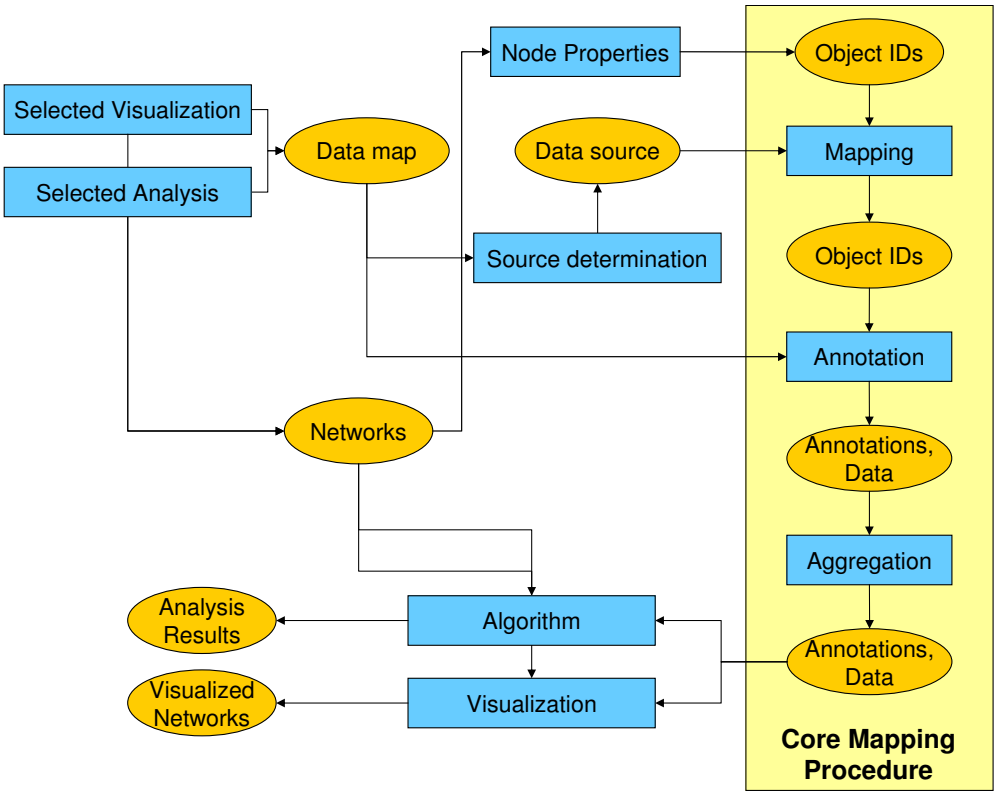

Figure 4.5: The mapping algorithm and how it works together with visualization and analysis requests. Such a request requires annotations from a certain *data map* to be mapped on a network. The *data map* in turn requires certain object identifiers (database keys) in order to find the requested annotation. If the identifiers available at the network nodes do not contain the required types, a mapping is constructed using shortest paths in the mapping graph. Then, annotations are produced and aggregated if necessary. These annotations are returned to the visualization or analysis algorithm for processing.

4.2.1 Significant Area Search

The *Significant Area Search* algorithm developed by Daniel Hanisch (Sohler et al., 2004; Hanisch, 2004) aims at detecting connected parts of the network, which are significant according to specified p-values. These might correspond to co-regulated pathways in metabolic networks or functionally related proteins in literature networks. The algorithm selects a set of seed genes according to a specified threshold and starts a greedy expansion by including the most significant neighboring molecule in each step. The significance of the selected gene set is quantified by combination of individual p-values using Fisher's inverse chi-square method (Fisher, 1932), which quantifies the probability that all individual values result from their respective null distributions. The individual p-values are adjusted for greedy selection based on local graph topology. This avoids the construction of subnetworks which are connected only via unspecific high-degree nodes. The detected significant areas are collected and pruned for highly overlapping redundant graphs. The resulting graphs are reported to the user in order of decreasing significance for further interactive exploration.

The algorithm described in Hanisch (2004) has been extended, such that it can now handle fold changes or any ordered values if no p-values are available. In order to do so, a new *data map* is generated that computes p-values according to the rank of the original data. If the original data are given as a function f on a graph with vertices V, the generated *data map* described by g is defined for each graph node $v \in V$ as

$$g(v) = \frac{|\{w : f(w) \leq f(v)\}|}{|V|} \tag{4.1}$$

Now, the *Significant Area Search* algorithm can be called with the function g providing the p-values. If f contains fold changes, the modified *Significant Area Search* algorithm will extract sub-networks with predominantly down-regulated genes. The $\leq$ operator in equation 4.1 can also be replaced by the $\geq$ operator, in which case the algorithm will extract up-regulated sub-networks.

4.2.2 Enrichment analysis

Two methods for detecting different kinds of enrichment have been implemented in ToPNet. The first one is based on Fisher's exact test as defined in 1.4.1: Given two sets of molecules with an intersection of size n in a network, the algorithm computes the probability of observing an overlap of size greater than or equal to n by chance. One convenient way to use this analysis, is to formulate two *basic queries*, compute the sets of matching molecules and a p-value for the size of the overlap. For instance, one could select molecules that are significantly up-regulated as the first set and molecules from a certain GO-class as the second set.

The second method takes a set of molecules and a data map that contains data that can be ordered (such as any numerical data). All molecules are ordered with respect to the data map, then a rank test (Wilcoxon-Mann-Whitney test, see 1.4.1) is used to

compute a p-value for finding molecules as far up or down the list by chance. This method is especially tailored for the use with raw expression data, if no p-values for differential expression are available. The algorithm can compute a meaningful p-value for any subnet and corresponding expression data. The subnet can be the result of prior analyses as long as those did not select molecules according to the expression values.

It is also possible to apply these enrichment analyses on lists of pathways in order to find out if there are pathways that are enriched with respect to a feature that can be described by a *basic query*. The most intuitive example is to look for pathways that are enriched in differentially expressed genes.

4.2.3 Pathway Query Language and Pathway Search

Pathway queries are another means to identify pathways or subnetworks that are interesting with respect to experimental data. Users from different areas of application will have specific restrictions as to what they consider interesting. For instance, in pharmaceutical research focus may be on pathways containing 'druggable' targets like kinases or phosphatases. A kinase could be considered interesting only if it phosphorylates a transcription factor which regulates genes that show a significant change in their expression pattern in a certain experiment.

To allow for such complex queries, we have developed an XML-based query language. In this language, *pathway templates* can be formulated as graph-like structures where vertices describe properties of the genes or proteins (e.g. must be a kinase or a transcription factor), and edges pose restrictions on the connections (e.g. the path between the kinase and the transcription factor must not be longer than two, or it must involve a phosphorylation). This language, the corresponding *pathway search* algorithm, scoring schemes, and visualization of the results will be described separately in chapter 5.

Chapter 5

Generating Contexts and Testing Hypotheses with Pathway Queries

5.1 Introduction

The methods introduced in chapters 2 and 3 attempt to analyze expression data using no additional context information or, as the unsupervised decision trees, using functional annotations. Including that kind of categorical data in the analysis is a very popular and fruitful approach, partly because of its conceptual clarity and the simplicity of the occurring statistical and algorithmic problems. Many biological phenomena, however, cannot be appropriately described in terms of functional annotations to genes or proteins; a representation as a network is much more natural. Such networks can represent very detailed information about the interplay of biological entities and make that knowledge available for the analysis of measured data. Using detailed network models for expression data analysis makes it possible to come up with very specific hypotheses about the data. Biological networks can be represented computationally as graphs, which is a well-studied data structure in computer science. Therefore, many basic graph algorithms, like path and distance computations, are readily available for the analysis of biological networks. In chapter 4, ToPNet was introduced, which provides many of these basic graph algorithms and some special algorithms for the analysis of expression data. This chapter is dedicated to a novel approach for exploiting network information in the analysis of expression data. The idea of this approach is to query existing networks representing biological knowledge in combination with experimental data. Before this approach is explained in detail, it is put into context with existing network-based analysis methods.

Special bioinformatics algorithms that work with expression data and biological networks can be classified using the following four categories according to their purpose:

1. Inference of regulatory networks from expression data
2. Refinement of biological network models or adding annotations
3. Interpretation of expression data using network information

4. Understanding and simulation of network dynamics

Algorithms of the first category aim at discovering regulatory relationships between genes and proteins with little or no prior knowledge about such relationships. The methods used range from simple graph algorithms (Wagner, 2001) to statistical methods like Bayesian network inference. Prominent works on network reconstruction include Friedman et al. (2000), the first work that proposed Bayesian network inference for the analysis of gene expression data. After introducing their model for expression data in the framework of Bayesian networks, the authors propose suitable learning algorithms and demonstrate the applicability of their methods on publicly available expression data from the yeast cell cycle. A review on network inference with graphical models can be found in Friedman (2004). A recent review on network reconstruction methods that also covers experimental methods is given by Lee (2005). A new approach for the reconstruction of signaling pathways was proposed recently by Markowetz et al. (2005). Here, measurements from RNA interference experiments were exploited in order to find interactions within signaling pathways where no transcriptional effects can be seen. While the results appear promising, the amount of data necessary makes the method practical only for small pathways. But it is also possible to use the method to extend pathways where most interactions are known before-hand. This set-up is of much practical importance and places the method also in the second category, which is explained next.

The second category deals with the refinement of biological networks based on experimental data like gene expression data. The goal is similar to the goal of the first category: To find a network that explains the given data best. But the advantage is obvious: As a part of the network is already given, the search space for the best network is greatly reduced. Furthermore, the search is directed toward biologically reasonable solutions, as the starting network represents biological knowledge. Tanay and Shamir (2001) define a fitness function based on expression measurements and propose an algorithm that finds an expansion of a given core pathway that is optimal with respect to the fitness function. A very recent, promising approach was suggested by Gat-Viks et al. (2005). They use factor graphs to model biological systems, show how existing knowledge can be represented in such factor graphs, and propose learning algorithms to refine the given models using expression data. Factor graphs have also been used by Yeang and Jaakola (2003) for annotating a given interaction network with directions and modes of regulation (activation or inhibition).

The third category is concerned with a somewhat less ambitious goal. Instead of finding new regulatory mechanisms, the goal is to use current knowledge on such mechanisms (or any kind of interactions between biological entities) and guide the researcher toward hypotheses about active pathways, relevant biological processes or simply interesting subnetworks. This is more promising than trying to infer complete networks or new mechanisms if the measured data is not abundant enough to support such new findings. The first step in that direction is to map the measured expression data on known pathways and score and visualize the pathways according to these data. Grosu et al. (2002) identify metabolic pathways with a significant number of regulated genes using Fisher's exact

test. Technically, the graph representation is used for visualization purposes only. The computation relies on the annotation of pathways to genes without considering the graph structure. If predefined pathway annotations are unavailable or inappropriate, it can be necessary to construct pathways from the given networks and select interesting ones using expression data. Zien et al. (2000) propose a method to generate interesting metabolic pathways using expression data. They use a metabolic network to enumerate all pathways satisfying certain completeness constraints and then score these pathways according to the expression data. Similarly, Steffen et al. (2002) propose a method to infer signaling pathways from interaction networks built from high throughput experiments and expression data. They enumerate all paths in the network that start at a membrane protein and end at a transcription factor. The candidate pathways are then scored using a clustering of genes based on expression data. If the genes on a candidate pathway cluster together in the given clustering, the pathway receives a high score, if the genes are spread over many clusters, the score will be low. With this approach, the authors can reconstruct some signaling pathways in yeast with good accuracy.

The two previously described methods have in common that they first generate pathways using only the network data, and then apply a scoring function that is based on expression data to identify interesting pathways. Other algorithms use expression data and graph structure simultaneously for determining interesting subnetworks. For instance, the Significant Area Search algorithm (Sohler et al., 2004; Hanisch, 2004) implemented in ToPNet detects connected sub-networks significantly enriched with regulated genes. Co-clustering (Hanisch et al., 2002) finds groups of genes with similar expression profiles and small distance in the network. Ideker et al. (2002) introduced another method for identifying sub-networks exhibiting significant changes of expression.

The last category covers the dynamics and simulation of biological networks. Today, this is considered a part of the field of systems biology, which has become very popular, recently. Websites like `systems-biology.org`[1] provide an overview over available resources including lists of publications, free and commercial software, common standards like the systems biology markup language (SBML), and databases of network models for biological systems like regulatory or metabolic pathways. Especially for metabolic networks, dynamics have been studied for a long time, although gene expression data are not often taken into account. Schuster et al. (1999) review the use of elementary flux mode analysis in metabolic networks. This is often termed a static analysis of metabolic network because it can be performed on static data only (stoichiometric structure and reversibility of reactions). But elementary modes correspond to steady states of the network *fluxes*, i.e. states where no internal metabolites are consumed or produced, therefore, we will consider it an analysis technique of network dynamics. In Schwarz et al. (2005) flux analysis is combined with gene expression data to find the fluxes of a network with the corresponding gene expressions. Voit and Radivoyevitch (2000) apply biochemical systems theory to the analysis of gene expression data in metabolic networks. Biochemical systems theory is based upon differential equations that describe the properties of enzymatic reactions.

[1] http://www.systems-biology.org/

Other simulation methods make use of hybrid Petri nets, e.g. the Genomic Object Net software (Matsuno et al., 2000; Nagasaki et al., 2004).

The methods presented here belong to the third category. We are interested in a detailed interpretation or mechanistic explanation of expression data. Network information in combination with associated annotations such as pathways or functional annotations are used to find biological contexts that appear conspicuous or significant with respect to the expression data or that can provide a hypothesis for a causative mechanism for the expression data.

If context information is not taken into account, analysis methods typically result in a list of genes that exhibit a relevant expression behavior in the experiment under consideration. While this is an important first step in understanding the data, it does not reveal the causative biological mechanism of the observed gene expressions. For pharmaceutical applications for instance, it could be interesting to measure the expression profile of an in vitro or in vivo disease model and compare it to a healthy expression profile. First of all it is necessary to find genes that are differentially expressed between the healthy and diseased states. However, a more interesting goal for the development of disease modifying drugs is finding the molecular mechanism that causes the observed changes in gene expression. Unfortunately, this mechanism does not need to be reflected by changes of gene expression; gene regulation often relies on molecular events other than transcription, such as protein modification (phosphorylation, cleavage), translocation, DNA methylation, etc. Thus, the causative mechanism cannot be identified using gene expression data alone. But if a hypothesis about the relevant mechanism is available, it can be tested on the basis of expression data and prior knowledge in form of a network model. Such a hypothesis could be that a certain kinase is active and phosphorylates one or more transcription factors which cause the observed differences in the expression profiles. This hypothesis can be visualized as a small network as shown in Figure 5.1. We call such network templates *pathway queries* and have developed a language for their specification and an algorithm to find specific instances in a given network representing prior knowledge.

Recently, a similar approach has been taken up by Leser (2005) who proposes a pathway query language to describe graph templates and find these in a graph database. While their approach is also aimed at life science applications, it is more concerned with the technical questions arising when implementing such a system within a database management system and when querying large databases. Our focus instead is on the biological applications, integration of different data sources and statistical scoring of the results.

5.1.1 A language for network-based hypotheses in molecular biology

The *pathway query language* allows specifying templates for biological networks using functional annotations of genes or proteins and their interactions. In many cases it is possible to translate hypotheses about the biological processes relevant for the measured data into such network templates. E.g. finding an instance of the template described in Figure 5.1

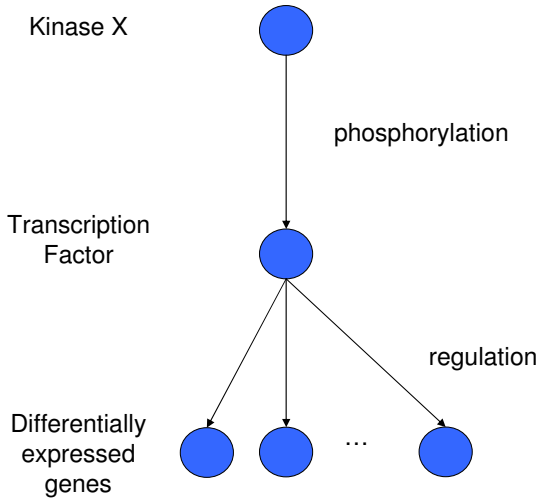

Figure 5.1: Example of a simple pathway query. This query describes sub-networks containing differentially expressed genes regulated by the same transcription factor that can be phosphorylated by a certain kinase X.

can be evidence for the hypothesis that the differentially expressed genes found in that instance are regulated by the corresponding transcription factor which might be activated or inhibited by the kinase X. In other cases, the *pathway query* may simply be viewed as a definition of the context in which the expression data should be analyzed. For instance, it is possible to define a *pathway query* that matches all sets of proteins that are connected to a certain protein of interest P in a text-mining network by at most two edges and share a common function (e.g. as annotated by GO). Instances of that query could be viewed as representing the P and related proteins in the context of the given function.

Given a *pathway query* and a biological network, we enumerate all instances of the query in the network using the *pathway search* algorithm which solves a special version of the subgraph isomorphism problem. In many situations it is necessary to examine rather unspecific queries so that many matching instances may be found. Therefore, the statistical significance of each instance has to be assessed. As the *pathway query* defines an individual context for the data, the scoring function may have to be defined individually as well. Some generally applicable scoring schemes have been implemented for *pathway queries* and will be described in Section 5.4. These scoring schemes can be combined in order to define versatile custom scoring methods. Furthermore, it is easy to implement new scoring methods based on the ToPNet API, which can then be used in all *pathway queries* and again combined with other scorings.

In general, conducting an analysis with *pathway queries* on a new data set involves four steps:

1. Develop a *pathway query* that describes the hypothesis or context.

2. Assemble networks that contain the relevant information.
3. Devise a scoring scheme to identify significant instances.
4. Run the *pathway search* algorithm and evaluate the results.

All of these steps are critical for a successful analysis. The first step can best be done by a biological expert. Steps two and three will in most cases need the cooperation of a computer scientist and a biologist, although some predefined scoring schemes can be specified in the *pathway query* and some general networks can be easily supplied. The *pathway search* algorithm is implemented in ToPNet. After all necessary data is imported, it can be started on a selected *pathway query*. All instances will be listed together with the computed score. The *pathway query language* also provides special elements for layout information that will be used by ToPNet when specific instances are visualized. That way, the user can quickly get an overview of an instance because it is displayed in a well-defined way. Therefore, the evaluation of instances is facilitated as no time and effort is lost by restructuring the layout.

5.1.2 Some possible applications

Identifying relevant pathways

If some constraints about pathways relevant for the expression data are known, a *pathway query* could generate pathway hypotheses for the data. Figure 5.1 constitutes a simple example of that kind of query, but many variations are possible, either going further upstream or including additional constraints like feedback loops, significant expression values on signaling molecules or restrictions on the target genes. Besides signaling pathways, it is also possible to design *pathway queries* for metabolic pathways.

Linking measurements to known relevant processes

In the common case where expression measurements were taken from diseased tissue or a disease model and compared to normal tissue, it is of interest to find links between genes with conspicuous gene expression and known disease-relevant processes. If genes are annotated with associated diseases, a *pathway query* could find all paths between conspicuous genes and genes annotated with the disease of interest and thus provide a possible link.

Finding evidence in text mining networks

Another interesting application of *pathway queries* is in conjunction with text-mining networks such as co-occurrence networks. In a co-occurrence network there is an edge between two nodes representing biological entities if these entities appear together in a sentence or an abstract in a given body of scientific literature (e.g. all abstracts in the Medline database). The generation of such networks is described in Hanisch (2004). Although

the number and size of curated databases describing all kinds of molecular interactions is steadily growing, most of the relevant knowledge is only available in textual form in scientific journals. Therefore, the most comprehensive network available is often a text mining network derived from the scientific literature. Using *pathway queries* on a co-occurrence network can help finding relevant literature for the data under investigation. While the interactions between genes or proteins in such a network are usually quite unspecific, the number of possible meanings can often be narrowed down a lot by putting them into a biological context. For instance, evidence for regulation through a certain transcription factor can be found by identifying regulated genes that are connected to that transcription factor in a literature network and a predicted binding site. This approach was used in Gebauer et al. (2005) where the transcription factor NF-κB was shown to be important for IL-1β-induced gene expression in two different cell types. The main results of that paper will be discussed in chapter 6.

5.2 Description of the query language

In this section, we specify the *pathway query language*, the language that is used to describe the network templates (*pathway queries*), and its semantics, i.e. which instances will be matched by a query. The *pathway query language* is based on the extensible markup language (XML) and was designed to allow easy description of biological network templates in the ToPNet framework. It contains elements that allow for selecting places and transitions and paths between these. In order to give a first idea of the *pathway query language*, Figure 5.2 shows a simplified version of the most important elements in extended Backus-Naur-form.

First, we define the search graph on which the *pathway search* algorithm works. The search graph is represented as bipartite graph $G = (P, T, E)$ with places P representing genes, proteins or other molecules and transitions T specifying relationships between the molecules. The edges E are used to define which molecules participate in an interaction and (if appropriate) the direction of the interaction.

A *pathway query* is a network template that describes a context of interest as a small sub-network which should be found in G. Formally, a *pathway query* is described as a labeled graph $Q = (N, C)$. Nodes $n \in N$ represent subgraphs defined by another *pathway query* or single places defined by *basic queries*, which are defined below; edges $c \in C$ represent paths between places. We divide the set of nodes N into two subsets $N = N_r \oplus N_s$ where N_r stands for recursive and N_s for simple nodes. All nodes $n \in N_r$ are labeled with a *pathway query* denoted by $\nu(n)$. All other nodes $n \in N_s$ are labeled by a *basic query* that is also denoted by $\nu(n)$. In the simplest and most common case, N_r is empty, so all nodes in the *pathway query* represent simple nodes. For simplicity, we will assume that we have $N = N_s$ and describe the changes that occur when recursive nodes are introduced later.

Subnet	=	SubnetName PathwayNode
	|	SubnetName Subnet {Subnet} {Connection}
PathwayNode	=	BasicQuery [BooleanOperator BasicQuery]
BasicQuery	=	DataMap Operator Literal
Connection	=	'From:' SubnetName
		'To:' SubnetName
		PathRestrictions
PathRestrictions	=	'Places:' PathwayNode
		'Transitions:' PathwayNode
		'Max Length:' Number

Figure 5.2: A simplified version of the *pathway query language* in extended Backus Naur form. This definition is not complete and serves only as a quick overview. More details (e.g. on *basic queries* and the restrictions that can be associated to a path) can be found in the text. A complete definition of the grammar is given as an XML schema in appendix A.

Query String	Numerical data	Ontology data	Strings
'$A < B$'	math.	$A \in$ subtree(B)	A before B in alphanumerical order
'$A > B$'	math.	$B \in$ subtree(A)	A after B in alphanumerical order
'$A = B$'	math.	A equals B	A equals B
'A *like* B'	undefined	undefined	A matches regular expression B

Table 5.1: Semantics of operators for *basic queries*.

5.2.1 Specification of places, paths, and networks

The constructs used in the *pathway query language* to restrict sets of places or interactions are *basic queries* which can represent constraints like

$$(\text{ GO molecular function } \textit{like} \text{ Kinase }) \quad \& \quad (\text{ fold change} > 2\ ). \tag{5.1}$$

In this example, all kinases that are up-regulated more than twofold would be selected. Formally, a *basic query* q encodes a boolean function $\xi(q, .)$ that evaluates to true if and only if the argument is a place in the search graph that satisfies the conditions specified by q. In the *pathway query language* a *basic query* is represented by the `Query` tag which can contain multiple `BasicQuery` tags connected by boolean operators. The `BasicQuery` in turn has three elements: the `MapName`, the `Operator` and the `Value` elements. The `MapName` must refer to a *data map* available in ToPNet (e.g. expression data or functional annotations). The operator can be one of $<$, $\leq$, $=$, $>$, $\geq$ or *like*. While it is possible to redefine the meaning of these operators when a *data map* is implemented, they are supposed to follow standard semantics whenever possible. This standard is given in Table 5.1. The *basic query* described above is shown in its XML representation in Figure 5.3.

An edge $c \in C$ in Q corresponds to paths in G. These paths can be restricted in three different aspects. The places on that path must match a *basic query* associated with c,

```
<Query>
  <Query1>
    <BasicQuery Negated="false">
      <MapName>GO: molecular_function</MapName>
      <Operator>like</Operator>
      <Value>Kinase</Value>
    </BasicQuery>
  </Query1>
  <Operator>and</Operator>
  <Query2>
    <BasicQuery Negated="false">
      <MapName>fold change</MapName>
      <Operator>gt</Operator>
      <Value>2</Value>
    </BasicQuery>
  </Query2>
</Query>
```

Figure 5.3: The *basic query* from the expression (5.1) in XML representation. This *basic query* selects all kinases that are up-regulated more than twofold.

Symbol	Description
$\nu(n)$	*Basic query* associated with query node n
$\xi(q, .)$	Boolean function that is true if and only if the *basic query* q is satisfied for the argument.
$\eta(c)$	Maximum number of steps allowed for a path that represents the query edge c.
$\gamma_P(c)$	The *basic query* associated with the places on a path representing c.
$\gamma_T(c)$	The *basic query* associated with the transitions on a path representing c.

Table 5.2: Summary of symbols used to describe places and paths in a *pathway query*.

denoted by $\gamma_P(c)$. The transitions must match another *basic query* called $\gamma_T(c)$ and the length of the path is limited by $\eta(c)$. Note that there can be more than one edge between two nodes in the query. This is important if we look for instances that have more than one path between two proteins, e.g. different signaling cascades leading from one extracellular input to the same transcription factor. For simplicity however, we will only consider single edges between query nodes in this description. In an instance of query Q, a connection is represented by all shortest paths that satisfy the restrictions specified in the query. There are two reasons for computing shortest paths only: It is computationally more efficient than computing all paths of a given maximum length, and in most cases a large number of additional nodes is introduced that would make the results to hard to interpret. Table 5.2 summarizes the symbols introduced so far for the formal description of *pathway queries*. The following definition formally describes the paths that match an edge in the query graph.

Definition 5.2.1. A *legal path* for an edge $c = (n_1, n_2)$ in a *pathway query* is a path $p_1, t_1, \ldots, p_l, t_l, p_{l+1}$ in G such that

1. $\xi(\nu(n_1), p_1) = \text{true}$
2. $\xi(\nu(n_2), p_{l+1}) = \text{true}$
3. $l \leq \eta(c)$
4. $\forall i, 1 \leq i \leq l : \xi(\gamma_P(c), p_i) = \text{true}$
5. $\forall i, 1 \leq i \leq l-1 : \xi(\gamma_T(c), t_i) = \text{true}$

A legal path for an edge $c = (n_1, n_2)$ is a *valid path* for that edge, if no shorter legal path for that edge exists.

Thus, a legal path for an edge c is a path that contains only places and transitions that fulfill the conditions specified in the *basic queries* annotated to c, and that is also not longer than the specified maximal length. A valid path is a shortest legal path. Now, a first definition for sub-graphs matching a *pathway query* can be given.

Definition 5.2.2. A subgraph $G' = (P', T', E')$ of the search graph G is a *simple instance* of a *pathway query* $Q = (N, C)$ if there exist a function $\iota_{G'} : N \to P'$ and another function $\kappa_{G'} : C \to \text{Paths}(G')$, where $\text{Paths}(G')$ denotes the set of all paths in G', such that:

1. $\forall n \in N : \xi(\nu(n), \iota_{G'}(n)) = \text{true}$.
2. $\forall c = (n_1, n_2) \in C : \kappa_{G'}(c)$ is a valid path for c in G' with starting point $\iota_{G'}(n_1)$ and end point $\iota_{G'}(n_2)$.
3. G' is completely covered by the image of $\iota_{G'}$ and $\kappa_{G'}$.

If $p = \iota_{G'}(n)$ we say that p instantiates n in G'. Analogously, we say that a path pt instantiates c if $pt = \kappa_{G'}(c)$.

In a simple instance, each node in the *pathway query* must be matched by a single place and edges in the *pathway query* are matched by a single valid path. There are no nodes in the instance that do not instantiate a part of the query.

5.2.2 Aggregation of instances

Finding simple instances in the search graph is a very intuitive graph matching problem. It corresponds to finding sub-graphs that look exactly like the query, except that edges of the query can be represented by paths in the simple instance. The problem can be solved algorithmically using standard approaches, but it is computationally hard.

The main problem, however, is that in most cases the enumeration of all simple instances is not desired. Instead, we would like to aggregate many of the simple instances into a combined result. There are two reasons to do so: Firstly, the resulting instances should represent sensible biological units. Such a unit could be a kinase, a transcription factor and all its targets as in our first query example (Figure 5.1). As the number of targets is not known a priori, it should be possible to indicate in the query that all matching targets should be included in a matching sub-graph. More generally, the query language should allow to formulate queries containing all matching places for a query node. In our example, the query still contains only three nodes, but the node representing the transcription factor targets is annotated such that an instance should contain all matching targets. If we can find that kind of aggregated instances, the results will correspond to the biological contexts that the user is interested in. This will also allow to develop sensible scoring methods that reflect properties of the biological contexts as a whole.

The second reason for the aggregation of instances is that the number of simple instances may be exceedingly large. If with our example query, we find 10 matching kinases, each of which is connected to 10 transcription factors, each of which in turn has 100 targets, the number of simple instances is 10000. It is impossible for a user to inspect such a large number of instances manually. If all instances with the same transcription factor and kinase are merged together, i.e. if for a kinase and transcription factor all matching targets are aggregated, the number of instances drops to 100, a number that is still manageable. Simple instances as defined above also contain only a single valid path for each edge in the query. Thus, the number of instances becomes even higher when there are multiple valid paths.

The aggregation mechanism available in the *pathway query language* is controlled by the `multiplicity` attribute $\mu(n) \in \{0, 1\}$ for each query node. We call nodes n with $\mu(n) = 0$ simple nodes, N_s the set of all simple nodes, and $N_m = \{n \in N : \mu(n) = 1\}$ the set of all merge nodes. If $\mu(n) = 0$ the resulting instances will have exactly one place (or sub-graph) instantiating n. Otherwise, one instance will contain all places that are possible in that role, effectively merging all instances that are equal except for the instantiation of n (see Figure 5.4). Furthermore, we will generally assume that an edge in the query can be represented by many paths in an instance.

In order to cover these aggregation mechanisms, we have to extend the definition of matching instances.

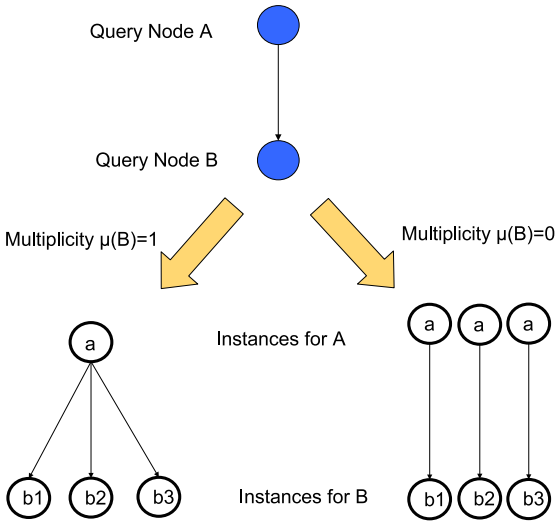

Figure 5.4: The `multiplicity` attribute in *pathway queries*. Setting the attribute to 1 effectively merges all instances that differ only with respect to that query node.

Definition 5.2.3. A sub-graph G' of the search graph G is a *merged instance* of the *pathway query* Q if it is a union of simple instances $G' = \bigcup_i G'_i$ under the following conditions:

1. $\forall i : G'_i$ is a *simple instance* of Q.
2. $\forall n \in N_s \; \exists p \in P \forall i : \iota_{G'_i}(n) = p$.

A *maximal merged instance* for Q is a merged instance that is no proper sub-graph of another merged instance.

Thus, a merged instance is a union of simple instances that have the same instantiation for all simple nodes. This definition is quite intuitive and solves the problems with aggregation. A maximal merged instance for the example query in Figure 5.1 is exactly what we have been looking for. It contains a single kinase, a single transcription factor, and all of the transcription factor's targets if the `multiplicity` attribute of the query nodes is set accordingly. In order to illustrate merged instances, Figure 5.5 shows two pathway queries containing merge nodes, simple instances and the resulting maximal merged instances. The definition can also be turned into an algorithm to find all maximal merged instances when a method for finding simple instances is given. All simple instances can be enumerated and merged whenever the instantiations of simple nodes coincide. The problem with that approach is that it still requires the enumeration of all simple instances. In order to avoid that enumeration, we give another alternative definition that is the basis of a more efficient matching algorithm and show that this definition is equivalent to the merged instances in most cases.

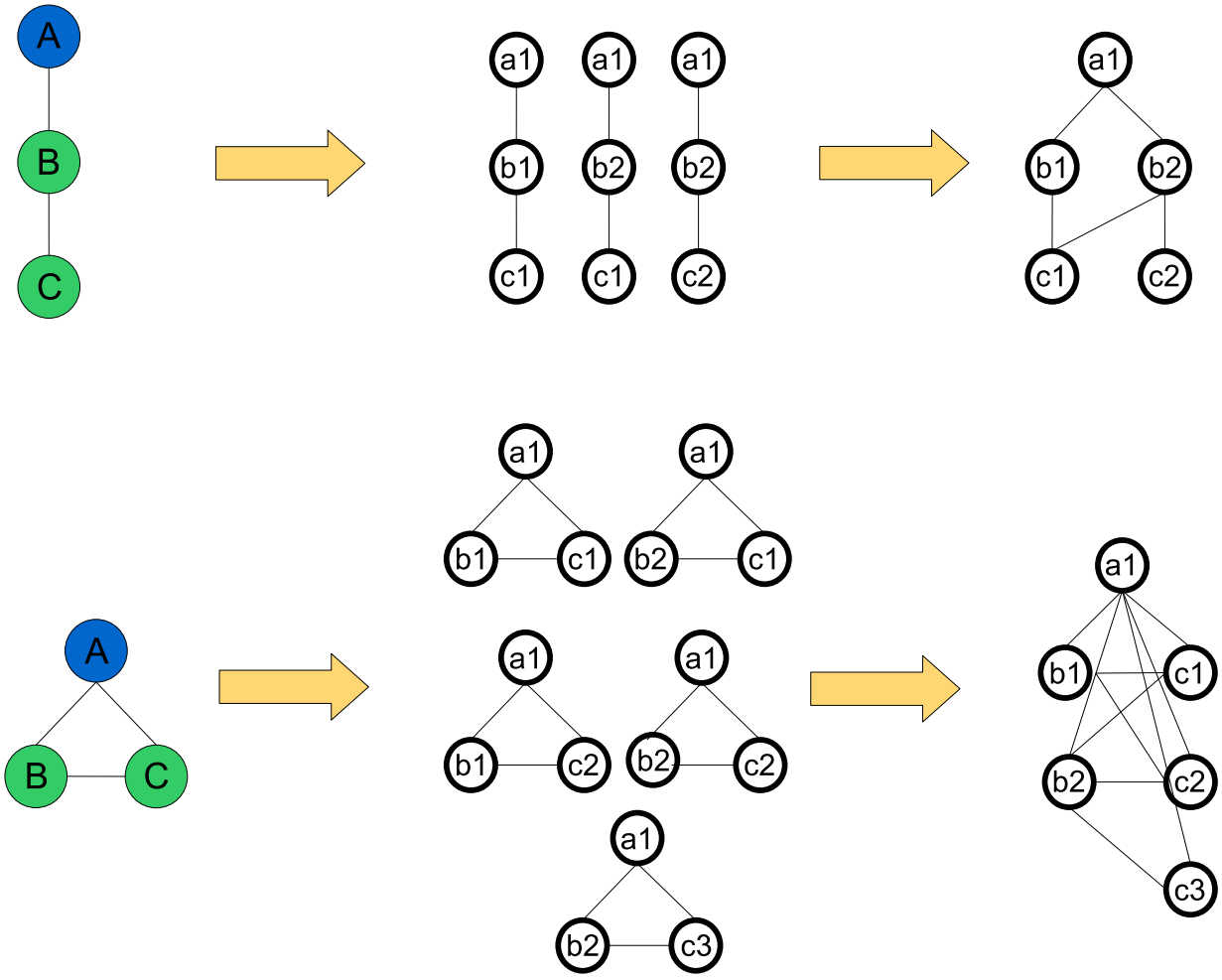

Figure 5.5: Two *pathway queries* containing merge nodes, corresponding simple and merged instances. On the left-hand side the *pathway queries* are depicted, simple nodes (A) are colored in blue, merge nodes (B,C) in green. In the middle, a number of simple instances is shown, which are merged into the merged instance on the right. In these examples there is only one merged instance, because a1 is the only instantiation of the single node A.

Definition 5.2.4. A sub-graph G' of the search graph G is a *matching instance* of a *pathway query* Q if there exist functions $\iota_{G'} : N \to \mathcal{P}(P)$ and $\kappa_{G'} : C \to \mathcal{P}(\text{Paths}(G'))$ (where $\mathcal{P}(X)$ denotes the power set of X) satisfying the following conditions:

1. $\forall n \in N \; \forall p \in \iota_{G'}(n) : \xi(\nu(n), p) = \text{true}$.
2. $\forall c = (n_1, n_2) \in C \; \forall P \in \kappa_{G'}(c) : P$ is a valid path for c in G' with starting point $p1 \in \iota_{G'}(n_1)$ and end point $p_2 \in \iota_{G'}(n_2)$.
3. $\forall n \in N_s : |\iota_{G'}(n)| = 1$.
4. $\forall n \in N_m : |\iota_{G'}(n)| \geq 1$.
5. $\forall c \in C : |\kappa_{G'}(c)| \geq 1$.
6. $\forall p_1 \in P' \; \forall c = (n_1, n_2) \in C :$
   $p_1 \in \iota_{G'}(n_1) \Rightarrow \exists p_2 \in P'$ such that $\iota_{G'}(n_2) = p_2$ and there is a valid path $P \in \kappa_{G'}(c)$ with start point p_1 and end point p_2.
7. $\forall p_2 \in P' \; \forall c = (n_1, n_2) \in C :$
   $p_2 \in \iota_{G'}(n_2) \Rightarrow \exists p_1 \in P'$ such that $\iota_{G'}(n_1) = p_1$ and there is a valid path $P \in \kappa_{G'}(c)$ with start point p_1 and end point p_2.
8. G' is completely covered by the image of $\iota_{G'}$ and $\kappa_{G'}$.

A *maximal matching instance* is a matching instance that is no sub-graph of another matching instance.

In comparison to the definition of simple and merged instances, definition 5.2.4 is not very intuitive, but it allows us to develop an efficient matching algorithm. The reason is that the conditions are more local, as they only refer to single nodes in the query graph or instance, while the condition for a merged instance is a global one. It must be a combination of sub-graphs that must be simple instances. Furthermore, the following theorem shows that for most *pathway queries*, matching instances and merged instances are the same.

Theorem 5.2.5. *A merged instance for a* pathway query Q *is also a matching instance for* Q*. If* Q *does not contain circles consisting only of merge nodes (*Q *is* μ*-circle-free), a matching instance is also a merged instance.*

Proof. By definition, a merged instance is a combination of simple instances with the same instantiation for simple nodes. Therefore, we have to show that (1) the union of simple instances results in a matching instance and (2) all instantiating places and valid paths in a matching instance are also part of a simple instance if Q is μ-circle-free. Without loss of generality, we can assume that Q is connected. Otherwise, we can apply the proof to each connected component separately.

1. If a set of simple instances $\{I_1 \dots I_k\}, k > 1$ is merged, we get a new graph H and define the functions ι_H and κ_H as follows:

$$\iota_H(n) = \bigcup_{i=1}^{k} \iota_{I_i}(n)$$

$$\kappa_H(c) = \bigcup_{i=1}^{k} \kappa_{I_i}(n)$$

With these definitions the conditions 1 and 2 of definition 5.2.4 hold as these are conditions on the elements of $\kappa_H(c)$ and $\iota_H(n)$, which are fulfilled as the elements come from simple instances. Condition 3 holds because for all simple nodes $n \in N_s$, their instantiation in the simple instances $\iota_{I_i}(n)$ is the same for all i according to the definition of a merged instance (5.2.3, condition 2). Conditions 4 and 5 obviously hold because $k > 1$. If there is an edge $c = (n_1, n_2)$ in the query and $p_1 \in \iota_H(n_1)$ then there is a simple instance I_i with $\iota_{I_i} = p_1$. This instance must contain a valid path for c with start point p_1. This path is also present in the merged instance. Therefore, condition 6 holds. Condition 7 can be verified analogously. Finally, condition 8 is true because the simple instances I_i are completely covered by the image of ι_{I_i} and κ_{I_i} and their images are added to the image of ι_H and κ_H. Thus, every merged instance is a matching instance.

2. Let Q be a μ-circle-free query and H a matching instance for Q. As H is completely made up of places that are instantiations of query nodes and valid paths, we only have to show that any such place or path can be extended to a simple instance contained in H. First, we build a spanning tree T for Q, such that none of the remaining edges connects to merge nodes. This can be easily achieved by starting with all edges that connect two merge nodes. The resulting graph is a forest, as Q is μ-circle-free. Then we use the remaining edges to complete the spanning tree.

   Now, let $p \in \iota_H(n)$. We construct a simple instance G' for Q with $\iota_{G'} = p$ along the spanning tree T. We use an arbitrary tree traversal algorithm starting at node n to determine the order of query nodes that will be used to build the tree. Let this order be given by $(n, n_1, \dots, n_{|N|})$. At the i-th step of the construction we add a node n_j and an edge $c = (n_i, n_j)$ or $c = (n_j, n_i)$ with $j < i$. According to conditions 6 and 7 of definition 5.2.3, there must be a valid path P in H that connects $\iota_{G'}(n_j)$ to a place $p_i \in \iota_H(n_i)$. We add P to G' and set $\iota_{G'}(n_i) = p_i$ and $\kappa_{G'}(c) = P$. After the traversal of T, we have instantiations of all query nodes and all edges in T. To complete G', valid paths for the remaining edges must be found. Let $c = (n_1, n_2)$ be such an edge. Due to the construction of T, either n_1 or n_2 must be a simple node. We assume without loss of generality that n_1 is a simple node. We have to show that H contains a valid path for c with start point $\iota_{G'}(n_1)$ and end point $\iota_{G'}(n_2)$. According to condition 7 of definition 5.2.4 there must be a valid path P for c with some start point $p_1 \in \iota_H(n_1)$ and end point $\iota_{G'}(n_2)$. As n_1 is a simple node, $\iota_H(n_1)$

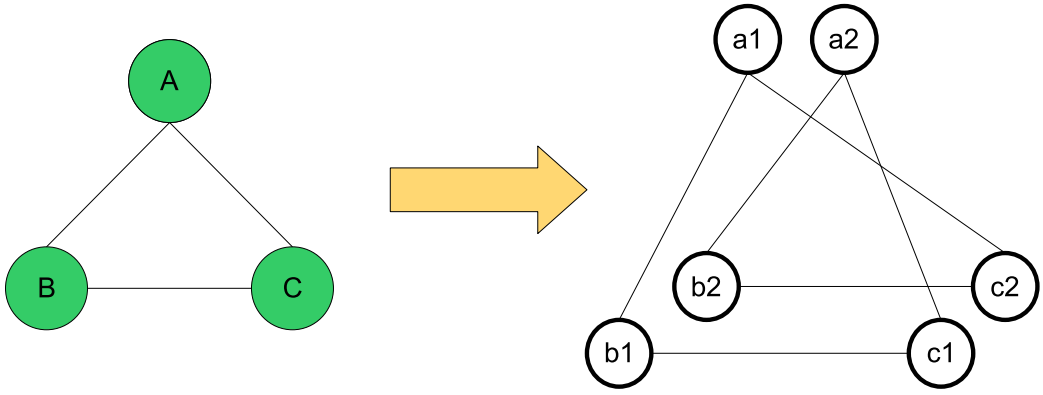

Figure 5.6: A *pathway query* and a matching instance that is not a merged instance. If all nodes of the query shown on the left are merge nodes, the graph on the right is a matching instance. However, it is not a merged instance as there is no complete triangle for instantiations of the three query nodes.

contains only one place, which must be $\iota_{G'}(n_1) = p_1$. Thus, we can add P to G' and after processing all remaining edges G' is a simple instance of Q and a sub-graph of H that contains p with $\iota_{G'}(n) = p$, the place that we started from. In the same way, we can start from an arbitrary valid path in H to construct a simple instance that contains that path.

□

For *pathway queries* that are not μ-circle-free, it is easy to find examples where matching instances exist that are not merged instances. Figure 5.6 shows such a query and a matching instance: a triangle of merge nodes is instantiated by a graph that does not contain a triangle. This demonstrates the locality of the definition of matching instances. It requires only that the neighborhood of each instantiating place must fulfill the conditions of the query. Larger structures such as the triangle in the example are not considered. Theorem 5.2.5 shows that the more local definition is sufficient as long as the query graph is μ-circle-free. From theorem 5.2.5 directly follows

Corollary 5.2.6. *A maximal matching instance for a μ-circle-free* pathway query Q *is also a maximal merged instance for Q and vice versa.*

In section 5.3 an algorithm for finding maximal matching instances will be introduced. This algorithm is more efficient than the naïve approach for finding merged instances as it does not require the enumeration of all simple instances. Instead, only partial simple

instances are enumerated, containing instantiations of all simple nodes of the query. Then, these partial instances are extended to maximal matching instances.

For the user, it is only important to understand the definition of merged instances, and that queries containing circles of merge nodes may produce something slightly different and should, if possible, be avoided.

In summary, we have developed two different notions of aggregation of instances. The first one can be understood intuitively and reflects the goal of the aggregation mechanism. The second notion, called matching instances, leads to an efficient algorithm, and we could show that in most cases the two notions are equivalent. Only when the *pathway query* contains circles of merge nodes, differences can arise.

5.2.3 XML Representation

The *pathway query language* is defined using the XML schema language. XML schemata describe the 'grammar' of an XML document. The main advantage of schemata as compared to the more conventional Document Type Definitions (DTDs) is the possibility to define data types. Thus, many errors in XML documents can be caught by automatic validation software. For instance, the choice and exact spelling of certain operators for *pathway queries* is defined in the XML schema. A user writing a *pathway query* can detect a mistake in his use of operators without trying to run a *pathway search* using validation tools. Another advantage of an XML schema-based language is that XML has become a standard for data interchange on the Internet. Thus, for instance a web service that executes *pathway queries* on a server with a network database can be provided easily.

The root node of a *pathway query* is `TheNet` which can contain `Subnet` and `Connection` nodes. A `Subnet` can either contain more `Subnet` and `Connection` nodes or a single `PathwayNode`. Therefore, in the easiest and most common case a *pathway query* is simply a collection of `PathwayNode`s and `Connection`s. A `PathwayNode` describes a single node in the query; it contains a *basic query* as described above and the `multiplicity` attribute. A `Connection` describes a path between two `Subnet`s, usually `PathwayNode`s. If there is a `Connection` between two `Subnet`s, each conforming instance must include a path between the instances of the two `Subnet`s that satisfies the three conditions specified in the `Connection`: The maximum path length is specified in the attribute `maxEdges`, the conditions on the places on the path are specified in an optional element `PlacesQuery` and the conditions on the transitions must satisfy the *basic query* specified by the element `TransitionQuery`.

We have also developed a simple cascading style sheet (CSS) to display *pathway queries* in a convenient manner. Figure 5.7 shows the visualization of a *pathway query* in a web browser.

The XML schema for *pathway queries* and the corresponding style sheet are listed completely in Appendix A.

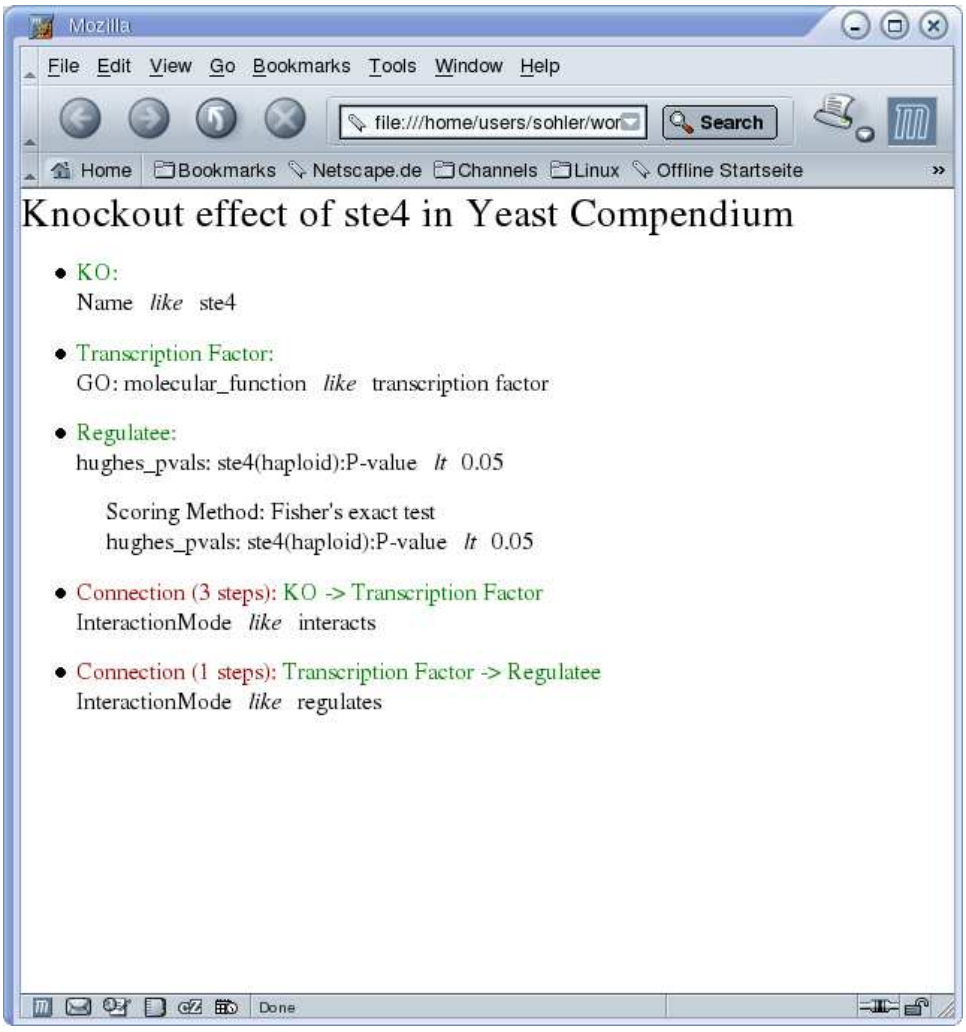

Figure 5.7: Visualization of a *pathway query* in a web browser using the cascading style sheet for *pathway queries*.

```
Graph buildInstanceGraph(Query q, Graph searchGraph) {
  for (Node n : q.nodes()) {
    instances = n.findInstances(searchGraph);
    instanceGraph.add(instances);
  }
  for (Connection c : q.connections()) {
    paths = c.findPaths(instanceGraph, searchGraph);
    instanceGraph.addEdges(paths);
  }
}

List buildPathways(Graph instanceGraph) {
  findSimpleNodeCliques(instanceGraph);
  return extendCliques();
}

List findPathways(Query q, Graph searchGraph) {
  inst = buildInstanceGraph(q, searchGraph);
  return buildPathways(inst);
}
```

Figure 5.8: Overview of the *pathway search* algorithm. Given a search graph and a *pathway query* it returns all subgraphs (instances) that match the query.

5.3 The pathway search algorithm

The search algorithm is basically an algorithm for the Subgraph-Isomorphism (SI) problem on labeled graphs (see section 1.4.2). This problem is known to be NP-complete in its decision version, and therefore no polynomial algorithm is known (Garey and Johnson, 1979). Subgraph-Isomorphism can be reduced elegantly to a Clique Search problem using a compatibility graph (also called association graph or product graph). The *pathway search* algorithm is also based on this approach. The main difference to the classical SI problem is that *pathway queries* are not simply subgraphs of the search graph:

- Edges in the query can correspond to paths in the search graph.
- Merge nodes in the query can correspond to more than one place in the search graph.

The *pathway search* algorithm is summarized in Figure 5.8. First we build the instance graph, which corresponds to the compatibility graph in the classical algorithm. A node in the instance graph corresponds to a place in the search graph taking the role defined by a query node (e.g. c-Jun in the role of the transcription factor in Figure 5.1). An edge is added between two nodes if the corresponding places can be in their respective roles in one instance. E.g. if there is a protein in the role of the kinase and another protein in the role

of the transcription factor, the kinase must be known to phosphorylate the transcription factor if that is required by the query.

A clique in the instance graph that contains nodes for all query nodes is then a simple instance of the query.

The construction of the instance graph $\Pi = (\hat{N}, \hat{E})$ can be formalized as follows: Π contains a node $\hat{n} = (n, p)$ if and only if

$$\xi(\nu(n), p) = \text{true}, \tag{5.2}$$

i.e. a place p can take the role of query node n if it satisfies the conditions made by the *basic query* $\nu(n)$. Let R_Q be a function that recovers the corresponding query node from a node in the instance graph and R_G a function that recovers the corresponding place in the search graph. All nodes in the instance graph corresponding to a query node n are denoted by $\Xi(n) = \{\hat{n} \in \hat{N} : \xi(\nu(R_Q(\hat{n})), p) = \text{true}\}$.

Then there is an edge $\hat{e}$ between two nodes $\hat{n}_1$ and $\hat{n}_2$ if one of the following two conditions is true:

1. There is no edge in the query between $R_Q(\hat{n}_1)$ and $R_Q(\hat{n}_2)$.
2. There is an edge $c = (R_Q(\hat{n}_1), R_Q(\hat{n}_2))$ and there exists a valid path for c starting at $R_G(\hat{n}_1)$ and ending at $R_G(\hat{n}_2)$.

To keep the instance graph small, we do not explicitly add an edge $(\hat{n}_1, \hat{n}_2)$ if there is no edge in the *pathway query* between $R_Q(\hat{n}_1)$ and $R_Q(\hat{n}_2)$. Still, $\hat{n}_1$ and $\hat{n}_2$ are compatible as they may appear together in a valid pathway instance. Therefore, we define a function θ to represent the compatibility of two nodes in the instance graph:

$$\theta(\hat{n}_1, \hat{n}_2) = \begin{cases} 1 & (\hat{n}_1, \hat{n}_2) \in \hat{E}, \\ & \text{or } (R_Q(\hat{n}_1), R_Q(\hat{n}_2)) \notin C, \\ & \text{or } R_Q(\hat{n}_1) = R_Q(\hat{n}_2) \text{ and } \mu(R_Q(\hat{n}_1)) = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

Using the compatibility function instead of explicit edges can save a lot of memory when queries have many unspecific nodes and only few edges. For instance, a linear query with ten nodes that have 100 possible instantiating proteins could have up to 500,000 edges most of which will be present as there are no conditions associated with them. Using the compatibility function reduces the number of edges to at most 50,000 and each one is subject to a condition.

For all pairs of places that have to be connected according to the query, all valid paths between these places must be found. An edge $\hat{e} = (\hat{n}_1, \hat{n}_2)$ in the instance graph should represent all valid paths for the corresponding edge in the query graph $R_Q(\hat{e})$, i.e. $R_G(\hat{e})$ should be the sub-graph that consists of all valid paths for $R_Q(\hat{e})$ between $R_G(\hat{n}_1)$ and $R_G(\hat{n}_2)$. In order to achieve this, we build a subgraph $S = (P_S, T_S, E_S)$ of the search graph for each edge $c = (n_1, n_2)$ of the query graph with $P_S = \{p \in P : \xi(\gamma_P(c), p) =$

true$\} \cup \Xi(n_1) \cup \Xi(n_2)$, $T_S = \{t \in T : \xi(\gamma_T(c), t) = \text{true}\}$ and E_S all edges between places and transitions in P_S and T_S. If, for instance, the `PlacesQuery` of the connection requires that places on the path are all in a certain biological process, the subgraph S will contain only such places. With the `TransitionQuery` it is possible to restrict the transitions on the path, e.g. to protein-protein interactions only. Given the two sets of nodes in the instance graph $\Xi(n_1)$ and $\Xi(n_2)$, we use breadth first searches (BFS) on S starting at each $p_1 \in \Xi(n_1)$ to find paths to places $p_2 \in \Xi(n_2)$. During the searches, the discovery time of each node is recorded. Then, the valid paths can be backtracked from each place in $\Xi(n_2)$ and the sub-graph containing all valid paths can be determined and annotated at each $\hat{e}$. Thus, $R_G(\hat{e})$ for $\hat{e} = (\hat{n}_1, \hat{n}_2)$ is the sub-graph of G that consists of all valid paths for the corresponding query edge $R_Q(\hat{e})$ between the places $R_G(\hat{n}_1)$ and $R_G(\hat{n}_2)$.

As a certain area around the places from $\Xi(n_1)$ must be explored, we always start the BFS at the smaller set of places. This can make the search much faster if the size of the sets is unbalanced. Suppose n_1 has hundreds of instances and n_2 has only a few instances. If we look for paths of length two between those sets and start with n_1, we have to visit all places that have a distance of two or less from those hundreds of places that are instances of n_1. If the search is started at the instances of n_2, much fewer places will be visited, assuming that the connectivity of nodes is similar around the two sets.

In order to avoid unnecessary path computations, we drop nodes from the instance graph as soon as they lack a required connection. Thus, we do not need to check other connections for that node and keep the graph as small as possible.

Building the pathway instances from the instance graph requires an extended clique search consisting of a standard clique search algorithm for all simple nodes and a subsequent extension step. The clique search uses a rather simple backtracking algorithm. A candidate clique $\mathcal{C}$ and a set $\mathcal{S}_i$ of all nodes compatible with $\mathcal{C}$ are maintained during the search. In the beginning $\mathcal{S}_0$ is set to $\hat{N}$. In each subsequent step i, $\mathcal{C}$ is expanded by a node $\hat{n} \in \mathcal{S}_{i-1}$ with $\mu(R_Q(\hat{n})) = 0$ and $\mathcal{S}_i$ is updated as $\mathcal{S}_i = \mathcal{S}_{i-1} \cap \{\hat{m} \in \mathcal{S}_{i-1} : \theta(\hat{m}, \hat{n}) = 1\}$. Omitting all nodes from S_{i-1} that are not compatible with $\hat{n}$, i.e. nodes $\hat{m}$ with $\theta(\hat{m}, \hat{n}) = 0$, ensures that S_i contains only nodes that are compatible with the candidate clique. When no node $\hat{n}$ with $\mu(R_Q(\hat{n})) = 0$ exists, all remaining nodes are added to $\mathcal{C}$. At this point, we have a candidate instance that contains instantiations for all simple nodes and candidates for the instantiation of all merge nodes. These candidates are compatible with the instantiations of all simple nodes, but we have not checked if all required connections between merge nodes are present. Therefore, we iterate through all places and check if they have at least one neighbor for each incident edge of the query. Places lacking a neighbor are dropped and the process is repeated until the instance does not change any more.

Effectively, we search only for cliques in the subgraph of Π containing all nodes $\hat{n}$ with $\mu(R_Q(\hat{n})) = 0$. All remaining compatible nodes are first added, then the connections between merge nodes are verified. When the resulting candidate instance contains nodes representing each query node, it is returned as a maximal matching instance.

In the last step of the algorithm, the instance is built from $\mathcal{C}$ by recovering the places and transitions in the search graph that correspond to nodes and edges in $\mathcal{C}$.

Theorem 5.3.1. *The sub-graphs identified by the* pathway search *algorithm represent maximal matching instances of the query, and all maximal matching instances are found by the algorithm.*

Proof. First, we have to validate if all conditions of definition 5.2.4 are met for an identified sub-graph H. We define the functions ι_H and κ_H as follows:

$$\iota_H(n) = \{p \in P : \exists\ \hat{n} \text{ with } R_Q(\hat{n}) = n \text{ and } R_G(\hat{n}) = p\}$$

$$\kappa_H(c) = \{P \in \text{Paths}(H) : \exists\ \hat{e} \text{ with } R_Q(\hat{e}) = c \text{ and } P \text{ is a valid path in } R_G(\hat{e})\}$$

From the construction, it is immediately clear that conditions 1 and 2 are true. Furthermore, conditions 3-5 are true because we require the instance to have instantiations of all query nodes, and two nodes in the instance graph corresponding to the same query node are never compatible. Conditions 6 and 7 are ensured by the extension step, when nodes that do not fulfill these conditions are eliminated. Condition 8 is clear from the construction of the functions ι_H and κ_H. Thus, H is a matching instance for the query. Furthermore, it is maximal: The path search algorithm ensures that for each edge in the query all shortest paths are found, and the extension algorithm starts from all nodes compatible with the partial instance representing the simple nodes and then only eliminates nodes that can not be part of this instance.

Finally, it is also clear that all maximal matching instances are found: A maximal matching instance is uniquely defined by the instantiation of all simple nodes. As the applied clique search algorithm is exhaustive, all possible combinations are discovered and are then extended to a maximal matching instance as described above. □

5.3.1 Hierarchical pathway queries

So far, only *pathway queries* were considered where a node n represents a single place; this is the case, if in the XML representation each `Subnet` element contains a `PathwayNode` element. But `Subnet` elements can contain complete *pathway queries* in form of further `Subnet` and `Connection` elements. In that case, the *pathway search* algorithm runs recursively, building the instances of the innermost `Subnet` elements first, and then finding paths between these instances according to the corresponding `Connection` elements. This recursive approach can speed up the search significantly, as the cliques that must be searched are much smaller than from the complete query. This fact can be made clearer by investigating the complexity of the different steps of the *pathway search* algorithm.

5.3.2 Complexity of the pathway search algorithm

As was already mentioned, the subgraph isomorphism problem is NP-hard, therefore our algorithm will have a running time that is exponential in the size of the query graph. NP-hardness can easily be shown by reducing the clique search decision problem to subgraph isomorphism. To answer the question if there is a clique of size k, we build a query graph

that is such a clique. Then there is a clique of size k in the search graph if and only if an instance of the query can be found. The decision problem is also in NP, as it is easy to verify in polynomial time if a given node mapping is indeed one-to-one and maps the query graph to an isomorphic subgraph of the search graph.

Nevertheless, we investigate the complexity of the algorithm in some more detail in order to find out more about possibly difficult queries and how to optimize those. Furthermore, we demonstrate the advantage of the *pathway search* algorithm over the naïve approach of enumerating all simple instances and merging these instances when the instantiation of all simple nodes is the same.

The *pathway search* algorithm has three parts, building the instance graph, enumerating partial pathway instances by a clique search algorithm, and finally extending the cliques to maximal matching instances.

For the analysis, let Q be a *pathway query* with k sub-queries Q_i having $s_i, 1 \leq i \leq k$ instances that have already been computed. Without loss of generality, let Q_1 to $Q_{k'}$ have `multiplicity` 0 and $Q_{k'+1}$ to Q_k `multiplicity` 1. For a connection c let c_{in} and c_{out} be the indices of the connected sub-queries.

As we assume that the instances of the sub-queries have already been computed, the algorithm only has to check the connections. For a query edge c between the i-th and the j-th sub-query, shortest paths between each instance of Q_i to each instance of Q_j have to be computed. For each instance $R_l, 1 \leq l \leq s_i$ of Q_i we build a subgraph S as described above and search for paths to all instances of Q_j. S can be computed with a breadth first search checking if places and transitions satisfy the restrictions of the `Connection`. The search can be aborted after the number of steps is as large as the maximal allowed length of paths for c. Within that graph, all shortest paths between R_l and all instances of Q_j have to be found. This is accomplished with a breadth first search starting from each R_l again, marking all visited nodes with their discovery time. From all instances of Q_j we then find the paths back to R_l by another breadth first search where in every step only nodes with a discovery time one less than the discovery time of the current node are included. The graph of all nodes visited in that last backtracking step is the graph of all shortest paths. The whole path computation is based on breadth first searches which can be bounded in total by the size of the search graph G, i.e. by $O(|P| + |T| + |E|)$. For all paths from one instance R_l to instances of Q_j the backtracking procedure has to be repeated s_j times. The total running time for paths between instances of Q_i and Q_j is therefore bounded by $O(s_i s_j(|P| + |T| + |E|))$. This path computation has to be done for all connections in the query, thus, the total running time for building the instance graph is $O(|C| \max_i\{s_i^2\}(|P| + |T| + |E|))$. We will summarize this result in

Lemma 5.3.2. *Building the instance graph in the* pathway search *algorithm takes* $O(|C| \max_i\{s_i^2\}(|P| + |T| + |E|))$ *time. More precisely, it can be bounded by* $O(\sum_{c \in C} s_{c_{in}} s_{c_{out}}(|P| + |T| + |E|))$.

When the instance graph is built, a clique search for the instances of Q_1 to $Q_{k'}$ has to be executed. The clique search algorithm is a branch and bound algorithm that computes all cliques of all sizes in its basic version without bounds. As we are only interested in

cliques of size k' and these are the maximum possible cliques (not taking into account nodes representing sub-queries with `multiplicity` 1), we can abort the search as soon as the candidate set does not contain instances for all remaining sub-queries. Nevertheless, in the worst case, all combinations of instances from the sub-queries are cliques and must be enumerated. At each expansion step for a clique, a set operation has to be performed that can be computed in $O(|\hat{N}|)$ in order to update the candidate set. This is done for each clique encountered during the search.

Let n_c be the number cliques in the instance graph. Then we can state the next Lemma:

Lemma 5.3.3. *The clique search during the* pathway search *algorithm runs in $O(|\hat{N}|n_c)$ time. This can be bounded by $O(|\hat{N}|\Pi_{i=1}^{k'} s_i)$ in the worst case.*

After the cliques of size k' have been identified, the instances are extended for merge nodes. During the clique search, nodes that are not compatible with all nodes of the clique have already been eliminated, but it has not been tested if the remaining nodes have all required connections, i.e. the edges between merge nodes have not been verified. Therefore, all nodes that do not have all required neighbors are eliminated. This can cause other nodes to lose required neighbors, so the process is repeated until the set of remaining nodes does not change any more.

Lemma 5.3.4. *For a μ-circle-free query, the number of iterations in the extension step is not greater than the length of the longest path in the query that contains only merge nodes.*

Proof. As all nodes left in the candidate instance are compatible to all simple nodes, we are only concerned with merge nodes and edges between merge nodes. If a node $\hat{n} = (n, p)$ of the candidate instance is eliminated, it can cause another node, say $\hat{n}_2$, to get eliminated as well, if there is an edge in the query between n and $R_Q(\hat{n}_2)$ and $\hat{n}$ is the only neighbor of $\hat{n}_2$ that instantiates the query node n in the candidate instance. Thus, there is a flow of information along the edges of the query graph. But the flow can not travel back and forth. When $\hat{n}_2$ is eliminated it can not cause another node $\hat{m}$ instantiating n to be eliminated. This could only happen if $\hat{m}$ was a neighbor of $\hat{n}_2$. In that case, the elimination of $\hat{n}$ would have no consequences for $\hat{n}_2$. Thus, the information can only travel in one direction, and in each iteration it is passed along one edge. If there are no circles, the number of iterations can therefore be no greater than the longest path of merge nodes. □

Checking all nodes of the candidate for their required neighbors takes $O(|\hat{N}| + |\hat{E}|)$ steps. As shown in Lemma 5.3.4, the number of iterations is at most as high as the length of the longest path of merge nodes in the query graph. This can be bounded by the total number of merge nodes $|N_m|$. The extension has to be performed for all candidate instances. Therefore, we get the following Lemma for the running time of the extension step:

Lemma 5.3.5. *The extension step of the* pathway search *algorithm requires $O(n_{cand}|N_m|(|\hat{N}| + |\hat{E}|)$ steps, where n_{cand} denotes the number of candidate instances, i.e. the number of maximal cliques of simple nodes.*

The last part missing in the complexity analysis is the computation of instances for non-recursive `Subnet` elements. These elements consist only of a single `PathwayNode`, therefore the computation of instances involves only checking a *basic query* for all places in the search graph. Thus, the running time is $O(|P|)$, assuming that a *basic query* can be checked in $O(1)$. From this and the preceding lemmas directly follows

Theorem 5.3.6. *Given a query Q with a set of sub-queries R and the definitions from above, the running time $T(Q)$ of the* pathway search *algorithm can be bounded by the following recursive formula:*

$$T(Q) = O\left(\sum_{c\in C} s_{c_{in}} s_{c_{out}}(|P| + |T| + |E|) + |\hat{N}|\Pi_{i=1}^{k'} s_i + n_{cand}|N_m|(|\hat{N}| + |\hat{E}|)\right) + \sum_{Q_r\in R} T(Q_r) \tag{5.4}$$

If the set R is empty, Q consists only of a single `PathwayNode` *element and its instances, i.e. instantiating places in the search graph, can be computed in $O(|P|)$.*

This estimation can be used to formulate a strategy for query optimization. The running time to compute the instance graphs depends mainly on the number of instances of the sub-queries. This number can be minimized by careful query formulation. Additionally, the number of nodes that can be eliminated from the instance graph after each path computation depends on the order of the path computations. One simple optimization that appears promising is to sort the query edges such that connections between subnets with few instances are processed first. This is done in the *pathway search* algorithm and does not require a different query formulation.

In the clique search, nothing can be done about the number of maximal cliques, as this corresponds to the number of valid pathways, which of course have to be enumerated eventually. But the number of non-maximal cliques that have to be enumerated depends strongly on the query formulation. It grows with the number of simple nodes in the query k' and their instances s_i, as can be seen from Lemma 5.3.3. Thus, a possible strategy to minimize this number is to formulate a query such that it contains only few sub-queries, and these should result in as few instances as possible. The latter condition is also important for the running time of the path computations, as already mentioned. As a rule of thumb, queries with few sub-queries with few instances can be achieved by splitting a query into sub-queries that are as densely connected as possible. As an example, consider a linear query consisting of five nodes $n_1 \dots n_5$ where n_i is connected to n_{i+1} for $1 \leq i \leq 4$. All triples of nodes representing n_1, n_3 and n_5 are valid cliques as there are no connections and, thus, no restrictions between them, and they might have to be enumerated during the clique search algorithm. We try to avoid this by our branch and bound method, but in the worst case all cliques are enumerated. If the query is restructured into sub-queries consisting of pairs of nodes, or adding one node in each recursion layer, no such unrestricted tuples of nodes will be considered.

A second conclusion can be drawn from the run time analysis of the *pathway search* algorithm. The aggregation strategy chosen here is much more efficient than merging

instances after enumerating them completely. The latter approach increases the running time of the clique search immensely, as bigger cliques have to be found. The worst-case running time would be $\Pi_{i=1}^{k} s_i$, as all query nodes would be treated as simple nodes first. This increases the running time by a factor of $\Pi_{i=k'+1}^{k} s_i$ as compared to an additive increase of $n_{\text{cand}}|N_m|(|\hat{N}| + |\hat{E}|)$ for the *pathway search* algorithm. These observations only deal with the worst-case complexity, but the practical improvement is immense as well. Usually, merge nodes are used when there are many possible instantiations of a query node that should be summarized. Especially when multiple merge nodes are connected, the number of resulting simple instances instances can be enormous. Avoiding the enumeration of these instances is crucial and furthermore allows a better presentation of the results to the user and meaningful statistical scoring methods as will be demonstrated in the next section.

5.3.3 Summary of the *pathway search* algorithm

The basic idea of the *pathway search* algorithm is rather simple and follows standard approaches. A compatibility graph is built from the graph template and the biological network. Then a clique search retrieves sub-graphs matching the template. The aggregation feature, which first of all serves the purpose of increasing the expressiveness of the *pathway query language*, gives rise to an improvement in the running time for queries containing merge nodes. The possibility of aggregating instances allows to formulate queries that represent meaningful biological contexts, and the *pathway search* algorithm can efficiently enumerate the instances of such contexts.

5.4 Scoring pathway queries

There are two reasons why it is necessary to assign scores to the instances of a *pathway query*. Sometimes a *pathway query* results in many instances and manual inspection of all results becomes infeasible. In that case, a ranking is needed, so the user can work down a sorted list, starting with the best scoring (usually the most significant) results.

But even if only few instances are available, it is often useful to assign a significance score to the instances to help the user assessing the results.

Scoring methods can be indicated in the `PathwayNode` or `Connection` elements of a *pathway query*. In every instance, these partial scores are computed and then combined to a total score. Combining scores requires certain properties of the scores. The *pathway query language* and the implemented scoring mechanism allow for scores that are additive or scores that are valid p-values.

5.4.1 Map scores

The easiest way to compute a score for a single node in a pathway instance is to refer to a *data map*. Every *data map* that provides p-values can be used as a score for a part of a pathway instance. For instance, a p-value for differential expression available in a *data*

map could be a good score for a gene if this p-value is not used in the query to select the gene. On the other hand, it does not make sense to select genes according to their p-value and then score the pathway instances using the same p-value on these genes. For instance assume a query like the one depicted in Figure 5.1: If the regulated genes are selected according to their significance of differential expression, using the same p-value to score the instances will result in overly significant p-values for the instances. Using the expression p-value to score the transcription factor on the other hand is a valid method if the transcription factor has been selected only because of its function and its connection to regulated genes.

5.4.2 Enrichment scores

Using enrichment or over-representation analysis, we can assign scores to instances of a common kind of pathway queries where we have two nodes n_1 and n_2 with $\mu(n_2) = 1$ and an edge $c = (n_1, n_2)$. The question we would like to answer is: Given a pathway instance with places p_1 and $p_{2_i}, 1 \leq i \leq k$, how likely is it to find a pathway instance with at least k instances of n_2? Obviously this depends on the number of places in the search graph that satisfy the *basic query* in n_2 and on the number of edges of p_1. Thus, the question can be formulated more precisely: Are the instances of n_2 over-represented in the set of neighbors of p_1? One way to compute a p-value for the over-representation in such a set-up is Fisher's Exact Test, which uses the hypergeometric distribution.

Let N be the set of places for which the *basic query* $\nu(n_2)$ is applicable and n the set of neighbors of p_1 in G. A *pathway query* instance I including k places as instances of n_2 is assigned the p-value

$$p_{\text{FET}}(I) = \sum_{i=k}^{min\{|n|,|\Xi(n_2)|\}} \frac{\binom{|\Xi(n_2)|}{i}\binom{|N|-|\Xi(n_2)|}{|n|-i}}{\binom{|N|}{|n|}} \tag{5.5}$$

or the corresponding FET score

$$S_{\text{FET}}(I) = -\log_{10} p_{\text{FET}}(I). \tag{5.6}$$

Another enrichment score is based on Wilcoxon rank scores. Let us again assume a query with two nodes n_1 and n_2 with $\mu(n_2) = 1$ and an edge $c = (n_1, n_2)$. If there is a function $f : P \to \mathcal{X}$ (implemented by a *data map*) where $\mathcal{X}$ is a completely ordered set, e.g. $\mathcal{X} = \mathcal{R}$, we can compute the ranks of all instances $p_{2_i}, 1 \leq i \leq k$ of n_2 compared to all places in the graph with respect to the function f.

Using these ranks, we can apply the Wilcoxon-Mann-Whitney test (WMWT) as described in 1.4.1. Let W_I denote the Wilcoxon score (the sum of the ranks) of the current instance I, then the p-value p_{rank} is defined as

$$p_{\text{rank}}(I) = 2\min\{P(W > W_I), P(W < W_I)\} \tag{5.7}$$

and the rank score is

$$S_{\text{rank}}(I) = -\log_{10} p_{\text{rank}}. \tag{5.8}$$

The WMWT tests the null hypothesis that the values from places representing query node n_2 have the same distribution as the values from all other places in the search graph. In addition, when the null hypothesis is rejected, the magnitude of the Wilcoxon score tells us if the tested values are generally higher or lower than the rest. We call it an enrichment score because the null hypothesis is rejected when the tested places are enriched in high or low values.

5.4.3 Scoring transcription factors and kinases

The enrichment scores S_{FET} and S_{rank} can be used to find regulators that are relevant for a set of expression data where the regulators themselves do not exhibit significant expression values. Transcriptional regulators like transcription factors, kinases and signaling molecules are not necessarily regulated transcriptionally. Especially in higher organisms, other modes of regulation, e.g. by phosphorylation or translocation, and combinatorial interaction with other regulators, are more important. Thus, the activity of a regulator cannot be estimated by its expression level alone, but must include the expression levels of its potential regulatory targets.

In order to compute the FET score for the relevance of a transcription factor or any other regulator F as described above, we need the number of potential target genes $t(F)$. Let $r(F)$ be the number of significantly regulated genes in that set. Then S_{FET} can be computed as

$$S_{\text{FET}}(F) = -\log_{10} \sum_{k=r(F)}^{\min\{R,t(F)\}} \left(\frac{\binom{R}{k}\binom{N-R}{t(F)-k}}{\binom{N}{t(F)}} \right), \tag{5.9}$$

where N is the total number of measured genes and R the number of genes that are significantly regulated.

The rank score $S_{\text{rank}}(F)$ can be computed using the gene expression levels of all potential targets of F and compare them to the expression levels of all other measured genes using the rank sum test as described in section 5.4.2.

When the expression data reflect a change of expression (e.g. ratios or p-values for differential expression), we call the scores $S_{\text{FET}}(F)$ and $S_{\text{rank}}(F)$ activity scores of F, although they actually represent a *change* of activity. These scores are high when a significantly large number of the regulator's potential targets is differentially expressed in the experiment under consideration.

Of course, it is crucial to have high quality sets of potential targets. To extract these from the network, we use *pathway queries*, but first the knowledge about the targets has to be encoded in the network. Such networks can be built from databases like TRANSFAC (Wingender et al., 2000); we have also used a network containing co-occurrence edges and binding site predictions to compute potential targets (Gebauer et al., 2005). In another application we computed binding site predictions incorporating information about the conservation of non-coding sequences using multiple alignments between different species.

The pathway queries that extract the necessary information on the targets from the network are particularly simple. For transcription factors, there are only two query nodes,

the first representing the transcription factor itself, the second representing the targets. The connection must not be longer than a single transition which should be labeled 'regulates'. We run the query with no restriction on the expression data of the targets to select all potential targets. Then, for each expression measurement, we select all targets that have a p-value of less than 0.01.

For kinases, we use a *pathway query* that is a little more complicated. We need the first node to represent the kinase which should also be at least slightly regulated, so we require it to have a p-value of less than 0.1. It must be connected via at most one more kinase to a transcription factor. The transcription factor is again connected to regulated target genes. The reason for requiring the kinase to be differentially regulated is that otherwise too many instances are found that differ only in the kinase. One way to select kinases that are especially interesting is to require differential expression. Both *pathway queries* are visualized in Figure 5.9.

5.4.4 Power of enrichment scores

In statistics, the power of a test is defined as the probability that the null hypothesis is rejected if indeed an alternative hypothesis is true. In this section, some related properties of the proposed enrichment scores are investigated.

In order to find out if the described scoring methods are applicable to biological data containing errors and noise, some experiments on simulated data were performed. In particular, it was intended to get a rough quantification of the method's robustness with respect to the amount of errors in the prediction of a regulator's targets and the difference in expression between the regulated genes and non-regulated genes. Therefore, a model for expression data and the detection of target genes was defined as follows, and random data were generated according to that model. These simulations provide us with data, of which we know that an alternative hypothesis is true. Thus, we can estimate, how sensitively the enrichment scores detect such an alternative hypothesis.

Let us assume we have gene expression data for n genes and we are interested in the activity of a regulator R. The regulator R has k targets, l of which are indeed regulated in the data. There are another m genes that are also regulated but are not targets of the regulator under consideration. Our method for determining targets of a regulator (e.g. predicting transcription factor binding sites) is supposed to miss a target with probability p_{FN} and erroneously classify a random gene as a target with probability p_{FP}. Now we need to define a model for the gene expression data for regulated and non-regulated genes. We simply assume that the expression levels of unregulated genes follow a standard normal distribution and the regulated genes are all up-regulated and can be modeled by a normal distribution with mean s and standard deviation 1. Thus, we have two classes of gene expression data, corresponding to regulated and unregulated genes and separated by s standard deviations. The parameters of the model are summarized in Table 5.3.

Figure 5.10 shows a typical histogram of simulated expression data generated with parameters $n = 10000$, $m = 250$, $k = 100$, $l = 25$ and $s = 4$. Although the distributions of targets and non-targets are clearly different, this observation is actually based on only

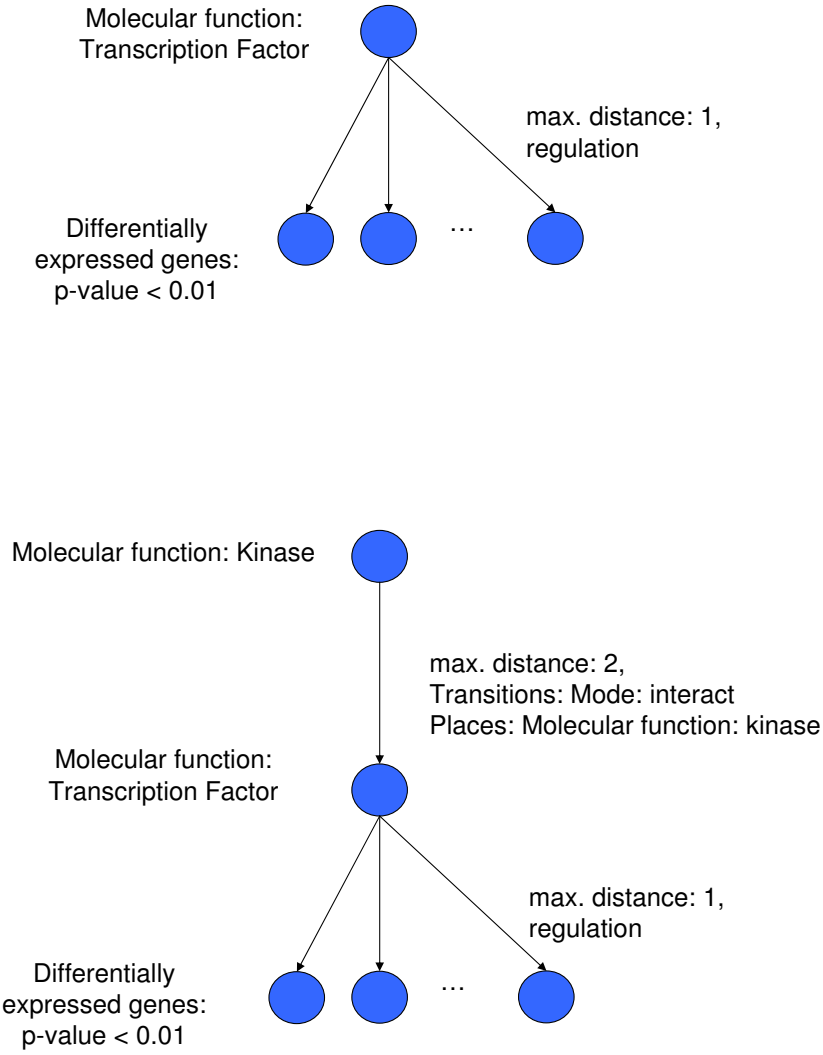

Figure 5.9: Pathway queries used to find relevant transcription factors (top) and regulated kinases (bottom). With these simple network templates, regulated genes can be put into context with their regulating transcription factors and kinases.

Parameter	Value	Description
n	10000	Total number of genes
m	250	Number of regulated genes that are not targets
k	100	Total number of target genes
l	25	Number of regulated target genes
p_{FP}	0.01	Probability for false positive targets
p_{FN}	0.3	Probability for false negative targets
s	4	Separation of expression means between regulated and unregulated genes
t	1.5	Threshold for detection of regulated genes (for FET score only)

Table 5.3: Parameters of the model for gene expression data and target finding. Values indicate standard parameter values that were used in experiments where nothing else was specified.

few data points, since there are only 100 targets with only 25 of these regulated, i.e. being in the class of genes with higher expression values. Furthermore, in this histogram the possibility of false negative and false positive targets is not taken into account. The arising question is if the scores are powerful enough to recognize this difference of distributions even in the presence of false positive and false negative targets.

Scores based on rank test

For random data generated with the same parameters as in Figure 5.10 and target assignments with $p_{FP} = 0.01$ and $p_{FN} = 0.3$, the average score S_{rank} from 50 test runs was 4.1 (corresponding to a p-value of $10^{-4.1}$), the histogram of the p-values is shown in Figure 5.11.

In order to estimate how the score depends on the separation of the two classes of expression values and on the specificity of target detection, the parameters s and p_{FP} were systematically varied. The results of these variations are shown in Figure 5.12. The shape of the contour lines suggests that there is an approximately logarithmic dependency between the false positive rate of the target detection and the resulting activity scores, while the dependency between separation and scores is linear. A saturation occurs at scores around eight. Furthermore, it is clear from the almost rectangular shape of the contour lines that both a good separation of expression values and a low number of false positives is required to get good scores. While a sufficiently good separation of expression values can be expected in real biological data, the identification of target genes can be a problem. If it is necessary to infer target genes using binding site prediction in genomic sequences, a higher false positive rate is to be expected for typical approaches using position-specific weight matrices (Stormo, 2000). In that case we will use different methods to minimize the number of false positives in the applications that will be discussed in chapter 6.

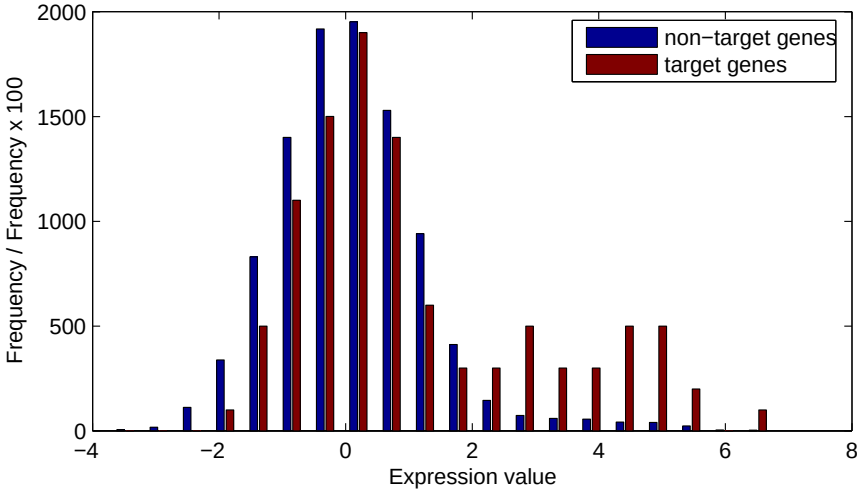

Figure 5.10: Histogram of simulated expression data. Blue bars indicate the frequencies of expression values of non-targets while red bars represent frequencies of expression values of target genes multiplied by 100 (in order to have the same total number). There are 10000 genes, 100 of which are targets. 25 of the targets are regulated and 250 of the non-targets. The separation of the regulated and unregulated genes is four standard deviations.

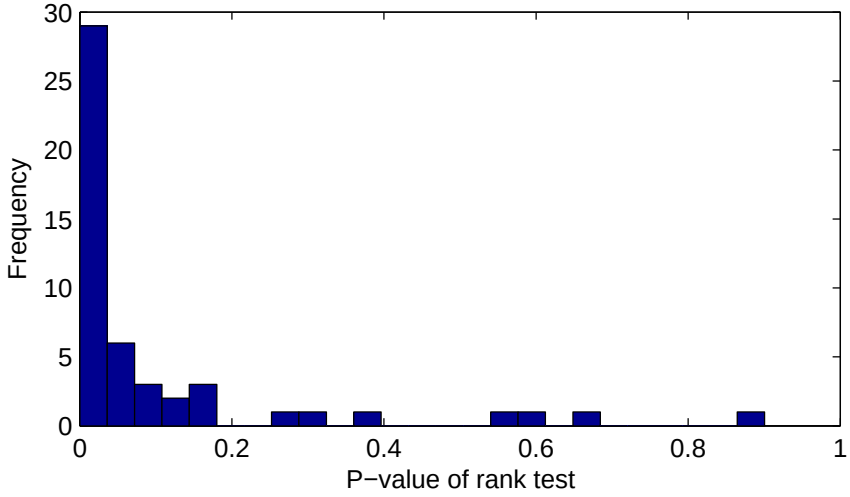

Figure 5.11: Histogram of rank p-values computed on 50 sets of simulated expression data with the same parameters as in Figure 5.10 and $p_{FP} = 0.01$ and $p_{FN} = 0.3$. For most of the data sets, the enrichment can be detected with highly significant p-values.

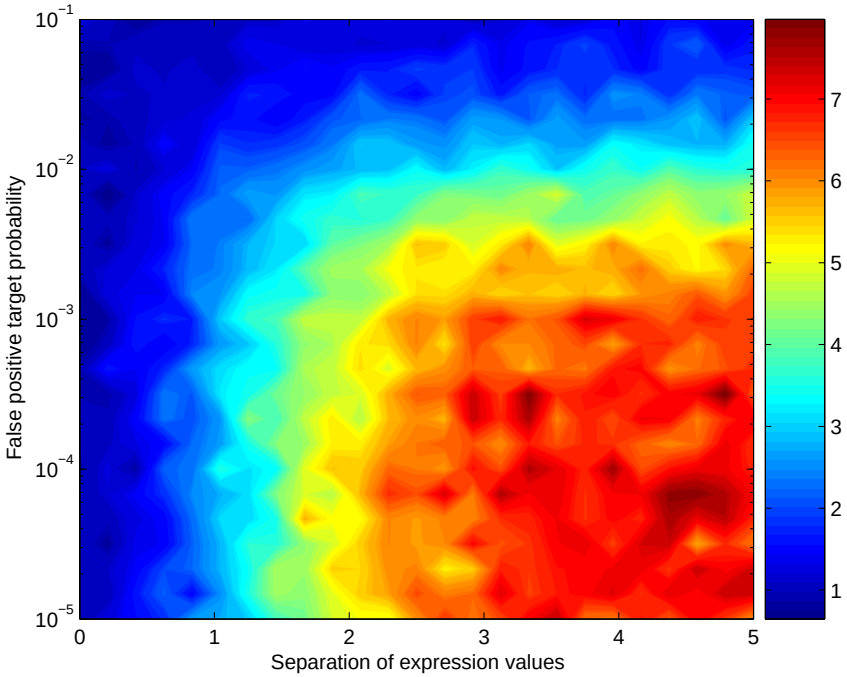

Figure 5.12: Rank activity scores on simulated data over a range of parameters s and p_{FP} (separation of expression values between regulated and unregulated genes and probability for false positive targets respectively).

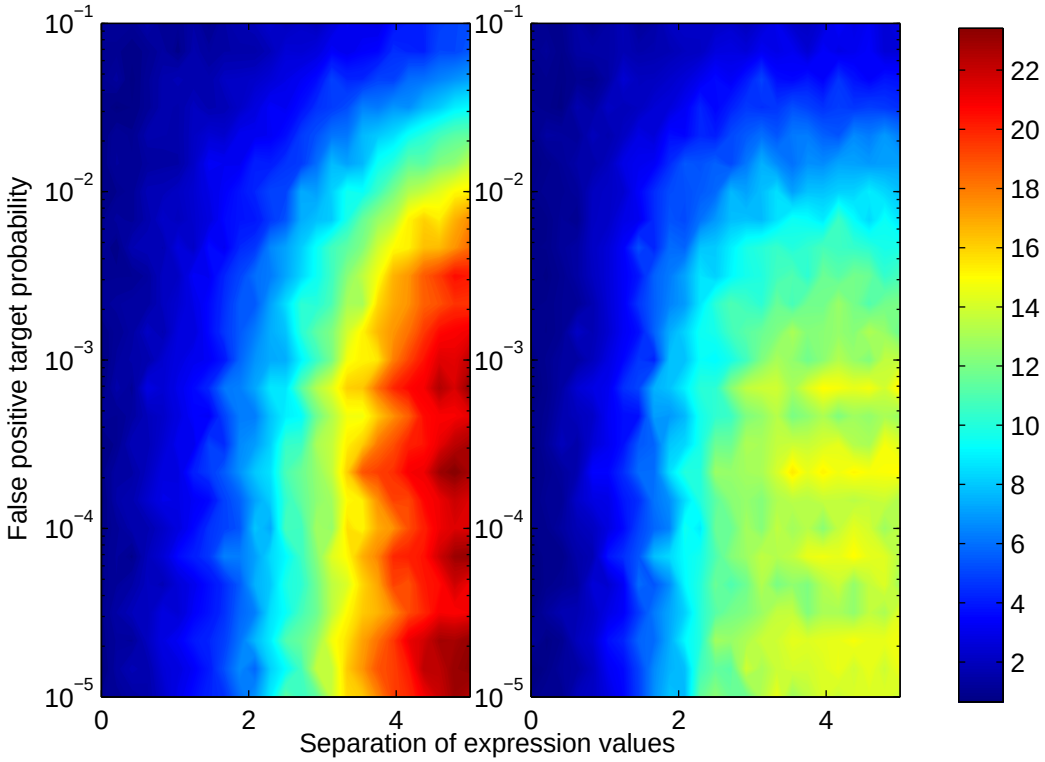

Figure 5.13: FET activity scores on simulated data over a range of parameters s and p_{FP} (separation of expression values between regulated and unregulated genes and probability for false positive targets, respectively). Threshold t for defining the set of regulated genes in FET was set to $s/2$ (left) or to 1.5 (right).

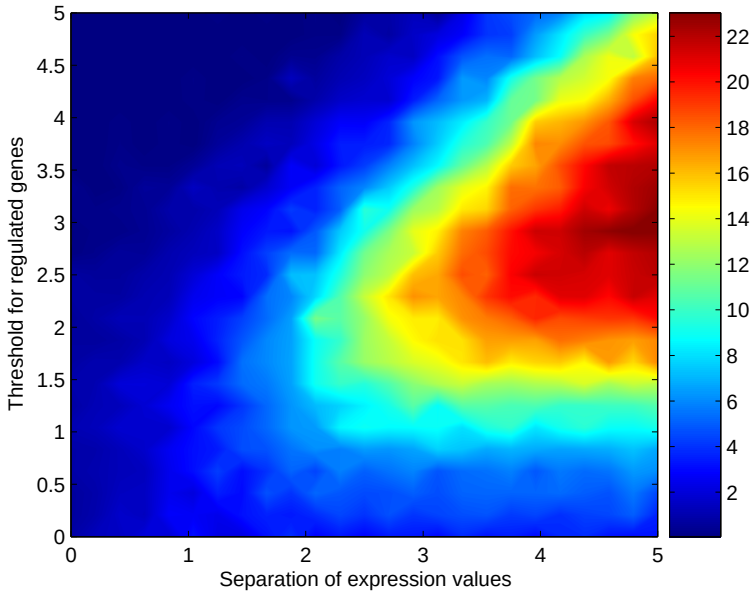

Figure 5.14: FET activity scores on simulated data over a range of parameters s and t (separation of expression values between regulated and unregulated genes and threshold for the definition of regulated genes, respectively).

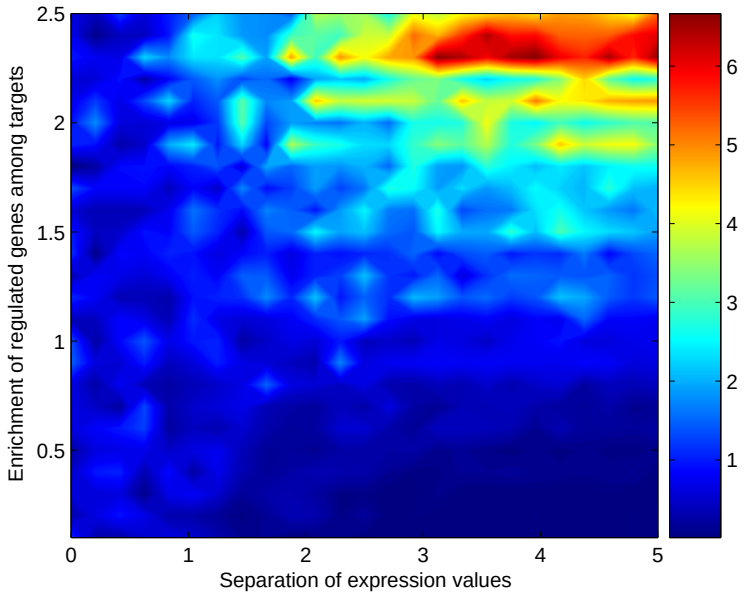

Figure 5.15: FET activity scores on simulated data over a range of parameters s and l (separation of expression values between regulated and unregulated genes and enrichment of regulated genes among targets defined as $\frac{l}{k}/\frac{m}{n}$, respectively).

Scores based on Fisher's exact test

For similar analyses for the score S_{FET}, a threshold t has to be defined above which an expression value is considered significant. A good threshold should be the midpoint between the means of the two normal distributions used to model the expression values. Unfortunately, the separation of regulated genes from the unregulated genes is not known in real applications. In that case one would usually choose a fixed value, maybe one or two standard deviations from the mean. The resulting scores for a fixed threshold of $t = 1.5$ and a variable threshold $t = s/2$ are shown in Figure 5.13. Using the variable threshold, much more significant scores can be attained at high separation values. With the more realistic fixed threshold, the results look very similar to what was observed for the rank score. In order to get a more complete overview of the effects of changing the threshold, another contour plot was created with $p_{FP} = 0.001$ and varying threshold (Figure 5.14). This plot clearly suggests that the choice of the threshold t is important, but with greater separation between regulated and unregulated genes, the range of values that yield significant scores becomes quite big. Furthermore, a threshold of 1.5 or 2 seems to produce good results for the complete range of separation values.

Of course, other parameters also play an important role, but they do not change the shape of the contours in the previous figures. Let r_T be the ratio of regulated to unregulated genes among the targets and r_N the same ratio among the non-targets. The enrichment of regulated genes among the target genes can then be defined as the ratio $\frac{r_T}{r_N}$. The null hypothesis for the rank test is the equality of medians, and the null hypothesis for Fisher's exact test is that $r_T \leq r_N$. Therefore, it is crucial that there is an enrichment in the target genes. Otherwise the tests cannot find significant differences. Making r_T and r_N more similar by changing the parameters for the number of regulated genes m and l accordingly will result in generally lower scores, making the difference larger will result in higher scores. The dependency of the FET score from the enrichment is illustrated in Figure 5.15 with parameters $n = 10000$, $m = 1000$, $k = 100$, $p_{FN} = 0.3$, $p_{FP} = 0.001$, $t = 2$, and various values of l and s. Clearly, if the enrichment is less than 2-fold, almost no significant scores can be found, even if the expression values are well separated.

Increasing the number of targets l also produces more significant scores, since with more samples even a small enrichment can be detected confidently. Finally, a higher false negative rate for target identification also decreases the resulting scores, as this basically changes only the ratio of regulated to unregulated targets (dividing the number of regulated targets l by 2 has almost the same effect as doubling the probability for false negative targets p_{FN}).

In summary, it seems that the sensitivity of the proposed scores is sufficient for real biological data sets; the rank score seems to be somewhat less sensitive than the FET score. Naturally, there has to be an enrichment of regulated genes among the targets, otherwise nothing can be detected. Furthermore, the specificity of the target detection must be quite high which should be taken into account when conducting a study on biological data.

5.4.5 Combining scores

Combination of scores representing p-values can be performed using a χ^2 test. For each `Subnet` and `Connection` of a *pathway query* that contains a `Scoring` element of p-value type, the indicated method is used to compute a p-value. If these p-values $P_1, \ldots, P_n$ are independent and all originate from their respective null distributions, then $\sum_{i=1}^{n} -2 \log P_i$ follows a χ^2 distribution with $2n$ degrees of freedom (Fisher, 1932). This method is used in the Significant Area Search algorithm as well, and is generally used in ToPNet for combining p-values (see chapter 4 and Hanisch (2004)). Of course this method only works if all single scores represent valid p-values. But in that case it is possible to create very powerful and flexible scores for *pathway queries*. For instance, it is possible to score parts of the query using p-values from a *data map* and other parts using one of the described enrichment scores.

5.4.6 Implementing additional scoring methods

Through the modular architecture of ToPNet it is easy to implement further statistical scoring methods. For instance, one could implement goodness-of-fit tests like the Kolmogorov-Smirnov test, the t-test or the chi-squared test in order to compare the expression values of two gene sets. Other interesting approaches for scoring methods could be based on correlation coefficients between genes over a set of microarray experiments.

5.4.7 Specifying scoring methods in the pathway query

The XML element that contains scoring information is called `Scoring`, it can contain either a `P-Value` or an `AdditiveScore` element.

A p-value score can be a map score or one of the enrichment scores. As an example, an FET score can be annotated at a `PathwayNode` n with `multiplicity` 1. It is defined by a *basic query*, which specifies the set of places that should be compared with the instances of the n. Typical examples are *basic queries* selecting a GO class or all genes with a maximum p-value. Besides the two sets whose overlap is to be evaluated, a superset has to be defined for FET. This superset is either the set of all places in the graph or the set of all places that are found in any instance of n. The latter can be specified by setting the attribute `relativeToResult` to true. Figure 5.16 shows an example of a scoring element that uses FET to compare the set of instances of a `PathwayNode` to the set of genes that are differentially expressed in an expression data set on osteoarthritis.

5.5 Association rule mining in pathway instances

If a certain *pathway query* is run on many different data sets, it can be useful to extract rules from the resulting instances. Association rule mining is a well known problem in data mining. One of the most commonly used algorithms for rule mining is the *Apriori* algorithm (Agrawal et al., 1993). Given many so-called transactions, which are simply sets

```
<Scoring>
  <P−Value>
    <FETScore relativeToResult="false">
      <Query>
        <BasicQuery>
          <MapName>AffyOA: p−values</MapName>
          <Operator>lt</Operator>
          <Value>1e−5</Value>
        </BasicQuery>
      </Query>
    </FETScore>
  </P−Value>
</Scoring>
```

Figure 5.16: Example of a `Scoring` element. This XML code describes the scoring function at a node in a *pathway query*. Here, FET is used to compare the instances of the node to all differentially expressed genes, which are defined by via a p-value in a specific *data map*. The attribute `relativeToResult` is used to define the base set, in which the comparison is performed. It can be either all places in the underlying network, or all places that instantiate the query node that is being scored in any of the instances found.

of arbitrary items, the Apriori algorithm finds item sets that appear together frequently and reports them as rules such as $A, B \Rightarrow C$, i.e. if A and B appear in a transaction, then C also appears in that transaction (in most cases). A, B is called the antecedent of the rule and C is called the consequent. Rules have two important properties, support and confidence. The support of an item set is the fraction of transactions in which the items appear. The support of a rule is the support of the antecedent. The confidence of a rule is the support of all items in the rule divided by the support of the antecedent, i.e. the fraction of all transactions containing the antecedent where the rule holds.

A possible application of rule mining in *pathway queries* is the identification of special relationships between regulators and their targets. Let us assume a query to find transcription factors and their targets as in Figure 5.9. When the *pathway search* algorithm is executed with that query on many different sets of expression data, it results in a large set of pathway instances each of which contains one transcription factor and several regulated targets. Using association rule mining, it is possible to find genes that are only regulated by one transcription factor, or transcription factors that always regulate a certain gene.

Association rule mining has been performed on expression data before, resulting in rules like: "If gene A is differentially expressed then gene B is in most cases differentially expressed as well" (Creighton and Hanash, 2003). Recently, Georgii et al. (2005) have proposed quantitative association rule mining on expression data. Combining this new technique with *pathway queries* presents an interesting extension to our analyses as the output of the *pathway search* algorithm also contains partly numerical data: the scores for instances, the number of places representing an aggregation node, and of course the expres-

sion values on the instances. Rules on these data could possibly describe very interesting biological facts. For instance, if we look at transcription factors and their regulated target genes, such a rule could state that if the expression values of two transcription factors are high, then the set of their regulated targets is usually large, i.e. the set of genes that have binding sites for both of these factors and that are differentially expressed between two conditions is large. It remains to be shown that such rules can indeed be supported by real biological data.

5.6 Visualization of pathway instances

Pathway queries often have a more or less linear structure. This should be used for the layout of the resulting instances. When formulating a *pathway query* a visualization attribute called `layer` can be defined by the user for every `PathwayNode` and `Connection`. During the *pathway search* a *data map* is constructed that assigns to each node of a pathway instance the correct layer. For the layout itself, the display window is divided into rectangles corresponding to the layers and all nodes are placed within their layer; then a spring embedding algorithm is performed that optimizes the positions of the vertices within the layers. An example output of the pathway layout algorithms is depicted in Figure 5.17. Using this layout mechanism it is easy for the user to identify proteins that represent certain important parts of the query even if the instance is large. For instance, the receptors and transcription factors in Figure 5.17 can be easily spotted, while a normal spring embedding layout without the additional layer information would put those proteins somewhere in the middle of the bulk of regulated genes.

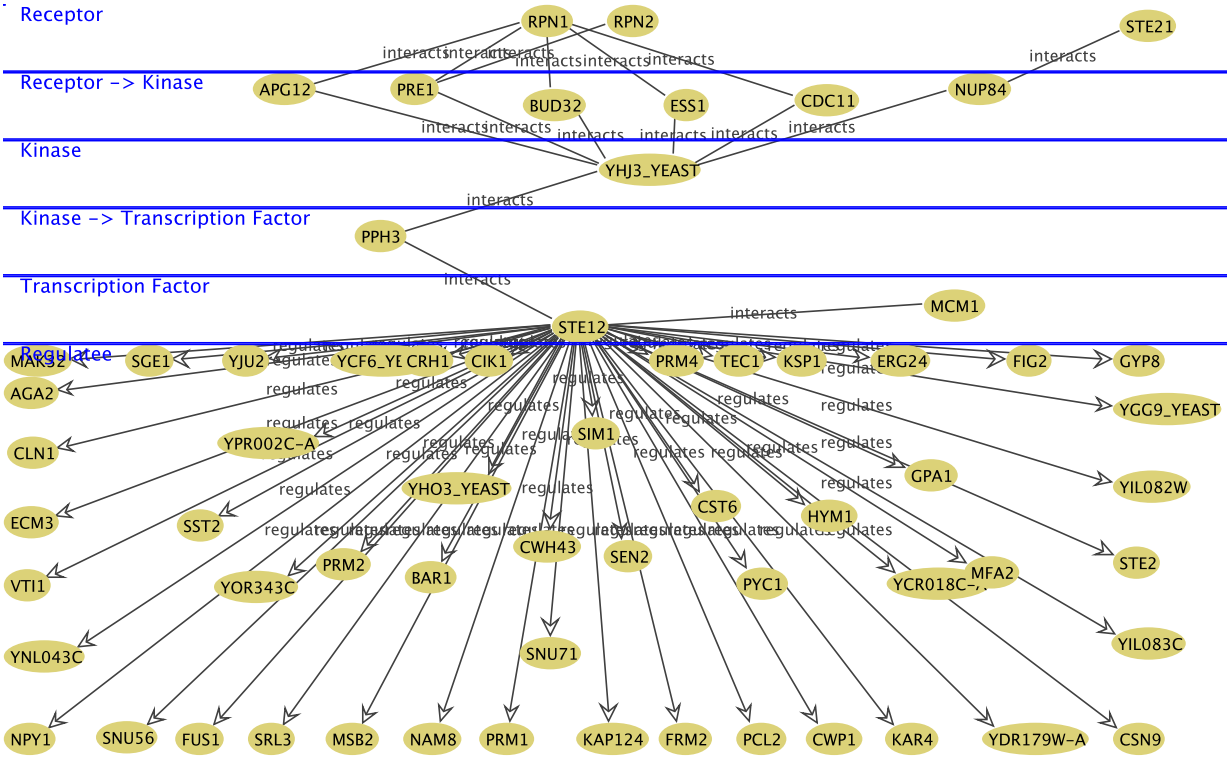

Figure 5.17: Visualization of a pathway instance with four layers. The corresponding *pathway query* has five nodes (receptors, kinase, transcription factor, interacting factor and regulatees) and four connections. The connections from the receptor to the kinase and from the kinase to the transcription factor can have intermediate nodes, therefore an additional layer was introduced for these connections.

Chapter 6

Applications

This chapter deals with several application of network-based analysis methods for gene expression data. First, some analyses of a publicly available expression data set for yeast are described. This data set serves as a kind of benchmark because it has been analyzed by many other groups, and, more importantly, much of the relevant biology is known. Therefore, results can be validated by checking if they conform to the biological knowledge.

Next, a dataset from Drosophila is discussed. Drosophila is also a well studied model-organism, but the regulatory processes are much more complex than in yeast. Furthermore, the data available on transcription factor binding sites is not nearly as comprehensive as it is for yeast.

Finally, some applications of network-based methods on osteoarthritis data are discussed. The corresponding data were collected in a BMBF-funded research project with partners from other universities and industry.

Thus, three datasets with increasing complexity and uncertainty are considered. While our goal for the first dataset is to recover known facts from the biology of yeast, in the second and especially in the third dataset, the analysis is directed more towards the finding of new hypotheses, which cannot be validated completely by existing biological knowledge. Still, the plausibility of the results will be checked by some literature research.

6.1 Analysis of yeast compendium data

To test our network-based methods on a well-studied system, we selected the Rosetta yeast compendium dataset (Hughes et al., 2000). It provides expression ratios as well as p-values for differential expression for 300 measurements, 287 of which are knock-out mutations while the last 13 are treatments with chemical compounds. The p-value for a gene in a certain experiment quantifies the probability that the observed ratio of expression is due to random fluctuations, although the gene is not really differentially expressed. These fluctuations may arise in the gene expression (biological variation) or in the measurement process (technical variation). The p-value is computed from an error model that was calibrated using 63 control experiments where expression values for two wild type samples

were compared.

We use a combination of two different networks that contain our prior knowledge: The yeast subset of the DIP network of protein interactions (Xenarios et al., 2000) and a network containing DNA binding information constructed from the genome-wide location analysis of Lee et al. (2002), adding an edge between a transcription factor and a target gene if the authors reported a p-value of less than 0.001 for that interaction.

Functional annotations are extracted from gene ontology (The Gene Ontology Consortium, 2001). In our examples, we only need annotations for kinases and transcription factors. The latter could have been extracted from the location data as well; we used gene ontology annotations for simplicity and consistency.

Pathway information from the KEGG database (Kanehisa, 1996) is used to assign to each gene the set of pathways it participates in.

6.1.1 Enrichment analysis in KEGG pathways

The first application is the detection of significant changes within pathways for each experiment. This is accomplished using Fisher's exact test (FET) on the p-values provided by Hughes et al. (2000) and the Wilcoxon-Mann-Whitney test (WMWT) on the corresponding expression ratios as described in section 1.4. The goal of this analysis is to figure out the effect of the observed regulation on the metabolism of the yeast. This application has been carried out with two students at the LMU, Maria Piskarev and Theresa Niederberger, and presented as a poster at the German Conference on Bioinformatics (GCB) 2005 in Hamburg.

For each experiment, all genes with a p-value less than 0.05 were selected. The significance of the overlap with each pathway was then computed using FET. As a second method to estimate the significance of the changes in a pathway, the rank test was employed. The ratios of the genes in each pathway were compared with the ratios of all other genes and the corresponding p-value computed. Figure 6.1 visualizes the results for both tests.

In order to test the performance of the methods, an automatic evaluation scheme was devised. The problem is considered as a classification task where it has to be decided in which pathways a knocked-out gene participates. This property has to be predicted for each pathway/gene pair. Obviously, the participation of a gene in a pathway is not what is measured by the statistical tests. The tests only indicate if the knock-out of a gene influences a pathway. Still, this validation approach is used, as no other 'hard' information about the relationship between genes and pathways is available. Furthermore, we assume that in many cases where a gene is part of pathway, the knock-out of the gene should affect that pathway.

We define a classifier that is based on the p-values from one of the two tests: A pathway/gene pair is classified as a correct pair by the prediction method if the corresponding p-value from FET or WMWT is lower than a given threshold. For a given threshold the number of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) assignments can be computed. The performance of the prediction algorithm can be

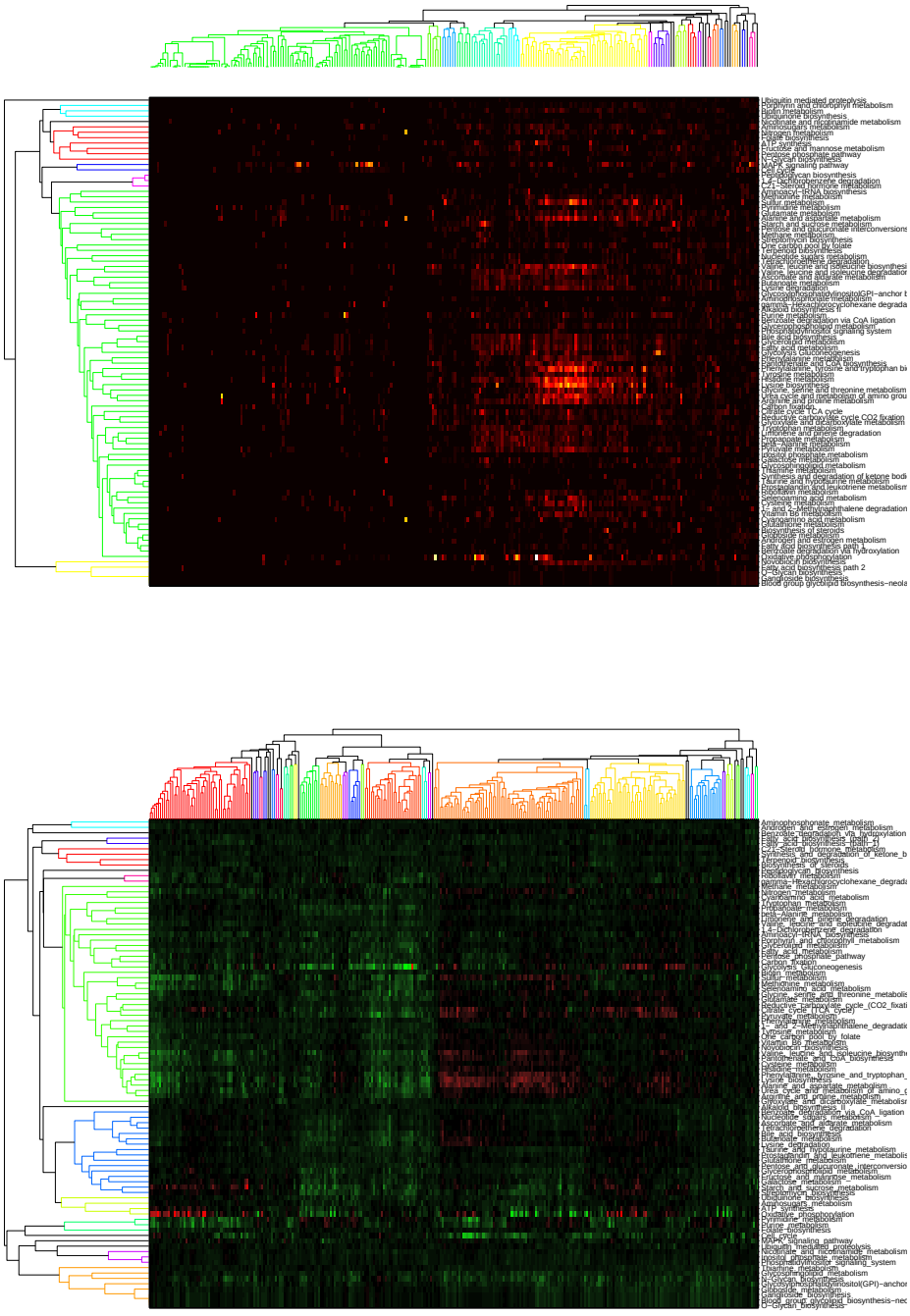

Figure 6.1: Significance values for pathways. Top: Significance values using Fisher's exact test. Bottom: Significance values using Wilcoxon-Mann-Whitney test. Red means up-regulated, green means down-regulated, more significant p-values get higher intensities.

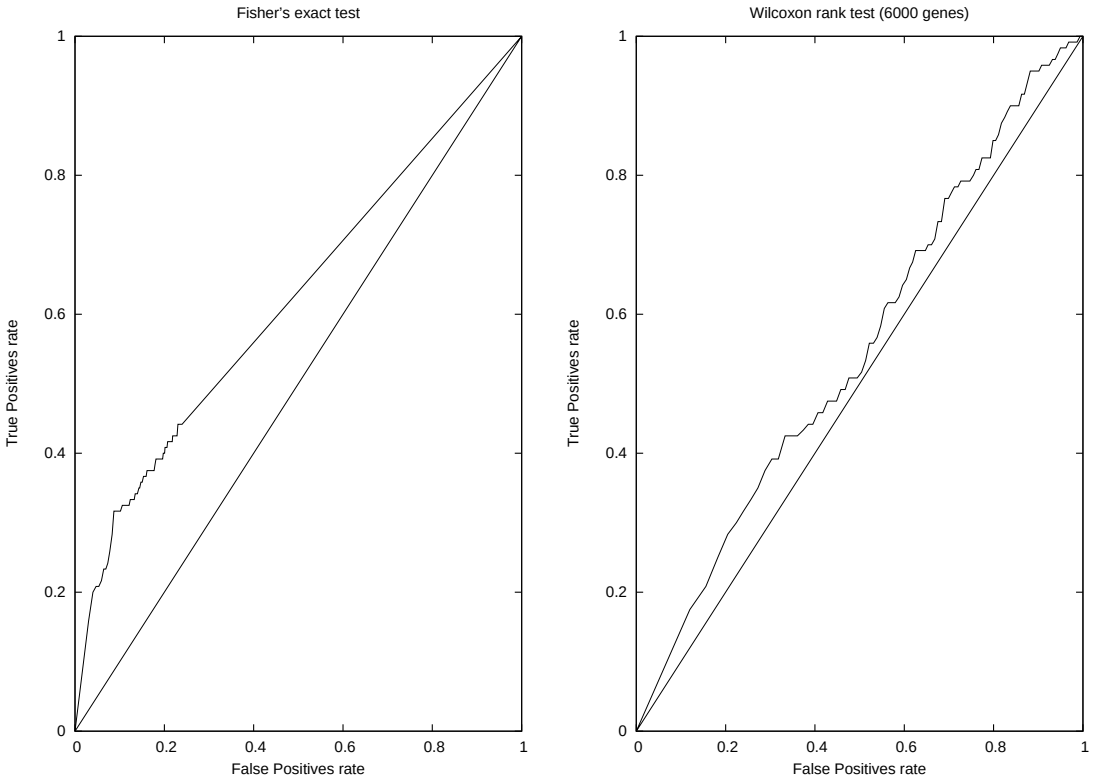

Figure 6.2: Receiver operator curve (ROC) for the threshold classifier based on FET (left) and WMWT (right).

visualized in an ROC plot. The ROC plot for both FET and WMWT is shown in Figure 6.2

The ROC plots seem to indicate that the classifier performance – although better than random guessing – is not very good. But the main reason is that the evaluation does not reflect what the method tries to identify. The tests simply find gene/pathway pairs where the knock-out of the gene affects the regulation of the pathway. This can happen, even if that the gene does not participate in that pathway. The effect can be indirect, for instance, if the knock-out results in a reduced production of some metabolite, another pathway that needs that metabolite may be down-regulated. These indirect effects can potentially lead to many false positives that are actually correct in a biological sense. Therefore, we manually validated some high-scoring hits that are considered false positives in the automatic validation. As such manual validation is usually very time-consuming, a method was desired that could ease the task of finding relevant literature. We use text mining networks generated by ProMiner (Hanisch et al., 2003) and ToPNet to navigate through the literature. Thus, we provide a complete workflow for identifying relevant effects of experiments on pre-defined pathways and validate the results with the help of text-mining tools.

The ROC plots also seem to indicate that the approach using FET works better than

the WMWT. The reason is probably that the two tests are not based on exactly the same information. FET uses p-values for differential expression which have been calculated taking the fluctuations in control measurements into account. These control measurements were not used for calculating the expression ratios that we used for the WMWT.

We have examined four examples for pathways identified as affected by a knock-out. These examples should demonstrate the kind of information that can be gained by using the proposed tests on known pathways. Furthermore, they should explain why there can be significant differences between the results from the two tests and why pathways can be affected even if the knocked-out gene is not part of it.

The Purine metabolism in the hpt1 knockout is identified as affected in the FET approach (p-value 4.910^{-12}, while the WMWT only results in a p-value of 0.31. In SGD, Hpt1p is annotated with the GO term hypoxanthine phosphoribosyltransferase activity. The corresponding EC number is 2.4.2.8. That enzyme is part of the KEGG reference pathway for purine metabolism, but Hpt1p is not present in the corresponding yeast pathway in KEGG. Maybe the enzymatic function of Hpt1p was not known to the KEGG annotators when the yeast pathway was designed. Thus, in our automatic evaluation for FET, this pathway will be counted as a false positive due to a missing annotation in KEGG. Figure 6.3 shows the important parts of the purine metabolism pathway with expression data from the hpt1 knockout experiment. Hpt1p is an enzyme necessary for a step in the salvage pathway, which basically recycles GMP. Thus, the recycling of GMP fails in the mutant, and therefore the de novo synthesis must be up-regulated. The reason for the high p-value for WMWT is that the complete pathway is very large and some genes are down-regulated as well. Therefore the effects on the ranks cancel out.

The Urea cycle and amino group metabolism in the arg80 knockout appears affected using both tests (p-value for FET is 2.010^{-12}, the p-value for WMWT is 0.023). As in the previous example, the identified pathway does not contain the knocked out gene. But using our text mining approach, we could quickly identify the role of arg80 for that pathway from the network around Arg80p (Figure 6.4, right-hand side). Arg80p forms a complex with Arg81p and Mcm1p, and that complex is required for the repression of arg1, arg3, arg5,6 and arg8, which take part in the argenine biosynthesis and for the induction of car1 and car2, which are necessary for argenine catabolism (Turner et al., 2002). Therefore, as Figure 6.4 (left-hand side) shows, arg1, arg3, arg5,6 and arg8 are up-regulated in the mutant and car1 and car2 are down-regulated.

The leucine biosynthesis in the gcn4 knockout gets a p-value of 2.310^{-6} using FET and 9.210^{-6} using WMWT. Again, Gcn4p is not contained in the affected pathway. A literature search using our text mining network reveals that Gcn4p induces Leu3p, which is a transcriptional activator of leu1, leu2, leu4, ilv2 and ilv5 (Natarajan et al., 2001). The corresponding proteins are all involved in the leucine biosynthesis pathway and down-regulated in the gcn4 knockout (Figure 6.5).

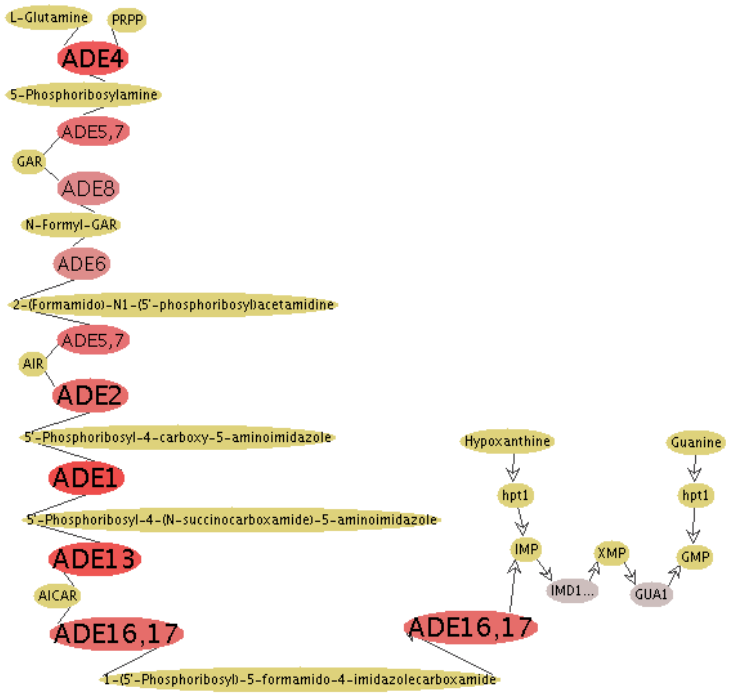

Figure 6.3: The purine metabolism with expression data from the hpt1 knockout experiment. Red color indicates up-regulation in the mutant, green down-regulation.

The MAPK signaling pathway in the ste4 knockout is our only example involving a signaling pathway. Here, the knocked out gene is part of the pathway, the p-values for FET and WMWT are 2.310^{-11} and 0.02, respectively. Therefore, both tests correctly identify the pathway as affected in the ste4 knockout. Figure 6.6 shows what is going on: The Ste4p protein is required for the signal flow from the receptor to the central transcription factor Ste12p. As a consequence, some genes participating in the pathway are down-regulated (the mechanism is not clear for all genes). Although the tests correctly identify the MAPK pathway in this case, it becomes clear that they are not quite appropriate to detect perturbations in signaling pathways. Even though the proteins in a signaling pathway may not be regulated on a transcriptional level, they can induce transcriptional changes in the target genes of the pathway. Therefore, looking at the target genes may often provide more insight into the activity of a pathway than looking at the proteins participating in the pathway. This approach will be used in 6.1.2, where we identify relevant transcription factors by looking at their target genes.

These examples demonstrate that the enrichment analysis on pre-defined pathways often delivers specific and biologically relevant results. With the help of visualization and text mining tools these results can be evaluated and interpreted by the user. On the other hand, the restriction to pre-defined pathways seems a bit strict. Some important links can easily be missed. Looking at metabolic pathways for instance, it is impossible to

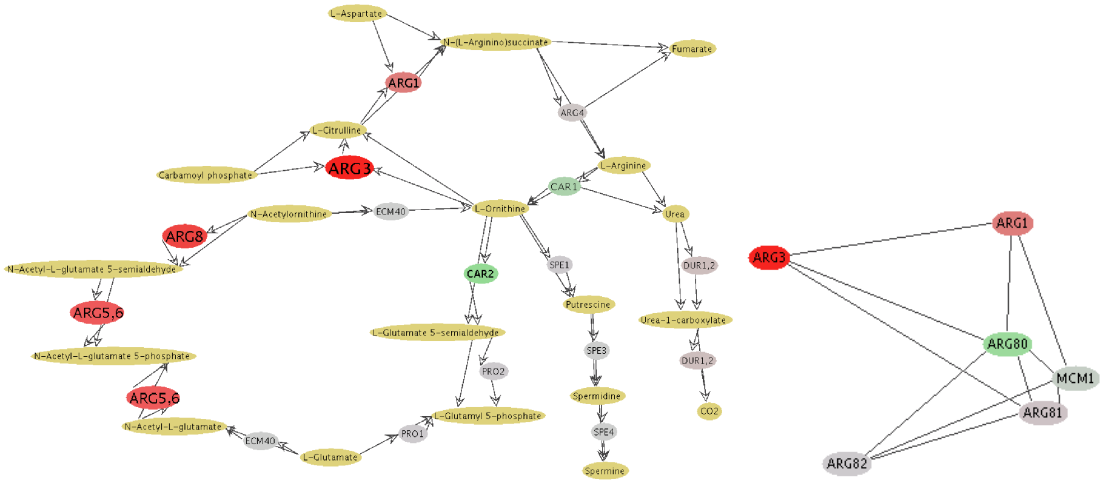

Figure 6.4: Left: The urea cycle and the metabolism of amino groups with expression data from the arg80 knockout experiment. Red color indicates up-regulation in the mutant, green down-regulation. Right: A text-mining network around the knocked out gene arg80.

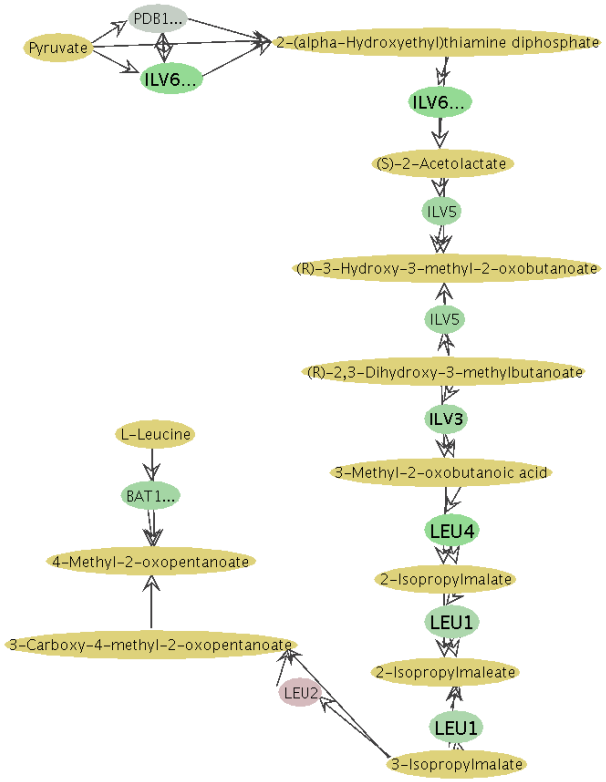

Figure 6.5: The leucine biosynthesis pathway with expression data from the gcn4 knockout experiment. Red color indicates up-regulation in the mutant, green down-regulation.

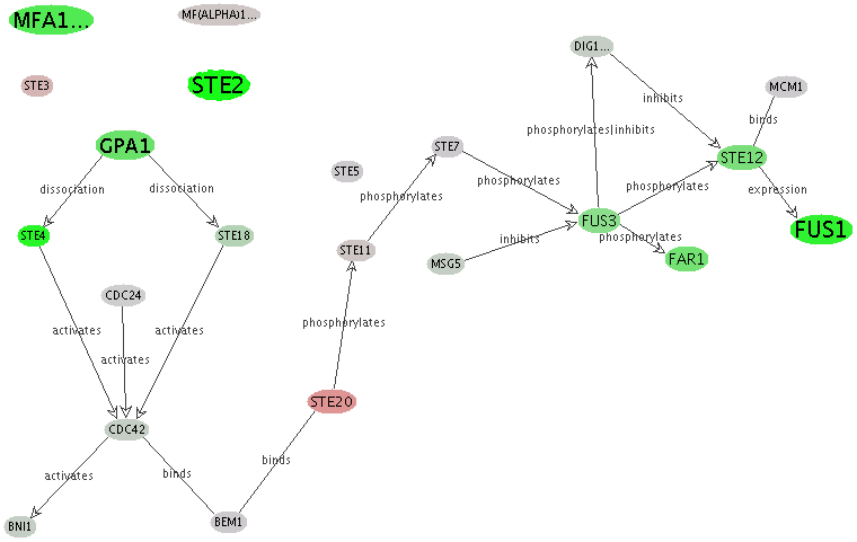

Figure 6.6: The MAPK signaling pathway with expression data from the ste4 knockout experiment. Red color indicates up-regulation in the mutant, green down-regulation.

find the regulatory relationships that are important for the measured expression data. In regulatory pathways, a transcriptional regulation of the participating genes will often not be present. Therefore, such an analysis is more suitable to characterize the effect of the observed expression pattern than to explain the causative regulatory mechanisms.

6.1.2 Activity of transcription factors

In order to determine transcription factors that are relevant for the different expression patterns in knock-out experiments, FET activity scores of every transcription factor were computed for all 300 expression measurements as described in section 5.4.3. In order to visualize the results, we constructed hierarchical clusters using the Spearman rank correlation and average linkage (Figure 6.7) on the activity scores for all transcription factors with at least one significant score. Table 6.1 lists the 25 best scoring results. For evaluation, we manually assess some prominent features of the results and look for evidence in the literature.

The highest value in the matrix is attained by the transcription factor Ste12p for the dig1/dig2 mutant. All targets of Ste12p are up-regulated, as expected, since Dig1p and Dig2p are needed for the repression of pheromone-responsive transcription (Bardwell et al., 1998). Figure 6.9 shows Ste12p and its targets with expression data from the dig1/dig2 knockout experiment.

In addition, Ste12p is identified as the most relevant transcription factor for the kss1/fus3 knock-out. These two MAP kinases are also known to regulate transcription of pheromone response genes through Ste12p (Tedford et al., 1997).

Bas1p has a very high score in the hpt1 experiment. As discussed in 6.1.1, hpt1 muta-

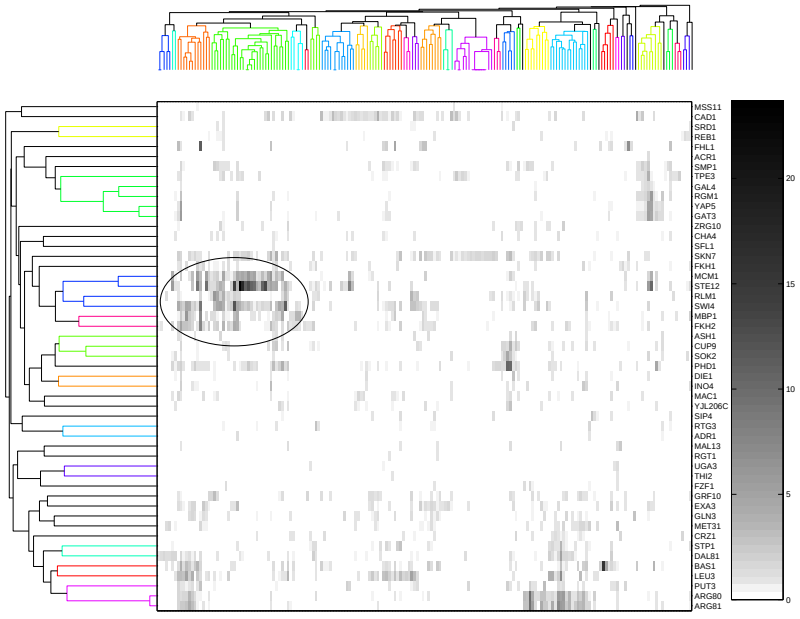

Figure 6.7: Inferred activities of 51 transcription factors. Average linkage hierarchical clustering was performed with spearman rank correlation on 300 expression measurements. Values are $-\log_{10} p$, where p is a p-value quantifying the significance of a transcription factor for the expression data. The marked area contains mainly experiments that affect the MAPK signaling pathway (e.g. knock-outs of ste4, ste5, ste7, ste12, ste18, ste24, fus3/kss1, and dig1/dig2). The affected transcription factors are Mcm1p, Ste12p, Rlm1p, Swi4p, Mbp1p, and Fhk2p.

tions affect the purine biosynthesis pathway, which is regulated by Bas1p (Guetsova et al., 1997; Daignan-Fornier and Fink, 1992).

Arg80p and Arg81p are the transcription factors with the highest scores in the knockout experiment of arg80. Figure 6.8 shows these two transcription factors with their regulated targets in that experiment. Indeed, the transcription factor that was knocked out should be important for the regulation. Furthermore, it is known that Arg80p and Arg81p are necessary for the repression of anabolic genes in the argenine biosynthesis. The four targets (Arg5,6p, Arg3p, Arg8p and Cpa1p) which are all up-regulated catalyze different steps in that metabolic pathway.

These results demonstrate that high scoring hits indeed deliver regulatory contexts that are important for the experiment under consideration.

Experiment	Transcription Factor	Score
dig1,dig2(haploid)	STE12	23.7
fus3,kss1(haploid)	STE12	17.4
hpt1	BAS1	15.7
dig1,dig2	STE12	14.0
sst2(haploid)	STE12	13.3
ste18(haploid)	STE12	12.9
dig1,dig2	SWI4	12.1
ERG11(tetpromoter)	FHL1	11.3
kin3	SWI4	11.2
ste12(haploid)	STE12	10.9
ste7(haploid)	STE12	10.6
ssn6(haploid)	PHD1	10.5
ste4(haploid)	STE12	10.4
tup1(haploid)	PHD1	10.3
FR901,228	STE12	10.1
ste5(haploid)	STE12	9.9
arg80	ARG80	9.8
ste24(haploid)	STE12	9.6
arg80	ARG81	9.5
ERG11(tetpromoter)	SWI4	9.2
sod1(haploid)	STE12	9.0
ste11(haploid)	STE12	7.9
ERG11(tetpromoter)	FKH2	7.7
ERG11(tetpromoter)	MBP1	7.2
pep12	ARG81	7.2

Table 6.1: The 25 top scoring transcription factors together with their activity scores as described in Section 5.4.2. The table displays the most relevant transcription factors for individual experiments, e.g. Phd1p received a score of 10.5 in the ssn6 knock-out experiment.

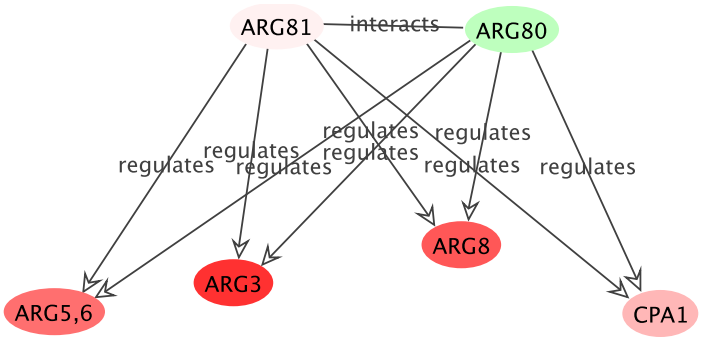

Figure 6.8: Arg80p and Arg81p and their regulated targets in the arg80 knock-out strain. The shade of the nodes represents the magnitude of the expression ratio (mutant vs. wild-type). All genes except arg80 are up-regulated. Arg80p and Arg81p are the two most relevant transcription factors for the arg80 mutant.

Transcription Factor 1	Transcription Factor 2	Spearman Correlation
ARG80	ARG81	0.85
YAP5	GAT3	0.79
RGM1	GAT3	0.70
RGM1	GAL4	0.67
MCM1	STE12	0.66
RGM1	YAP5	0.62

Table 6.2: The six most highly correlated pairs of transcription factors. Correlation was computed using Spearman rank correlation on the activity scores of the transcription factors. Interestingly, there are only few pairs with high correlation (see Figure 6.10), although related transcription factors often cluster together nicely (Figure 6.7 left).

6.1.3 Correlation analysis of activity scores

So far, all results have been obtained using only one experiment from the set of expression data at a time. In order to figure out if we can identify cooperating transcription factors, we look at pairs of transcription factors that correlate well in their activity scores. Table 6.2 shows all pairs of transcription factors with a spearman rank correlation greater than 0.6 which corresponds to a z-score of 4.6. Figure 6.10 shows a histogram of all correlations.

Arg80p and Arg81p have the highest correlation; their role was already discussed in the context of the best scoring single transcription factors. Next, we take a closer look at the pair Mcm1p and Ste12p. As with Arg80p and Arg81p, these transcription factors interact according to DIP. The experiments where they are active include knock-outs of ste4, ste5,

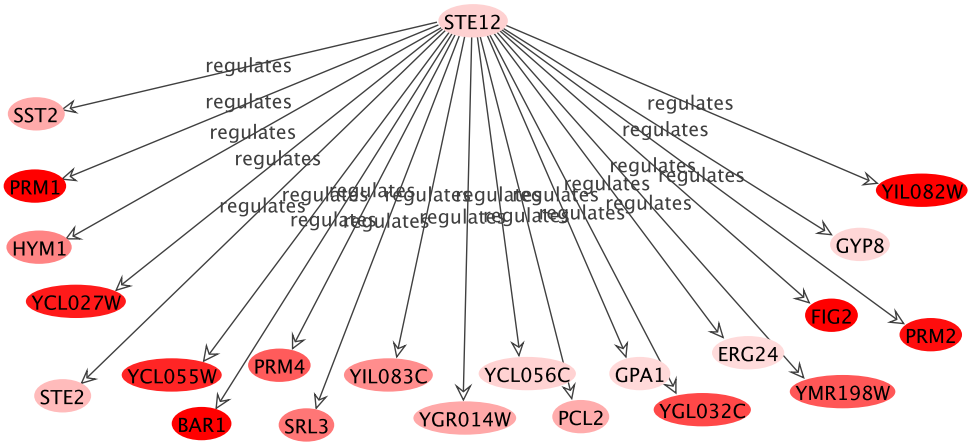

Figure 6.9: Ste12p and its regulated targets in the dig1/dig2 knock-out strain. The shade of the nodes represents the magnitude of the expression ratio (mutant vs. wild-type). All genes are up-regulated.

ste7, ste11, ste12, ste18, ste24, fus3/kss1 and dig1/dig2.

Indeed, as these results suggest, mating functions like pheromone maturation, pheromone response and cell fusion are cooperatively controlled by the transcription factors Ste12p and Mcm1p (Hwang-Shum et al., 1991; Dolan et al., 1989). Interestingly, this cooperative behavior could not have been identified using only the expression data of STE12 and MCM1, as these do not correlate. On the other hand, this correlation could probably have been predicted by looking at the target genes of Ste12p and Mcm1p. Although the potential targets of Ste12p and Mcm1p do not overlap to a large extend (88 potential targets for Mcm1p, 51 for Ste12p and 8 overlapping), the corresponding p-value from Fisher's exact test for this overlap is less than 6×10^{-7}. Still, the combination of potential targets and their expression data creates additional evidence and leads to the correct conclusion (Figure 6.11).

For the other pairs we did not find additional literature evidence for a functional relationship, but we believe that this is worth investigating.

6.1.4 Activity of kinases

Using the same scoring method and the *pathway query* described in Figure 5.9, we computed activity scores for kinases. Our algorithm detects much fewer high scoring kinases than transcription factors. The ten highest scores are listed in Table 6.3.

The highest score is attained for the dig1/dig2 mutant and the MAP kinase Kss1p. As was already mentioned, the Kss1p regulates transcription through Ste12p.

The network that constitutes the second best score contains the kinase Slt2p and its regulated targets in the kin3 knock-out experiment (Figure 6.12). We could find evidence that Slt2p regulates Swi4p (Baetz et al., 2001) and Rlm1p (de Nobel et al., 2000), but the

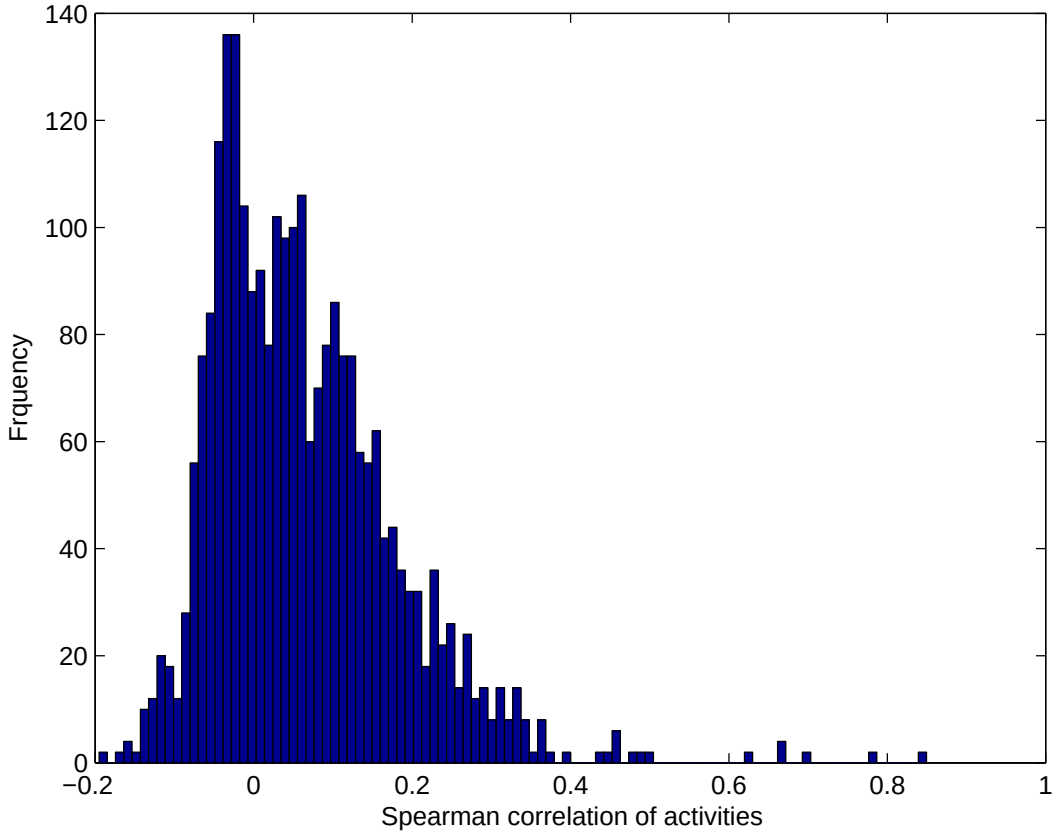

Figure 6.10: Histogram of the correlations between the activities of transcription factors.

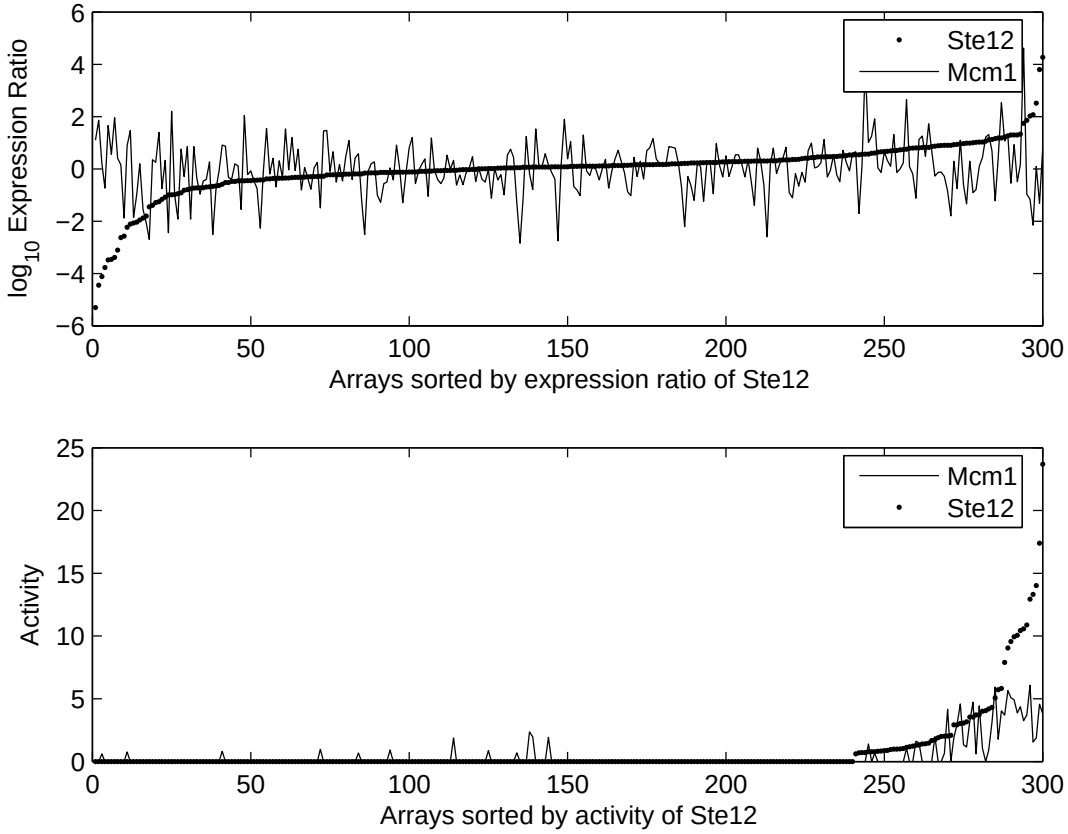

Figure 6.11: Correlation of Ste12p and Mcm1p with respect to expression values and activity score. While no correlation for the expression data can be detected, inferred activities clearly correlate.

connection to the kin3 mutant is not clear, since not much has been published about the function of Kin3p.

In general, the results for the kinases are harder to validate, but on the other hand, the implied hypotheses are more detailed and more interesting.

6.1.5 Cooperating transcription factors

In section 6.1.3 we tried to identify cooperating transcription factors by correlating activity scores of single transcription factors. We also defined another *pathway query* that uses the protein interaction data from our background network to find pairs of transcription factors that cooperatively regulate sets of genes (Figure 6.13). We computed all instances of this *pathway query* with the *pathway search* algorithm and scored pairs of transcription factors using the same method as before for single transcription factors. The 25 transcription

Experiment	Kinase	Score
dig1,dig2(haploid)	KSS1	17.0
kin3	SLT2	10.2
dig1,dig2	KSS1	9.3
ERG11(tetpromoter)	DUN1	9.2
ERG11(tetpromoter)	ESR1	9.2
swi4	ELM1	7.9
swi4	IPL1	7.9
ERG11(tetpromoter)	MKK2	7.8
ERG11(tetpromoter)	SLT2	7.8
ERG11(tetpromoter)	CKB1	6.6

Table 6.3: Top 10 scores for kinases together with their activity scores as described in Section 5.4.2. The table displays the most relevant transcription factors for individual experiments, e.g. Kss1p was identified for the dig1,dig2 knock-out experiment.

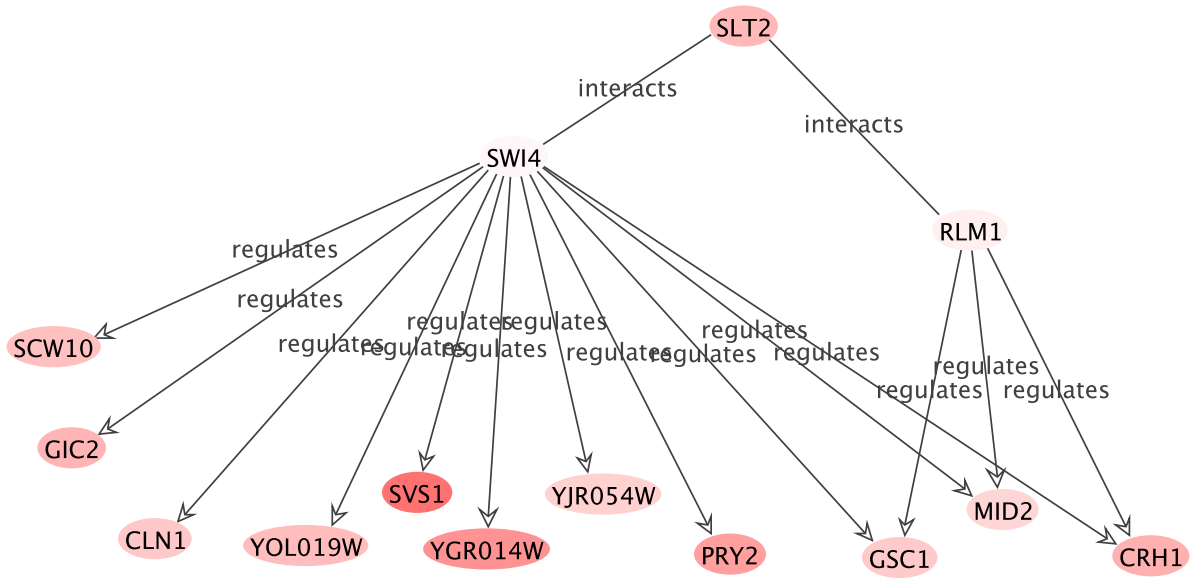

Figure 6.12: Kinase Slt2p, interacting transcription factors and regulated genes in the kin3 mutant. The shade of the nodes represents the magnitude of the expression ratio (mutant vs. wild-type). All genes are up-regulated.

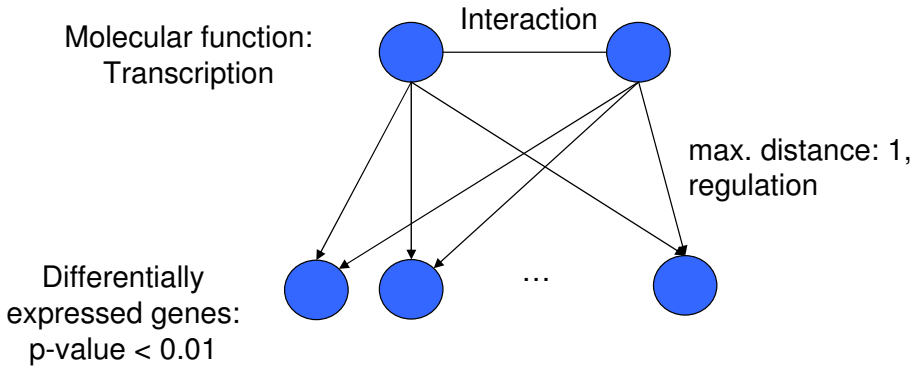

Figure 6.13: Pathway query for finding cooperating transcription factors. This query is used to find interacting pairs of transcription factors that both have protein-DNA interactions with a common set of regulated genes.

factor pairs with the highest scores are shown in Table 6.4. As expected from the previous results, we could find many high scoring instances with Arg80p, Arg81p and Ste12p, Mcm1p respectively. In addition, we found two instances with Hir1p and Hir2p which are involved in cell-cycle regulated transcription of histone genes (Sherwood et al., 1993); the knocked-out genes are hir2 itself and swi4 which is also a cell cycle dependent transcription factor. The last pair in the list is Fkh2p, Mcm1p which are known to bind co-operatively to their targets (Hollenhorst et al., 2001).

Again, these results demonstrate how our algorithm can extract relevant contexts from the data in a very flexible way.

6.1.6 Association rule mining

In order to find transcription factors which are unique regulators for certain target genes, or target genes that are always regulated when a transcription factor is active, we looked for corresponding association rules using the Apriori algorithm as implemented by Borgelt and Kruse (2002). Transactions are all instances of the *pathway query* for identifying active transcription factors with a p-value of less than 0.05. Items in a transactions are genes with their role in that instance, i.e. a gene can appear more than once in a transaction if it instantiates more than one query node. There is a total of 557 such transactions.

First, we enumerated all rules with transcription factors as antecedent, a minimal support of 1% (six item sets) and a minimal confidence of 60%. The result is a list of transcription factors and corresponding targets where the target is differentially expressed in most cases when the transcription factor's activity changes. Therefore, these targets can be interpreted as a high-confidence list of targets. For instance, if a transcription factor is necessary for the basal expression of a gene, and the transcription factor is inactivated in many experiments, that pair would show up in the generated rules. The complete list of

Experiment	Transcription Factors	Score
arg80	ARG80, ARG81	11.2
sod1(haploid)	STE12, MCM1	7.7
pep12	ARG80, ARG81	7.0
vps8	ARG80, ARG81	7.0
fus3,kss1(haploid)	STE12, MCM1	6.9
rtg1	ARG80, ARG81	6.7
FR901,228	STE12, MCM1	5.8
yor080w	STE12 ,MCM1	5.6
AUR1(tetpromoter)	ARG80, ARG81	5.6
ste18(haploid)	STE12, MCM1	5.5
dig1,dig2(haploid)	STE12, MCM1	5.2
yhl029c	ARG80, ARG81	5.0
ste12(haploid)	STE12, MCM1	4.9
erg3(haploid)	ARG80, ARG81	4.7
ste24(haploid)	STE12, MCM1	4.5
HMG2(tetpromoter)	STE12, MCM1	4.4
hir2	HIR2, HIR1	4.4
ymr010w	ARG80, ARG81	4.3
ste5(haploid)	STE12, MCM1	4.1
KAR2(tetpromoter)	STE12, MCM1	4.0
yjl107c(haploid)	STE12, MCM1	3.9
yer044c(haploid)	ARG80, ARG81	3.7
yer044c(haploid)	STE12, MCM1	3.7
swi4	HIR2, HIR1	3.7
ERG11(tetpromoter)	FKH2, MCM1	3.6

Table 6.4: Top 25 scores for cooperating transcription factors together with their activity scores as described in Section 5.4.2. The table displays the most relevant transcription factors for individual experiments, e.g. Ste12p and Mcm1p were identified for the ssn6 knock-out experiment.

```
<TheNet name="Transcription Factors in Yeast Compendium">

<Subnet multiple="false" name="Transcription Factor">
  <PathwayNode name="TF">
    <Query>
      <BasicQuery Negated="false">
        <MapName>GO: molecular_function</MapName>
        <Operator>like</Operator>
        <Value>Transcription Factor</Value>
      </BasicQuery>
    </Query>
  </PathwayNode>
</Subnet>

<Subnet multiple="true" name="Regulatee">
  <PathwayNode>
    <Query>
      <BasicQuery Negated="false">
        <MapName>anp1:P-value</MapName>
        <Operator>lt</Operator>
        <Value>0.01</Value>
      </BasicQuery>
    </Query>
  </PathwayNode>
</Subnet>

<Connection maxEdges="1" undirected="false">
  <ConnectFrom>Transcription Factor</ConnectFrom>
  <ConnectTo>Regulatee</ConnectTo>
  <TransitionQuery>
      <BasicQuery Negated="false">
        <MapName>InteractionMode</MapName>
        <Operator>like</Operator>
        <Value>regulates</Value>
      </BasicQuery>
  </TransitionQuery>
</Connection>
</TheNet>
```

Figure 6.14: A *pathway query* used to find transcription factors and their regulated targets in the anp1 knock-out experiment.

the inferred high-confidence targets is shown in Table 6.5. The rule with the highest support concerns the transcription factor Ste12p and its target sst2. The high support simply reflects the fact that Ste12p is the transcription factor that has high activity scores most often; it appears in 41 of the 557 transactions. Sst2p negatively regulates the pheromone response pathway by activating the GTPase activity of Gpa1p (Apanovitch et al., 1998). In the Saccharomyces genome database (Cherry et al., 1998), binding sites for Ste12p and Dig1p are annotated to the upstream region of the sst2 gene, supporting a regulation of sst2 by Ste12p. This regulation constitutes a negative feedback loop for the mating response, as the target transcription factor for the mating response pathway is Ste12p. When Ste12p up-regulates sst2, the production of Sst2p leads to pheromone desensitization, making the feedback loop complete (Bardwell, 2004).

Other rules are concerned with the regulation of argenine biosynthesis by Arg80p and Arg81p or the a-cell specific genes aga1 and aga2 by MCM1.

Next, we enumerated rules with target genes as the antecedent, a minimum support of 3% and a confidence of 100% (Table 6.6). A rule from this list means that whenever target gene (or set of target genes) that constitutes the antecedent appears in a significant regulation context, the corresponding transcription factor is always the one given in the consequent, i.e. that transcription factor is probably necessary for the regulation of the antecedents. For instance, if a transcription factor is the sole regulator for a gene, that pair would be listed among the rules.

6.1.7 Finding signaling cascades

In some cases, if the knocked out gene is part of a signaling pathway, it can be useful to find direct paths from that gene to a relevant transcription factor. These paths can serve as hypotheses for the observed regulation, as the signal flow along them obviously cannot function in the mutant. We have designed a corresponding *pathway query* and applied it to some mutants from the mating pathway. The query uses the FET score of the transcription targets to identify significant networks, i.e. everything above the transcription factor is determined by the network structure alone and does not take the expression values into account. Figure 6.15 shows the most significant networks that were found using that *pathway query*. While the resulting networks do not show the exact mechanism of regulation, they contain at least a part of it. The mating pathway, namely the kinases Ste5p and Kss1p and the central transcription factor Ste12p are part of the most significant network. As ste4 is knocked out, it appears down-regulated in the figure, and the signaling cascade activating the transcription factor Ste12p can not take place. Therefore the targets of Ste12p are down-regulated as well. Here it becomes obvious that networks of high quality containing directions of signal flow and further annotations are essential to extract detailed information. The genes upstream of Ste12p are not differentially expressed, therefore it is impossible to determine from the expression data of a single experiment which gene actually participates in the signaling cascade and which gene does not. The second network contains the transcription factor Swi4p, a cell cycle regulator which is connected to Cln2p and Slt2p, both known to be involved in cell cycle regulation as well. Furthermore,

Target gene (consequent)	Transcription factor (antecedent)	support	confidence
YNL057W	TF:GRF10	1.4%	62.5%
AGP1	TF:STP1	1.1%	83.3%
STR3	TF:MET31	1.3%	71.4%
HIS4	TF:GRF10	1.4%	62.5%
MCH5	TF:PUT3	2.2%	75.0%
CPA1	TF:PUT3	2.2%	66.7%
YLL066C	TF:YAP5	1.1%	66.7%
HOR7	TF:EXA3	2.5%	64.3%
ARG3	TF:ARG80	4.7%	61.5%
ARG5,6	TF:ARG80	4.7%	65.4%
ARG8	TF:ARG80	4.7%	76.9%
LEU1	TF:LEU3	4.7%	73.1%
BAT1	TF:LEU3	4.7%	80.8%
ARG3	TF:ARG81	4.7%	61.5%
ARG5,6	TF:ARG81	4.7%	69.2%
ARG8	TF:ARG81	4.7%	73.1%
ADE17	TF:BAS1	4.8%	70.4%
CRH1	TF:RLM1	3.9%	68.2%
RPL25	TF:FHL1	2.0%	63.6%
SST2	TF:STE12	7.4%	65.9%
AGA2	TF:MCM1	5.9%	60.6%
STE2	TF:MCM1	5.9%	69.7%
AGA1	TF:MCM1	5.9%	72.7%

Table 6.5: High-confidence targets of transcription factors inferred using association rules. All rules with a transcription factor as antecedent, a minimum support of 1% and a minimum confidence of 60% are listed.

Transcription factor (consequent)	Target genes (antecedent)	support	confidence
TF:LEU3	LEU1	3.4%	100.0%
TF:LEU3	BAT1	3.8%	100.0%
TF:SKN7	ZPS1	3.1%	100.0%
TF:BAS1	ADE17	3.4%	100.0%
TF:STE12	KAR4	3.1%	100.0%
TF:STE12	FUS1	3.6%	100.0%
TF:STE12	TEC1	4.1%	100.0%
TF:FKH2	DSE1	3.1%	100.0%
TF:SWI4	PRY2	3.2%	100.0%
TF:STE12	SST2	4.8%	100.0%
TF:STE12	FUS1, AGA2	3.1%	100.0%
TF:STE12	TEC1, SST2	3.4%	100.0%
TF:STE12	TEC1, STE2	3.1%	100.0%
TF:STE12	GPA1, SST2	3.4%	100.0%
TF:STE12	GPA1, STE2	3.1%	100.0%
TF:STE12	SST2, AGA2	3.1%	100.0%
TF:STE12	SST2, STE2	3.8%	100.0%
TF:MCM1	AGA2, AGA1	3.6%	100.0%
TF:MCM1	STE2, AGA1	3.6%	100.0%
TF:STE12	GPA1, SST2, STE2	3.1%	100.0%
TF:MCM1	AGA2, STE2, AGA1	3.1%	100.0%

Table 6.6: Necessary regulators for target genes inferred by association rules. All rules with target genes as antecedent, a minimum support of 3% and a confidence of 100% are listed.

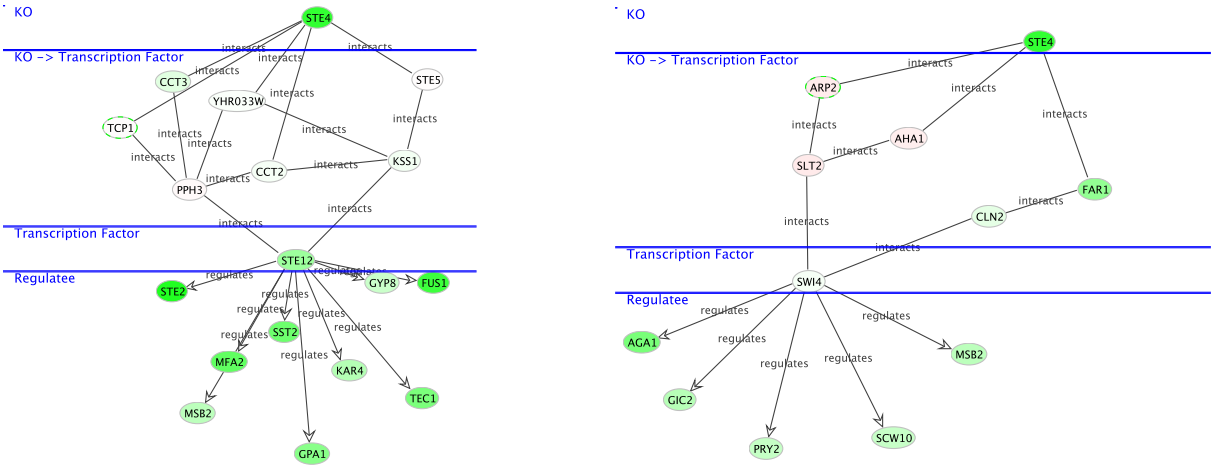

Figure 6.15: Signaling cascades explaining the effect of the ste4 knock-out. These networks were identified using a *pathway query* containing the knocked out gene ste4, a connection to a transcription factor via at most three protein-protein interactions, and genes regulated by that transcription factor. The depicted networks were the two most significant among five networks found by the *pathway query*.

it is well known that there is a connection between the mating functions and the cell cycle as the cell cycle is arrested in G1 phase for mating. Thus, the two networks identified using our *pathway query* contain a lot of relevant information and point the user toward biologically meaningful processes and proteins.

The expression data from the yeast compendium data was measured in a steady state and is not an immediate effect to some environmental change. Since the yeast was grown as a mutant, transcriptional effects can also be indirect. For instance, the knocked out gene might be part of a signaling pathway, and the target genes of that pathway include other transcription factors. Then, the target genes of these transcription factors are also expected to be regulated. Again, we have designed a *pathway query* that reflects this hypothesis and applied it to the tup1 knockout data. The *pathway query* now contains two transcription factors, the second of which must be a target of the first. In order to score the resulting instances, the rank score for both target groups is calculated separately and then the resulting p-values are combined. Figure 6.16 shows the most significant resulting instance according to that score.

Other indirect effects are also possible. For instance in the knockout of hpt1, the indirection is probably mediated via a metabolite, namely guanosine monophosphate (GMP). As described in 6.1.1 Hpt1p is part of the purine salvage pathway. In the mutant that pathway is not functional, therefore GMP concentrations drop, leading to an increase of purine production via an activation of Bas1p and its targets. In order to identify such indirections networks have to be considered that contain the corresponding information on the metabolism and more importantly about interactions between metabolites and tran-

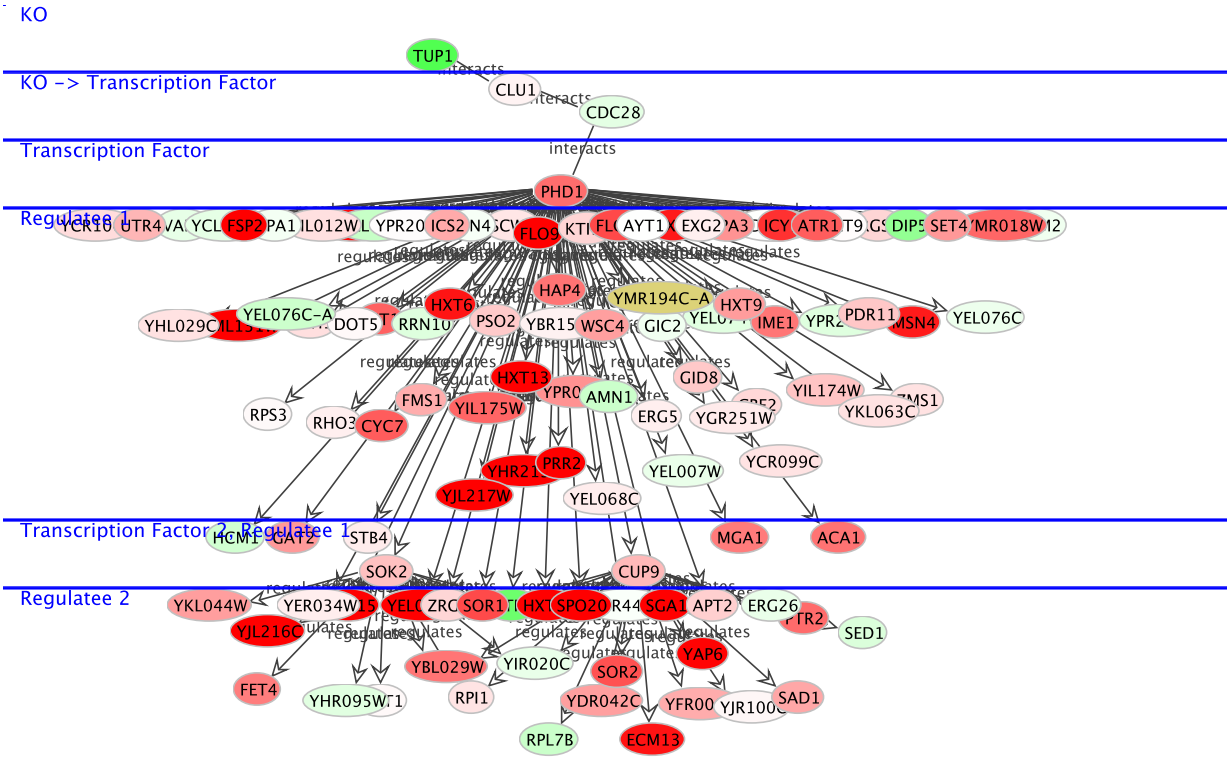

Figure 6.16: Direct and indirect effects of the tup1 gene knockout. This network shows a possible direct effect of the knockout, consisting of the regulation of target genes of the transcription factor Phd1p. In addition, some of these targets are themselves transcription factors, and their targets are differentially expressed as well.

scription factors or regulatory pathways. If such networks are available, it is possible to search for regulatory effects mediated via metabolites using *pathway queries*.

6.1.8 Discussion

Biologists using microarray technology are often confronted with the problem of interpreting lists of regulated genes. Sorting these lists with respect to functional annotation and identifying over-represented classes is often not sufficient to provide insight into the mechanisms leading to the observed expression patterns. Differentially expressed genes have to be interpreted in context with their regulators like transcription factors and signaling molecules in order to derive causal relationships and networks. Other interesting contexts could include proteins from the same metabolic pathway or even metabolites. The biological expert should be able to examine his data in a context that appears meaningful to him.

The *pathway query language* provides a formalism to formulate biological hypotheses that can provide such a context for the analysis of expression data.

In this work we have performed the necessary steps for using *pathway queries* in several research questions on public data sets with encouraging results. Using a background network containing protein-protein interactions and DNA binding information, we were able to show that information on regulatory contexts can be valuable for the interpretation of expression data. With the presented method, we could identify active transcription factors, active interacting pairs of transcription factors and to some degree active kinases in single expression measurements. In addition, the method can deliver a clear interpretation of the data or at least exhibit a testable hypothesis as the results include not only the regulators but also the regulated genes and can be visualized as networks.

Although one of the strengths of the approach is its ability to achieve results with single or few measurements available, we could demonstrate its merits also for large scale analyses. Correlation analysis of the computed activity scores revealed the potential to use the method to predict interactions between transcription factors.

6.2 Activity of transcription factors in the development of Drosophila Melanogaster

The results for the yeast compendium dataset are quite encouraging, but it remains to be shown that regulator activity can be inferred for more complex organisms as well. Therefore a study on five transcription factors and their activity during the Drosophila life cycle was performed. This study was presented at the Moscow Conference on Computational Molecular Biology 2005, and it is joint work with Jan Gewehr.

6.2.1 Data

In Arbeitman et al. (2002) gene expression data were collected at 66 time points during the life cycle of fruit fly Drosophila Melanogaster. The eight measurements in the adult stage were taken for the male and the female fly; all measurements were repeated twice. In addition, there are some measurements for mutations, yielding 182 measurements in total. From these data, we tried to infer transcription factor activities throughout the Drosophila life cycle. We chose five transcription factors known to be relevant in early Drosophila development: Caudal (Cad), Bicoid (Bcd), Hunchback (Hb), Knirps (Kni), and Krueppel (Kr). These transcription factors were studied by Berman et al. (2002) in order to find cis-regulatory regions of DNA by looking at clusters of binding sites. In order to generate binding site models, we extracted a set of known binding sites of the transcription factors from that publication. Genome sequences and GO annotations were downloaded from Ensembl using EnsMart (now called BioMart)[1].

6.2.2 Binding site prediction

In order to apply the scorings for transcription factor activity introduced in section 5.4, potential targets for the transcription factors have to be defined. As databases such as TRANSFAC (Wingender et al., 2000) contain only few such relationships, it was decided to rely on computational methods for binding site prediction. Using the binding sites from Berman et al. (2002), we constructed position specific weight matrices with the publicly available EMBOSS[2] tool *prophecy*. Using another EMBOSS tool, *profit*, the up- and downstream regions of all annotated Drosophila genes were scanned with resulting matrix. As such an approach is known to produce many false positives and, as shown in 5.4.4, good specificity is necessary for the activity prediction to be successful, different strategies were employed to reduce the number of false positives. The first approach is based on the assumption that many regulatory sequences contain multiple binding sites for the same transcription factor. Therefore we consider a gene a potential target for a transcription factor only if it has more binding sites than a predefined threshold. The second approach takes advantage of the availability of several insect genomes in draft or completed form. Multiple alignments between sequences of six Drosophila species (D. melanogaster, D. pseudoobscura, D. yakuba, D. ananassae, D. virilis, D. mojavensis) and Anopheles gambiae and Apis mellifera were downloaded from the UCSC website[3]. A transcription factor's target is then a gene that has a binding site which is conserved in at least four of these genomes in one of the multiple alignments. All subsequent analyses were performed for each of the target prediction approaches.

In order to investigate combinations of transcription factors, we computed the intersection of target sets for each pair of transcription factors. This way, cooperation between

[1] `http://www.ensembl.org/Multi/martview/`

[2] `http://emboss.sourceforge.net/`

[3] `http://hgdownload.cse.ucsc.edu/downloads.html#fruitfly`

transcription factors can be identified, if the targets of either of them do not appear regulated to a significant fraction, but the intersection does.

6.2.3 Predicted transcription factor activity

We computed the rank scores for the five transcription factors under consideration and for all pairs of these transcription factors using both target prediction methods. Results are shown in Figures 6.17 and 6.18. While the color intensities are derived from the rank scores, each row of the depicted matrix can be interpreted as an expression profile of the corresponding transcription factor's targets. A bright red value, for instance, means that these targets are strongly up-regulated. The highest scores in Figure 6.17 are attained for Caudal in the early embryonic and the female adult experiments with a value around -3, which corresponds to a p-value of 10^{-3} for down-regulation. Such scores are not significant enough to conclude much without additional support. Still, the observed pattern for caudal activity looks interesting since caudal is a maternally transcribed gene and, thus, in the stages of high scores (early embryo and adult female) maternal transcript is available. On the other hand, it is unclear why the target genes of caudal should be down-regulated in these stages. One possible explanation could be that caudal functions as a repressor for many of its target genes, but we could not find convincing evidence in the literature for that hypothesis. Bicoid is another maternally transcribed transcription factor, and it is known to inhibit the translation of caudal mRNA. As bicoid mRNA and protein levels follow a concentration gradient from anterior to posterior, it can establish a gradient for Caudal with high concentrations only at the posterior. Therefore, the Bicoid protein could contribute to the observed activity pattern of caudal. But if that would be the case, we would expect the activity pattern of Bicoid to follow its expression pattern, but it resembles more the activity pattern of Caudal.

It is also known that the translation of maternal transcription factors like caudal and bicoid is repressed in the unfertilized egg. This could explain the strongly negative activity values of caudal in the adult female. After the egg is fertilized, translation of caudal starts, but it takes some time until enough protein is produced to activate the transcription of many target genes. Furthermore, it is believed that Caudal itself only provides a basal level of activity; other transcription factors are required to induce expression of target genes specifically to the required levels. Thus, the activities of other transcription factors that were not studied here can be of importance. Such transcription factors also have to be produced before the targets that they have in common with Caudal can be transcribed efficiently. This could explain the slow increase of expression levels of Caudal targets during embryogenesis.

Looking at Figure 6.18, where targets were computed using information from comparative genomics, we find that scores are much higher, for Caudal ranging from -15 to +15. The pattern for Caudal looks quite similar to the pattern observed with the other strategy for target prediction (Figure 6.17). Only in the late embryonic stage, we get high scores with the comparative genomics strategy where scores were close to zero before.

In order to test the hypothesis that bicoid is involved in the activity pattern observed

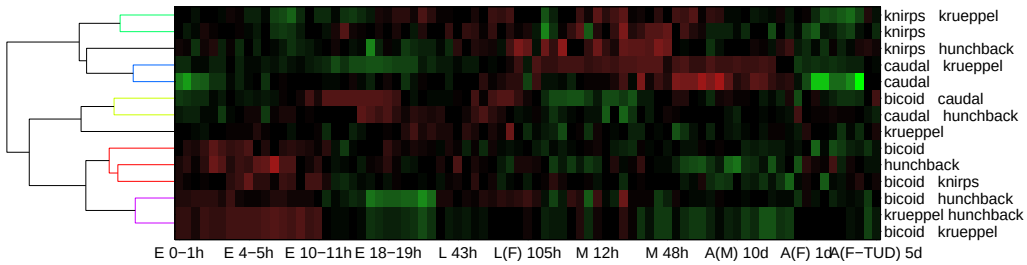

Figure 6.17: Predicted transcription factor activities during the Drosophila life-cycle. Genes with multiple binding sites are considered targets of a transcription factor. Color intensity represents the p-value from the rank test (lower p-values have higher intensities). Green spots show down-regulation of the target genes, red spots up-regulation.

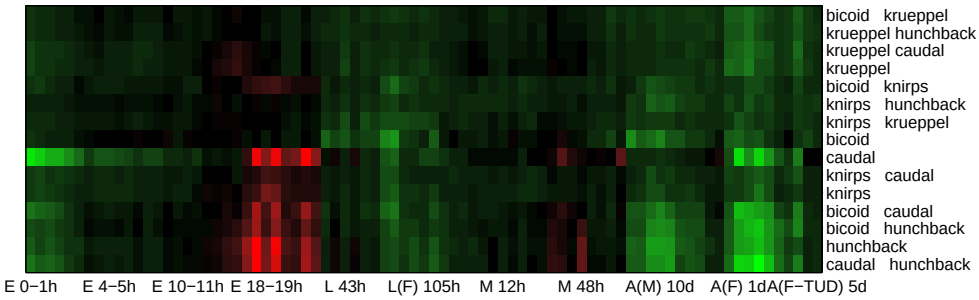

Figure 6.18: Predicted transcription factor activities during the Drosophila life-cycle. Genes with binding sites conserved in multiple species are considered targets of a transcription factor. Color intensity represents the p-value from the rank test (lower p-values have higher intensities). Green spots show down-regulation of the target genes, red spots up-regulation.

for caudal, we plotted the bicoid expression levels against the Caudal activity scores in Figure 6.19. There is a clear correlation of the two curves. In the early embryo and in the adult female, when bicoid message is present, the activity scores for caudal are negative.

While a concluding evaluation of the results can only be made with the help of biological experiments, we think that at least interesting hypotheses can be generated using our activity scores.

6.2.4 Activities in GO classes

As we model only the regulatory influence of transcription factors and ignore all other possible mechanisms of regulation, we expect that a transcription factor works differently on different biological processes due to regulation that is not covered in our approach. In order to produce specific results, that can capture the effects of other regulatory mechanisms, we performed the activity analysis for single biological processes. For instance, we

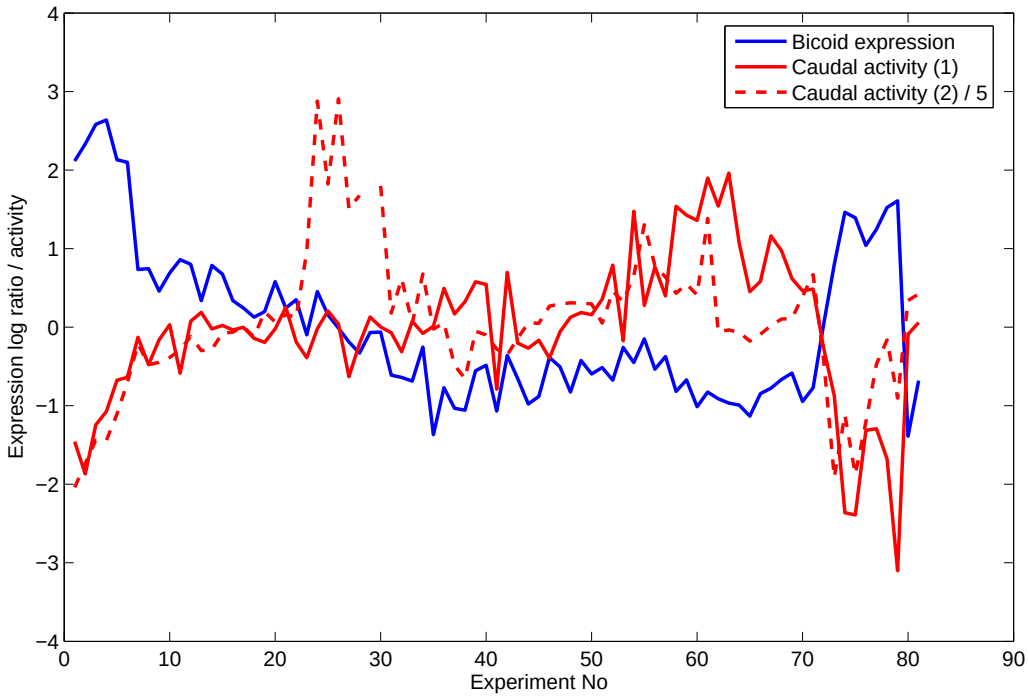

Figure 6.19: Measured expression values of Bicoid and predicted activities of Caudal considering genes with multiple binding sites as targets (1) or genes with binding sites conserved in multiple species (2).

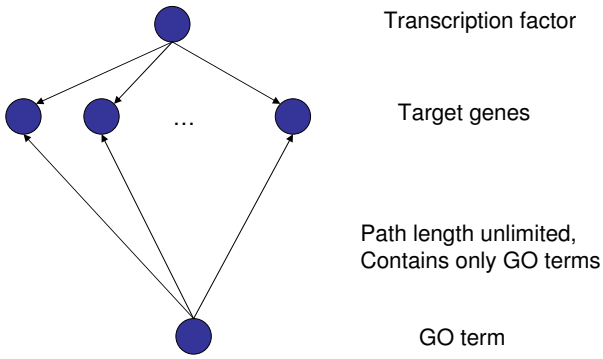

Figure 6.20: A *pathway query* that contains a transcription factor and all its targets sharing a common GO annotation. Using this *pathway query* one can identify regulatory effects of transcription factors on certain biological processes.

expect that a transcription factor can up-regulate genes from one biological process, while its targets from another biological process are not regulated or even down-regulated due to the activity of some other transcription factor or another regulatory mechanism that is not mediated by transcription factors, like chromatin remodeling. Thus, we defined a new *pathway query* that contains a transcription factor and all its targets from a common biological process according to our GO annotations. If a gene is annotated with a GO term, it is not necessarily annotated with all parent GO terms. As we would like to include such implied annotations, we build a graph by combining the GO hierarchy with annotated genes and the transcription factors with their targets such that each gene is connected to all GO terms it is annotated with, and each transcription factor is connected to all its target genes. The GO terms are connected as in the GO hierarchy. The *pathway query* that then solves our task by allowing unlimited paths from GO terms to target genes is depicted in Figure 6.20. The resulting activity scores can be interpreted as a summary of the expression values of a transcription factor's targets within a GO class.

Before computing the activities of transcription factors, we checked if the transcription factor's predicted targets are overrepresented in certain GO classes. For instance, we expect that genes known to be involved in development (from the GO class development) are enriched in targets of all five studied transcription factors. We computed a p-value for that enrichment, the values are listed in table 6.7. Except for Krueppel, the p-values are very significant, implying that our predicted transcription factor targets are indeed enriched in the GO class development.

Next, scores were computed for all transcription factors in combination with each GO class in the subtree below the term development. The results for the two different strategies for target identification are depicted in Figures 6.21 and 6.22, respectively. Again,

Transcription factor	P-value
Caudal	1.1×10^{-23}
Bicoid	4.8×10^{-13}
Knirps	5.9×10^{-7}
Krueppel	2.2×10^{-4}
Hunchback	3.5×10^{-27}

Table 6.7: Enrichment of transcription factor targets in the GO class development.

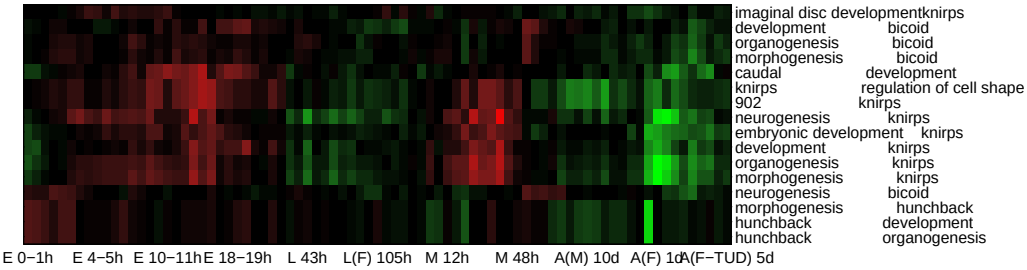

Figure 6.21: Predicted transcription factor activities during the Drosophila life-cycle in the GO subtree below the term development. Genes with multiple binding sites are considered targets of a transcription factor.

the scores are much more significant when information from different species is used to find conserved binding sites, but the general pattern is similar. Most GO classes are up-regulated in the late embryo and again in the late pupa independent of the transcription factor.

6.2.5 Discussion

There is one general problem with the transcription factors that we chose for our investigation: The transcription factors, for which we computed activity scores, have been studied extensively in the biological literature. Most of these studies deal with quite detailed local effects of the transcription factors, for instance the establishment of concentration gradients of mRNAs or proteins in the early embryo. Such local effects cannot be discovered by our method as the mRNA hybridizations were made from complete organisms.

As the function of the investigated transcription factors is closely related, we also do not observe large differences between the different transcription factors. Instead, the pattern of activity is similar for all transcription factors and is characterized by down-regulation of the targets in the early embryo and in the adult female fly. Only the Bicoid targets are not down-regulated in the embryonic stage.

The quality of the predicted potential transcription factor targets remains one of the main issues. The number of binding sites that we found using the multiple alignments seems unrealistically high. For Caudal, for instance, we found 2823 potential targets in the whole genome. From approximately 5000 genes for which expression data are available, almost

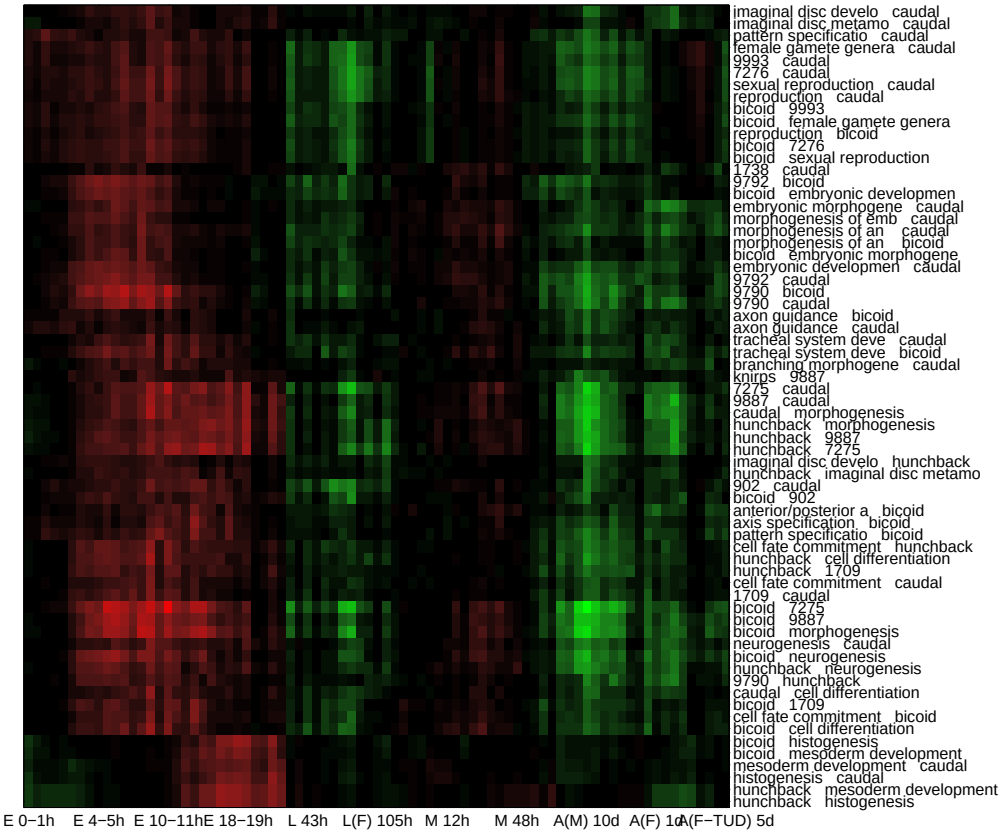

Figure 6.22: Predicted transcription factor activities during the Drosophila life-cycle in the GO subtree below the term development. Genes with binding sites conserved in multiple species are considered targets of a transcription factor.

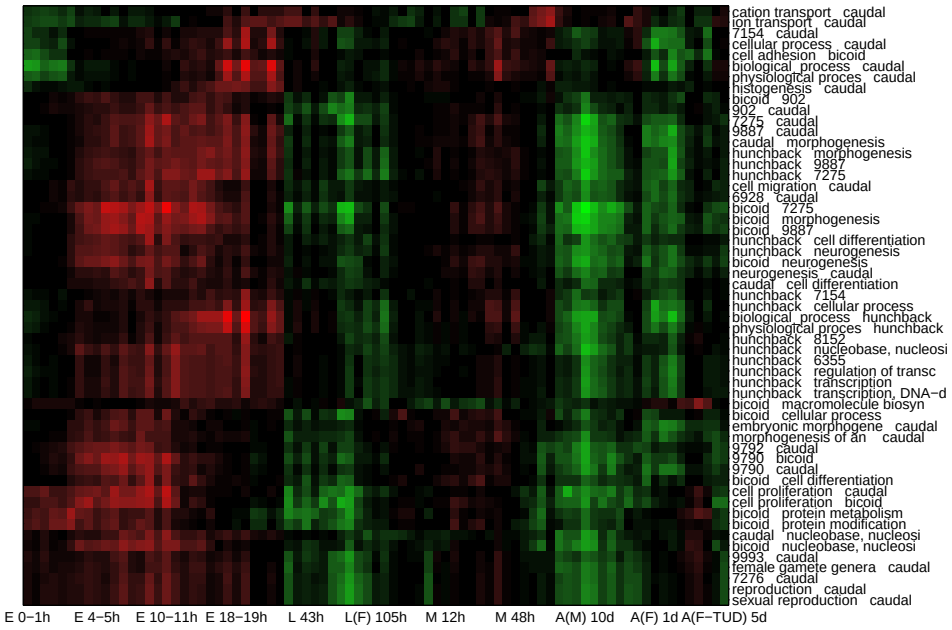

Figure 6.23: Predicted transcription factor activities during the Drosophila life-cycle in the GO category biological process. Genes with binding sites conserved in multiple species are considered targets of a transcription factor. Only GO classes with at least one score greater than six were considered.

1000 are predicted targets of Caudal. This does not appear plausible from a biological point of view. Still, the strong statistical signal picked up with these targets, such as their over-representation in the development class and the correlation of the estimated activity with bicoid expression, suggests that the predicted targets are at least strongly enriched with real targets of Caudal. A more detailed analysis of the predicted target set in terms of their expression data could be helpful to improve the target prediction.

6.3 Analysis of osteoarthritis data

6.3.1 Disease models for osteoarthritis

In-vitro disease models are very important for pharmaceutical research because only with such models several screening tasks can be performed efficiently. Target verification, for instance, may require that the effect of inhibiting a target protein is verified. For osteoarthritis, it is important to understand the effect of target proteins on certain collagenases because those are claimed to be largely responsible for cartilage degradation. Thus, a target validation could include measuring the amount of secreted collagenase with or without inhibition of a candidate drug target in a model system that is similar to the actual disease.

In a cooperation with Aventis, several model systems were established and their gene expression profile was measured with specially designed microarrays. These profiles were compared to actual human chondrocytes with certain stimulations or osteoarthritis. The studied systems consist of two cell lines, SW1353 and HCS-2/8, stimulated with IL-1β and BMP-4, respectively, primary human chondrocytes, human cartilage explants, and arthritic chondrocytes. A first impression about the similarity of the model systems can be gained by looking at the principal component analysis of the expression profiles depicted in Figure 6.24. The expression values used here are the ratios of the stimulated or diseased cells as compared to the unstimulated or healthy cells, respectively. Clearly, the data from the arthritic chondrocytes differ most from all other systems, implying that the similarity between model systems and the actual disease is not large in terms of gene expression. Therefore, these systems should be used with care and probably only for certain aspects of the disease.

The SW1353 cell line was studied in more detail, also with respect to relevant transcription factors, using the scoring methods described above. The goal of the study was to find out if SW1353 cells react to stimulation with IL-1β in a similar way as primary human chondrocytes. The question if such a stimulation can serve as a disease model was not addressed. The results are summarized in the following.

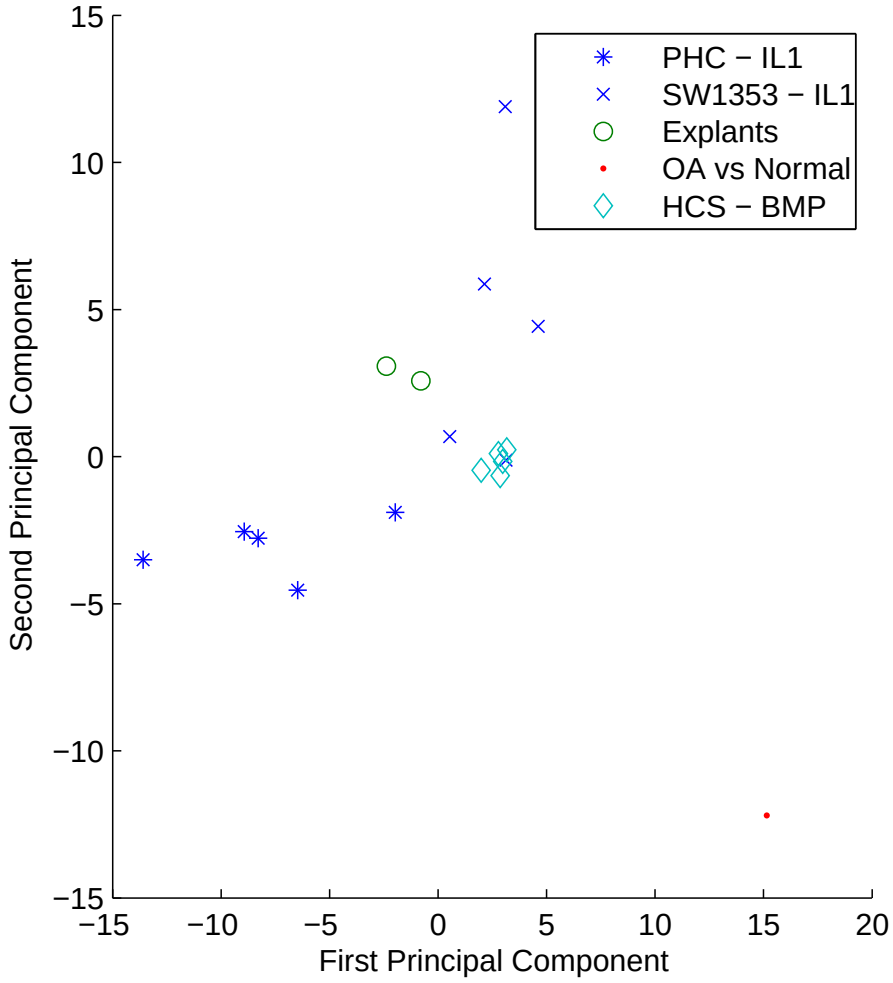

Figure 6.24: Principal component analysis of several model systems for osteoarthritis and arthritic human chondrocytes. This analysis implies that none of the model systems is indeed similar to the diseased tissue in terms of gene expression.

6.3.2 SW1353 cells as a model for catabolic processes in chondrocytes

A study was conducted by cooperation partners which investigated the gene expression profile of the chondrosarcoma-derived cell line SW1353 in order to validate it as an in vitro model for primary human (adult articular) chondrocytes (PHCs) (Gebauer et al., 2005). Time series measurements after stimulation with IL-1β were collected using specifically designed SensiChip DNA microarrays.

Experimental setup

The SensiChip technology (Qiagen, Zeptosens) is a two-color microarray platform which was designed for high sensitivity. The arrays were spotted in duplicates with 70-mer oligonucleotides representing the 3'-UTR of 312 housekeeping and human cartilage relevant genes. Every single gene was represented by one 70-mer oligonucleotide. The SW1353 time course study consisted of 3 independent culture series of SW1353 monolayers treated with 1 ng /ml IL-1β (Roche Diagnostics, Germany) in DMEM/F12 (Gibco BRL, Germany) containing 0.5% lactalbumin enzymatic hydrolysate (Sigma, Germany) or control medium (DMEM/F12 / 0.5% lactalbumin enzymatic hydrolysate) for 30 min, 6 hrs, 16 hrs, 24 hrs and 48 hrs. From each of the 30 cultures RNA was isolated using the RNeasy Kit (Qiagen). 1 g total RNA from IL-1β-treated and control cultures was reverse transcribed in the presence of Alexa-labeled or Cy5-labeled dUTP nucleotides using the Omniscript Kit (Qiagen) and generated cDNA was subsequently purified using the QIAquick Kit (Qiagen). Due to limited amounts of RNA starting material, 250 ng total RNA from IL-1β-stimulated PHCs and unstimulated controls from each time point (30 min, 6 hrs, 16 hrs, 24 hrs and 48 hrs) was amplified and thereby labeled with Cy3-UTP and Cy5-UTP respectively (Amersham Pharmacia) using the MessageAmp aRNA Kit (Ambion). After cRNA clean-up using the RNeasy kit (Qiagen), 5 g of Cy3-labeled cRNA from IL-1β-stimulated chondrocytes were mixed with 5 g of Cy5-labeled cRNA from the respective unstimulated control. cRNA was fragmented by incubation with 40 mM TRIS-acetate, pH 8.1, 100 mM KOAc, 30 mM MgOAc for 15 min at 95C and desalted using a Microcon YM-10 concentrator (Millipore). 600 ng of either mixed Cy-dye labeled cRNA or purified cDNA sample was hybridized for 16 hrs on a SensiChip microarray (Qiagen, Zeptosens). Hybridization was repeated with inversely labeled material generated by exchanging Alexa- and Cy5-labeled dUTP nucleotides for IL-1β-treated and control sample. Inverse labeling was performed to compensate for differential labeling efficiency and fluorescence intensity associated with the two dyes.

Analysis of transcription factor activity in SW1353 cells

In order to analyze transcription factor activity from the expression data, groups of potentially regulated target genes were determined for each transcription factor. The required knowledge on transcription factors and their target genes is partially contained in databases

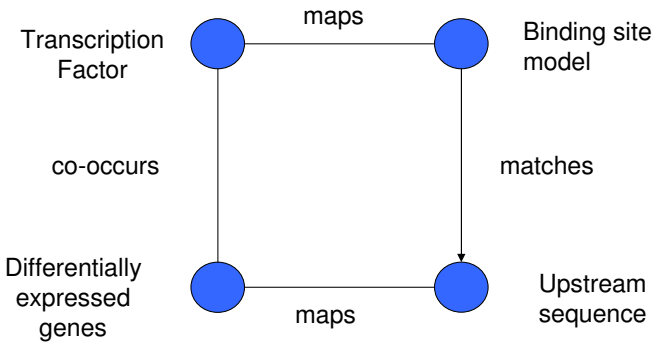

Figure 6.25: A *pathway query* for identifying potential transcription factor targets combining different network sources.

like TRANSFAC (Wingender et al., 2000) and the scientific literature. In this study, a combination of three types of evidence for the regulation of a target gene by the binding of a transcription factor to its promoter was used: predicted binding sites, known binding sites, and co-occurrences of transcription factors and target genes extracted from the scientific literature. Binding site predictions for target genes were computed with the TRANSFAC tool MATCH on sequences starting 3 kb upstream from the transcription start site and ending 1 kb downstream. Known binding sites were extracted from TRANSFAC's SITE table (Wingender et al., 2000). Co-occurrences in the literature were computed using the text-mining tool ProMiner (Hanisch et al., 2003). A gene was added to the group of potential target genes of a transcription factor (TF) if the gene contained a TRANSFAC binding site for TF or if the gene contained both a predicted binding site of TF and a literature co-occurrence with TF. This was achieved using the *pathway query* depicted in Figure 6.25. Table 6.8 shows the evidence for the predicted target genes of RelA and c-REL. For most of the targets additional literature evidence could be found easily. Only for some pairs, where there are well-studied other relationships (e.g. interactions) between the proteins, the number of documents with co-occurrences that was so large that it was not possible to read all abstracts in order to verify the predicted transcription factor-target relationship.

Target gene groups were checked for enrichment of significantly regulated genes (p-value < 0.01 for at least one time point) in either cell type. This enrichment was quantified by p-values for all the TF target gene groups using Fisher's exact test; the corresponding "group p-value" is the probability of finding the observed number of significantly regulated genes in the target gene group by selecting genes randomly. A group p-value lower than 0.05 was considered as significant.

The analysis of target gene groups showed IL-1β-mediated induction of five common target genes by the transcription factor NFκB in both cell types. Both Rel proteins, c-Rel

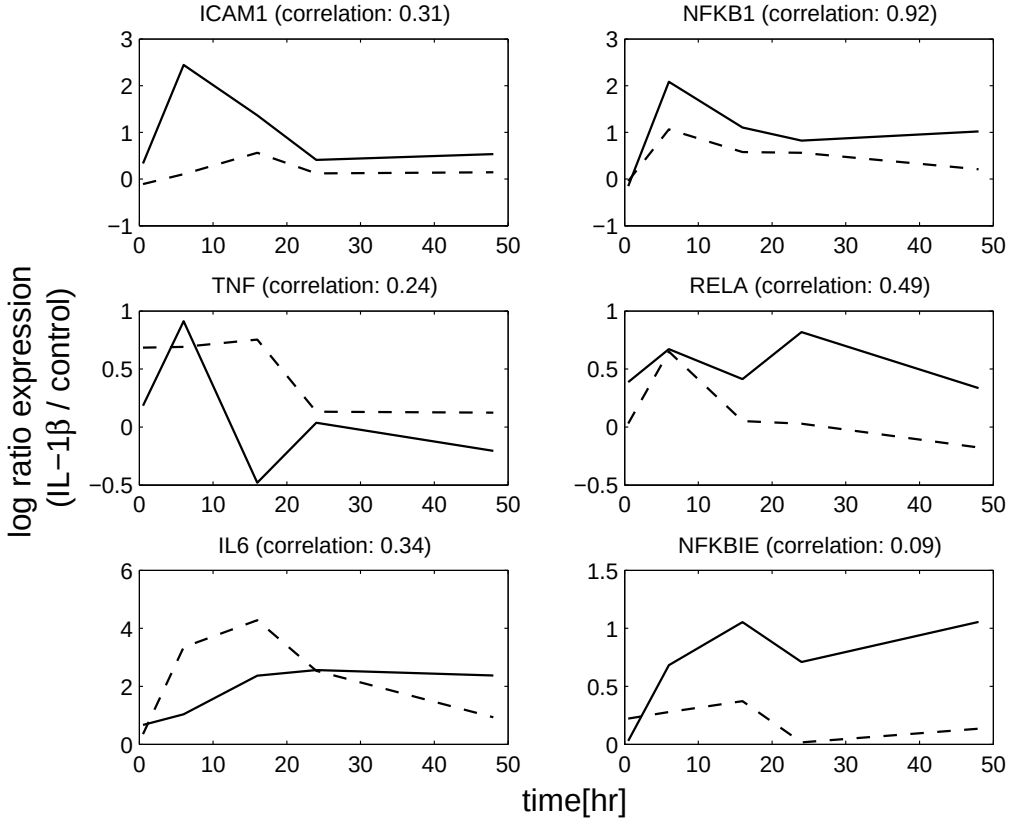

Figure 6.26: Expression profiles of genes in the predicted NFκB regulation context (the only regulation context common to both systems according to our analysis) as log-ratios of expression levels between Il-1β stimulated cells vs. untreated cells against time in hours after the treatment. Profiles for SW1353 cells are shown as dashed lines, profiles for PHCs as solid lines. Correlation values quantify the similarity between the two systems for each gene. Statistical analysis showed that the correlation of the NFκB targets is significantly higher than expected by chance (details are given in the text).

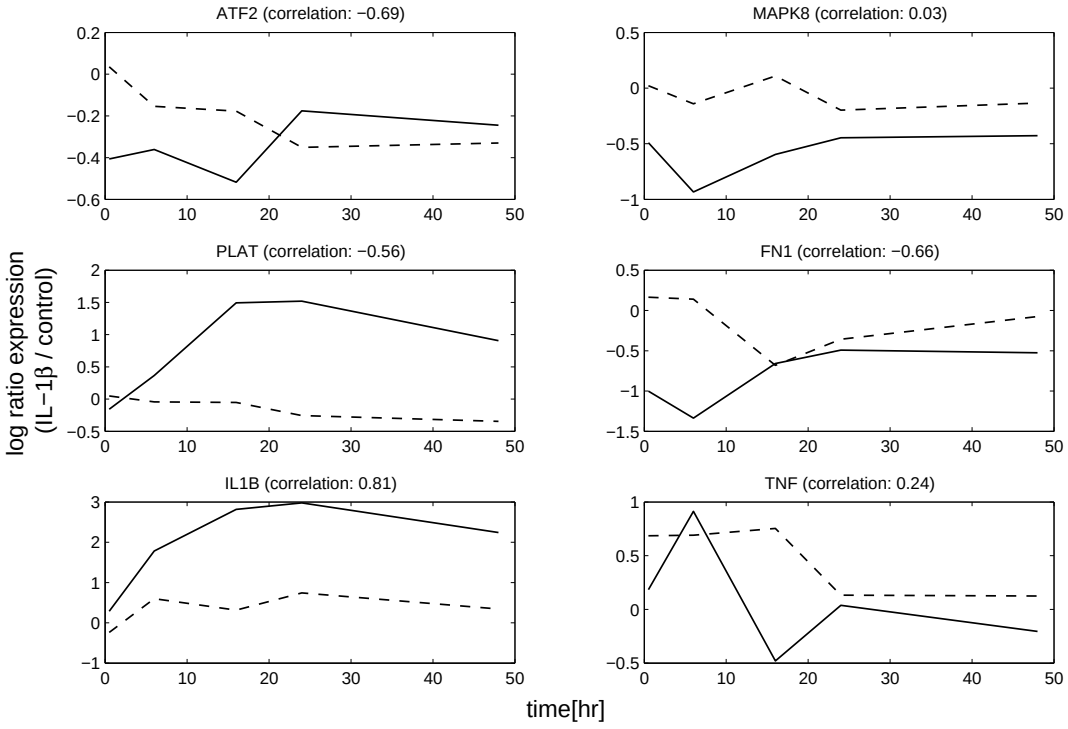

Figure 6.27: Expression profiles of genes in the predicted ATF2 regulation context as log-ratios of expression levels between Il-1β stimulated cells vs. untreated cells against time in hours after the treatment. Profiles for SW1353 cells are shown as dashed lines, profiles for PHCs as solid lines. Correlation values quantify the similarity between the two systems for each gene. Statistical analysis showed that the correlation of the ATF2 targets is not significant (details are given in the text).

Transcription Factor	Target Gene	Predicted Binding Site	Transfac Binding Site	Literature Co-Occurrence	Literature Reference
RelA	IL6		+	+	Catron et al. (1998); Kawashima et al. (2001)
	ICAM1		+	+	Catron et al. (1998)
	NFKB1	+		+	-
	TNF		+	+	Xu et al. (2001)
c-Rel	IL6	+		+	Civil et al. (1999)
	ICAM1	+		+	Spiecker et al. (2000)
	NFKB1	+		+	-
	NFKBIE	+		+	-

Table 6.8: Predicted targets of RelA and c-Rel, the evidence used for the prediction and supporting literature references that were found by browsing through the list of co-occurrences (automatically generated from text mining) or the references in the TRANSFAC database.

(REL) and RelA (RELA), had significant group p-values below 0.05 for PHCs and for SW1353 cells. Their predicted group of target genes (represented on the microarrays) was IL-6, TNF, NFKBIE (NFκB inhibitor epsilon), ICAM1 and NFKB1 (table 4). Figure 6.26 shows the expression profiles of RelA (c-Rel was not represented on the microarrays) and the predicted target genes of RelA and c-Rel, which were similar in the two cell types. The inter-cellular similarity for each gene of the NFκB target gene group and RelA was measured by Pearson correlation coefficients (Fig. 6.26) of the log-ratio expression levels. The overall similarity in the gene group is higher than expected by chance. Using the rank sum test to compare the inter-cellular correlations (of the five genes) with all other genes, significance at the 5% level was shown, i.e. the probability of picking five genes at random that are similarly correlated between the two cellular model systems is below 5%. NFKBIE is contained only in the predicted NFκB gene target group of PHCs cells but not in the NFκB target group for SW1353 cells. NFKBIE regulation distinguishes the activity of the transcription factor NFκB in the two cell types. Consistent with this finding, NFKBIE showed no correlation of expression levels between the measurements. For SW1353 cells, two other significant transcription factors were RUNX2 (regulating MMP-13, SMAD 2, MYC, RUNX1 and RUNX2 itself) with a group p-value of 0.05 and RARA (regulating BGLAP, ICAM1 and MYC) with a group p-value of 0.04. A second relevant transcription factor for PHCs was ATF2 (regulating MAPK8, IL-1β, FN1, PLAT and TNF) with a group p-value of 0.02. Figure 6.27 shows the expression profiles of the predicted ATF2 target genes. Except for IL-1β, we did not observe any correlations between the two cell

types. In fact, ATF2, PLAT and FN1 were anti-correlated, confirming the different roles of ATF2 in the two cell lines.

6.3.3 Analysis of patient data for osteoarthritis

Another dataset from the osteoarthritis project was collected from patient samples. Cartilage samples were obtained from patients suffering from OA and compared to healthy tissue on the gene expression level using Affymetrix GeneChip microarrays (HG-U133 plus 2.0). The data set contains measurements from 13 healthy and 13 diseased samples. Ratios and p-values for differential expression between the two groups were calculated by Katrin Fundel using SAM (Tusher et al., 2001). Only probe sets with a present call in at least 80% of all samples in one group were included in the computation. Affymetrix probe set identifiers were mapped to the synonym list for ProMiner with the help of Affymetrix annotations, resulting in 10277 synonym list entries (i.e. proteins) with associated expression data.

Significant areas

In order to get an overview of functionally coherent sets of differentially expressed proteins, we conducted significant area searches on a co-occurrence network that was generated using ProMiner on the Medline database with abstracts since 1990. A co-occurrence edge between two proteins is introduced when there are at least 5 co-occurrences of the proteins within sentences. The resulting network contains 9763 proteins and 81330 edges (proteins without any neighbors are not added to the network). 5209 of the proteins in the network have associated expression data.

We performed three different significant area searches. The first one based on the p-values, the other two based on fold changes, one searching for up-regulated sub-networks, and one for down-regulated networks. Each of the searches resulted only in one significant area with more than 5 proteins and a number of very small ones. The large significant areas for the three searches are depicted in Figures 6.29 and 6.28.

First, we will discuss the significant areas found by the original method based on p-values (Hanisch, 2004). It contains a cluster of highly up-regulated proteins from the extracellular matrix. Among these proteins are different collagens and other matrix constituents like versican (CSPG2), fibrillin (FBN1), tenascin (TNC), and thrombospondin (THBS2). There is also one protein involved in collagen catabolism, namely ADAMTS2. A second functionally coherent cluster contains proteins that play a role in actin filament organization: ARPC2, ARPC5, WASL, ACTR2, DOC1, CRK, and NCK1. Exploring the neighboring of these actin-related genes, we discovered an interesting regulatory context: protein kinase c (PKC) is known to play a major role in the regulation of the actin cytoskeleton (Larsson, 2006). It can bind to integrins, modify integrin-mediated signaling and phosphorylate the MARCKS protein and related proteins. This phosphorylation is supposed to influence the cytoskeleton by a translocation of the phosphorylated proteins from the membrane and a direct effect on the actin filaments. The PKC targets MARCK,

GO class	P-value
cell proliferation	5.45e-9
regulation of cell proliferation	1.26e-7
negative regulation of cell proliferation	1.09e-6
negative regulation of progression through cell cycle	1.99e-6
cell communication	2.64e-6
regulation of actin filament polymerization	2.88e-6
actin polymerization and/or depolymerization	2.98e-6
phosphate transport	3.29e-6
skeletal development	3.51e-6
actin filament polymerization	9.82e-6
cell adhesion	2.09e-5
regulation of biological process	3.27e-5
regulation of cellular process	3.92e-5
regulation of protein kinase activity	4.20e-5
regulation of transferase activity	4.29e-5
regulation of actin polymerization and/or depolymerization	4.73e-5
regulation of actin filament length	5.26e-5
actin cytoskeleton organization and biogenesis	5.87e-5
inorganic anion transport	5.90e-5
regulation of cellular physiological process	8.09e-5
actin filament-based process	8.13e-5

Table 6.9: Significant GO classes in the significant area based on p-values. The detected classes can be divided up into several groups: cell proliferation and cell cycle, actin filament organization, cell communication and cell adhesion, and some unspecific terms. The groups correspond to the protein clusters in the significant area described in the text.

MacMARCKS, GAP43 and CAP23 are all strongly up-regulated in osteoarthritis according to our data. For PKC, we only observe a weak up-regulation for the isoforms delta and zeta. PKC change transcription through NFκB and CREB. Thus, PKC is an interesting candidate for pathway cross-talk, connecting some of the observed regulatory effects.

Finally, there are some proteins involved in regulation of cell cycle and cell proliferation clustered around the cyclin-dependent kinase inhibitor CDKN1A. This group contains down-regulated proteins such as CDKN1A, CDKN2A, DDIT3, RAF1, GADD45B, and ING1, as well as the up-regulated proteins SAS, BAX, PCNA, and S100A11. We have analyzed the network for over-represented GO classes in the biological process hierarchy as described in Hanisch (2004), and listed all GO classes with a p-value less than 1e-4 and at least three genes in the network in Table 6.9. These significant GO classes coincide with the functions of the described clusters.

The up-regulated sub-network that was found contains two dense clusters, the first one consists of several class 2 major histocompatibility complex proteins. These proteins are responsible for presenting antigens to T-helper-cells. They are known to be involved in

inflammatory processes, and some of them have been associated with osteoarthritis before (Lance et al., 1993). Their expression has even been proposed to predispose inflammatory diseases like rheumatoid arthritis (Friese et al., 2005). Another study shows an association between two HLA-DR alleles and distal interphalangeal osteoarthritis (Riyazi et al., 2003). A review about associations between rheumatoid arthritis and MHC genes is presented by Reveille (2005).

The second cluster is similar to the extracellular matrix cluster found in the p-value based significant area. It contains different collagens as well as other extracellular matrix proteins. Most of these proteins are either involved in proteolysis, like the collagenase 3 (MMP13), ADAMTS2, and FAP, or in cell-cell and cell-matrix adhesion, like TNC, DPT, EDIL3, and POSTN. The proteolysis part of the cluster was not observed in the p-value based significant area, and adds an important biological context, as proteolysis is believed to be largely responsible for the destruction of cartilage tissue, and especially MMP13 has received much attention in osteoarthritis research (Vincenti and Brinckerhoff, 2002; Malemud et al., 2003; Martel-Pelletier et al., 2001). The adhesion proteins could play in role in the formation of chondrocyte clusters that has been observed in osteoarthritic cartilage (Poole, 1997).

The down-regulated significant area does not have a very clear structure. In addition to many cell cycle related proteins that are largely present in the p-value based significant area as well, the most conspicuous feature is a group of three cytochrome P-450 proteins. The P-450 complex is involved in detoxification and degradation of many drugs in the liver. Not much has been published about P-450 in the context of arthritis. Dulos et al. (2005) observe increased expression of the P-450 protein in fibroblast-like synoviocytes in patients with rheumatoid arthritis, but the specific enzymes that show up in the significant area are not mentioned. The GO analysis of the down-regulated significant area does not show any new classes either, except the term immune response, which shows up because of complement component 3 (C3), IL6R, and some other proteins that are not connected within the network.

In summary, the significant area searches in combination with GO analyses point our attention to a couple of processes, which are mostly known to be relevant for arthritic diseases. The ToPNet user interface allows to dive into the details by providing links to many gene and protein databases and to journal articles that might elucidate connections between the different processes. Nevertheless, the information we get at this point is somewhat vague and not sufficient to derive detailed hypotheses about disease mechanisms.

Signaling pathways and transcription factors in Osteoarthritis

In order to achieve detailed interpretable results, the data should be analyzed on well curated networks. Thus, we merged the information from the TRANSPATH and TRANSFAC databases into a single network and defined a *pathway query* as depicted in Figure 6.30, describing short pathways from a receptor via transcription factors to a set of target genes. As scoring function we used the rank score on the target genes. The purpose of this query was to identify signaling pathways that are involved in OA. Figure 6.31 shows the

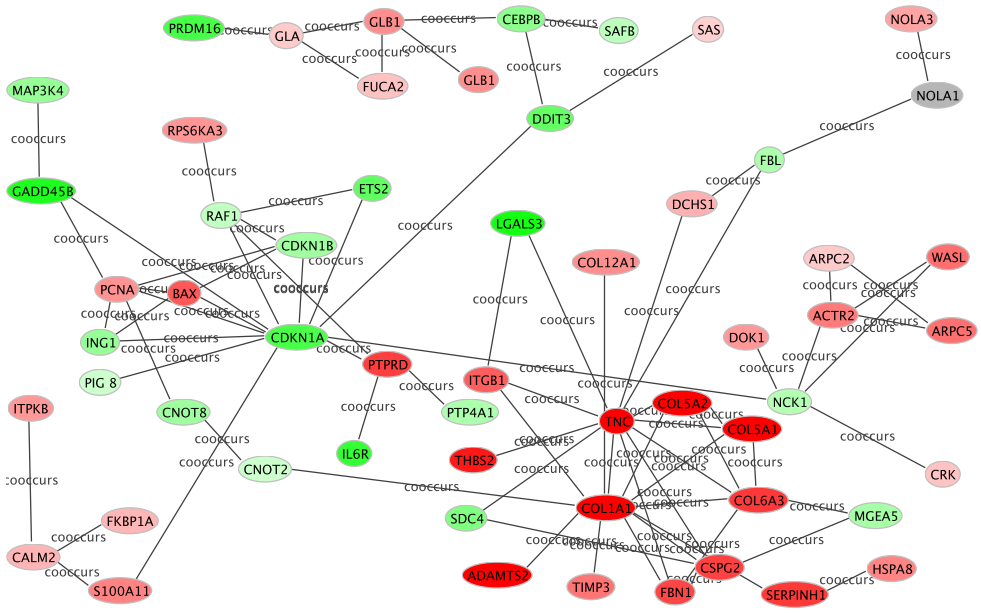

Figure 6.28: Significant area found using the original method based on p-values. The search was performed on a co-occurrence network generated by ProMiner. The significant area contains some clearly separated clusters corresponding to different functional contexts.

two most significant instances (from a total of 22 instances) for the query (p-values 0.032 and 0.033). The two instances overlap to a large extend because both connect a receptor via CREBBP to several transcription factors. The instances represent only short pathways as the connection between receptor and transcription factor was restricted to length two in the query. Allowing longer paths without further restrictions leads to unspecific instances containing many genes and receiving bad scores. Already with path length three no instance with a p-value less than 0.05 can be found. Like PKC, CREBBP is an interesting candidate for connecting some of the observed transcriptional regulations observed in osteoarthritis because CREBBP can interact with RelA, c-Jun, and CREB1. Furthermore, CREBBP participates in multiple pathways: Calcium signaling pathway, Wnt signaling pathway, Notch signaling pathway, TGF-beta signaling pathway, according to KEGG.

Next, KEGG pathways were systematically tested using the Mann-Whitney-Wilcoxon test on the fold change data, as described in 6.1.1. Figure 6.32 shows the distribution of p-values from these tests by plotting the sorted p-value list. If fold change values were randomly distributed across pathways, one would expect approximately a diagonal line in that plot since p-values are uniformly distributed when the null hypothesis holds. The plot shows that there are about 20 pathways with very low p-values. After these first 20 values, the distribution follows the expected behavior of a diagonal line. This suggests that there are about 20 true positive instances of pathways with fold change values that non-randomly high or low. Table 6.10 shows the best scoring 20 pathways. The highest scoring pathway, ECM-receptor interaction, is not a classical pathway. It lists proteins from the extracellular matrix and their receptors (mostly different integrins).

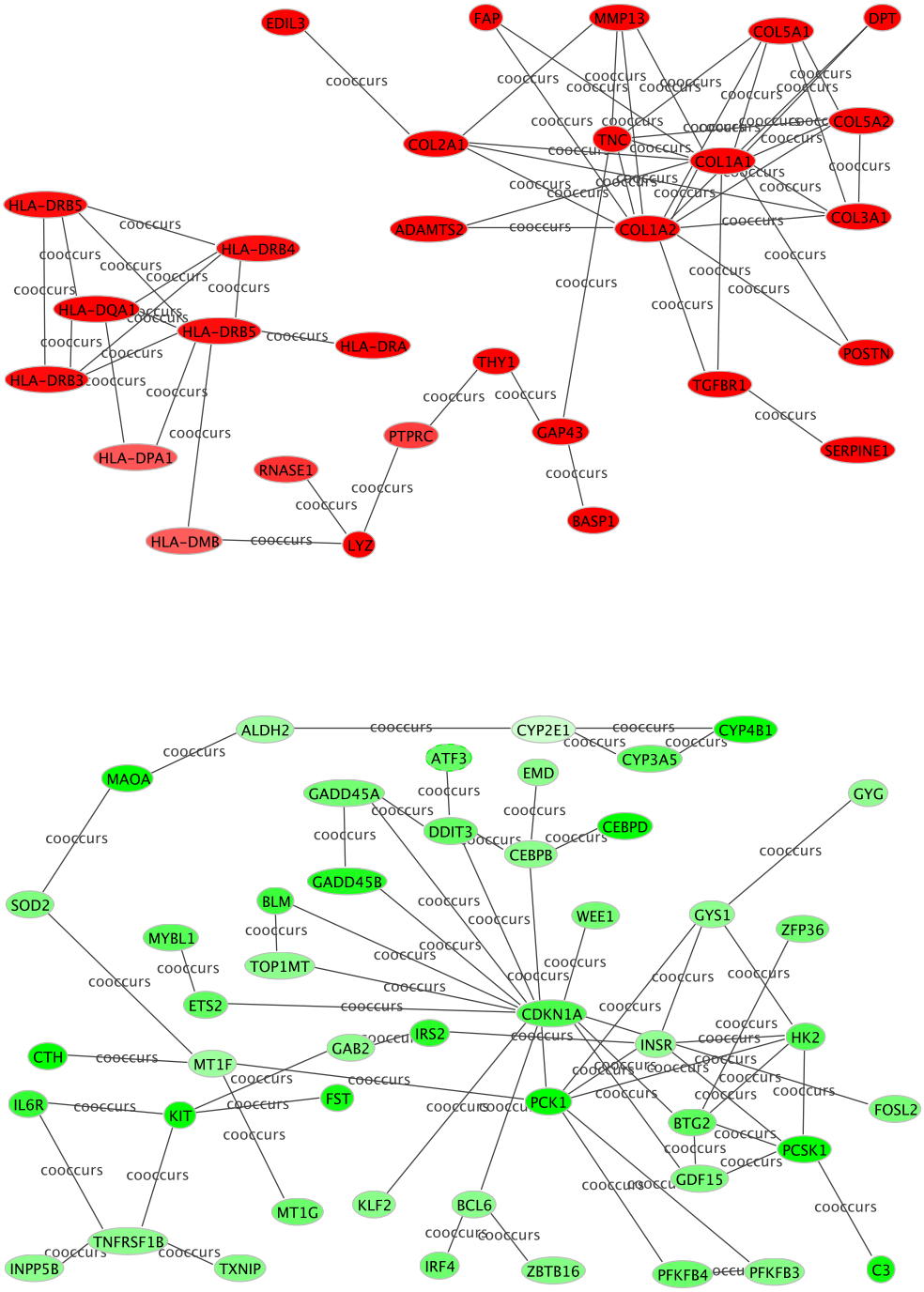

Figure 6.29: Significant areas found using the method based on fold changes. The up-regulated significant area (left) contains two tightly connected clusters. One of these clusters contains proteins from the major histocompatibility complex, class 2, the second one contains mainly proteins that are expressed in the extracellular matrix and play a role in cartilage turnover.

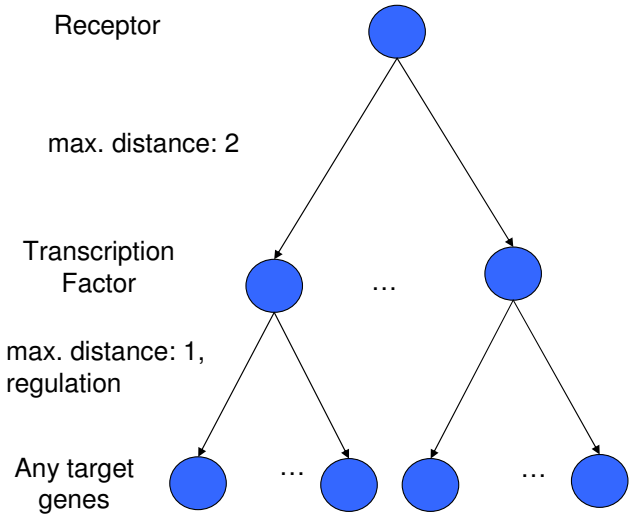

Figure 6.30: A *pathway query* to find signaling pathways relevant for osteoarthritis. This *pathway query* is performed on a graph containing information from TRANSFAC and TRANSPATH. Therefore, results should correspond to well characterized pathways.

Therefore it summarizes the proteins that are necessary for a communication between the extracellular matrix and the chondrocytes. The up-regulation of many matrix proteins in osteoarthritis is well known, and also the interplay of matrix-proteins like fibronectin with integrins has be discussed as a disease mechanism in OA (Peters et al., 2002). Integrins are also supposed to play a role in the transduction of signals from mechanical forces (Millward-Sadler and Salter, 2004).

Finally, we also looked at single transcription factors and scored them using the rank score either on the p-values or on the ratios. Potential targets of a transcription factor were defined as all genes that have a binding site for that transcription factor according to TRANSFAC. Here, we relied on annotated binding sites only and did not try to predict transcription factor targets. The ten best scoring transcription factors for each of these scoring methods are listed in Table 6.11, and Figure 6.33 shows the merged graphs of all relevant transcription factors (p-value < 0.05) and their target genes using both p-values and fold changes for the rank score.

The overlap between the transcription factors identified by the two methods is rather small, consisting of RFX1 and JUND only. Nevertheless, the processes that the identified transcription factors and their target genes are involved in, are similar, as can be seen in the graph containing the target genes (Figure 6.33). Firstly, the Rfx transcription factors regulate class 2 major histocompatibility complex genes, which have been observed in the significant areas already. Their possible role in osteoarthritis was discussed in section 6.3.3.

Another relevant process is the regulation of cell cycle and proliferation that could already be identified with the help of the significant area search algorithm as well. In our

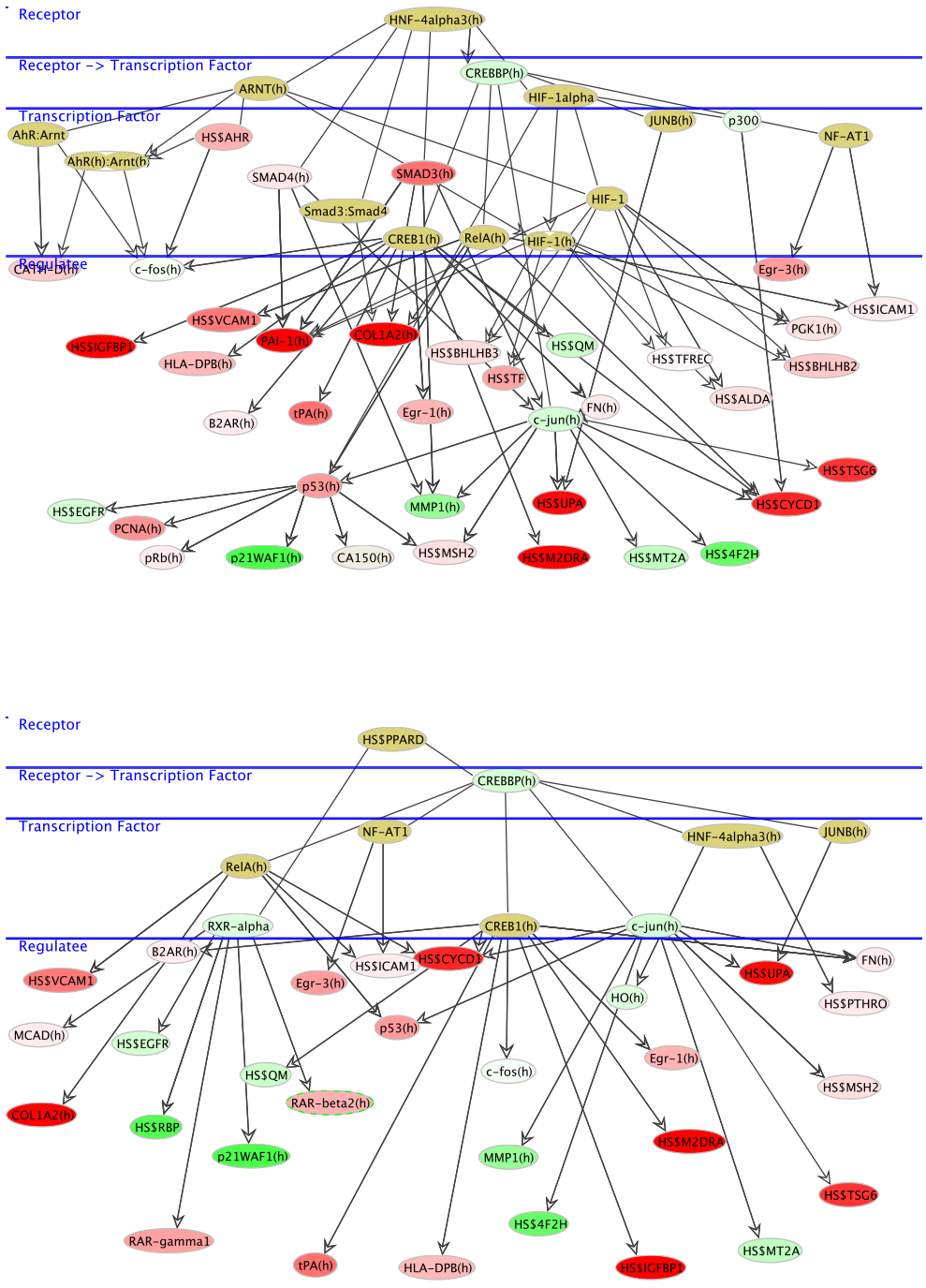

Figure 6.31: Most significant instances of the *pathway query* from Figure 6.30. The instances overlap to a large extend. Both contain the transcription factors CREB1, RELA, c-JUN, NF-AT1 and their targets.

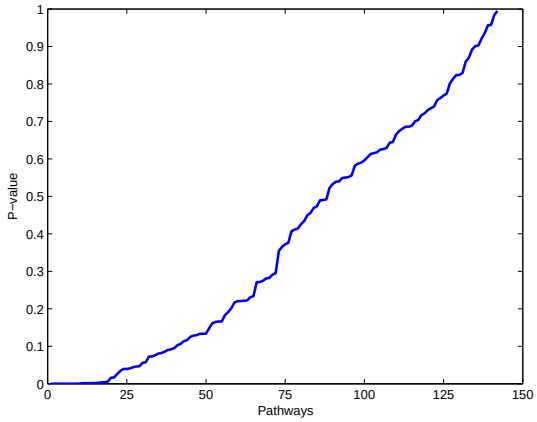

Figure 6.32: P-values for KEGG pathways using a rank test on osteoarthritis patient data. After the top 20-25 pathways, the plot approximates a straight line, suggesting that there are about 20-25 true positives.

Pathway	**P-value**	**Direction**
ECM-receptor interaction	2.28E-12	up
Focal adhesion	1.53E-09	up
Regulation of actin cytoskeleton	3.91E-06	up
Prion disease	2.49E-05	up
Cell adhesion molecules (CAMs) based on ligands classification	5.80E-05	up
Cytokine-cytokine receptor interaction	8.69E-05	up
Neuroactive ligand-receptor interaction	9.10E-05	up
Complement and coagulation cascades	1.10E-04	up
Wnt signaling pathway	2.19E-04	up
TGF-beta signaling pathway	4.22E-04	up
Valine, leucine and isoleucine degradation	1.45E-03	down
Gap junction	1.56E-03	up
Propanoate metabolism	1.61E-03	down
MAPK signaling pathway	1.75E-03	up
Fatty acid metabolism	1.82E-03	down
Glycosphingolipid metabolism	2.76E-03	up
Glycosaminoglycan degradation	3.61E-03	up
Axon guidance	4.33E-03	up
Chondroitin Heparan sulfate biosynthesis	5.36E-03	up
Inositol phosphate metabolism	1.59E-02	down

Table 6.10: Most significant KEGG pathways with respect to osteoarthritis patient expression data as computed using the Whitney-Mann-Wilcoxon test.

Transcription factor	Score	Score based on
STAT5B	0.020	p-values
RFX1	0.028	p-values
ATF1	0.029	p-values
JUN	0.029	p-values
HMGA1	0.030	p-values
JUND	0.034	p-values
CREM	0.045	p-values
TA p63 α	0.045	p-values
RXRA	0.048	p-values
EP300	0.049	p-values
RELA	0.0046	fold changes
RFX1	0.0069	fold changes
AP2α	0.012	fold changes
JUND	0.017	fold changes
NFBK1	0.018	fold changes
CREB1	0.018	fold changes
RFX3	0.020	fold changes
RFX2	0.020	fold changes
RFX5:RFXAP:RFXANK	0.020	fold changes
STAT3	0.036	fold changes

Table 6.11: Best scoring transcription factors in osteoarthritis data. Scores are rank scores either based on fold changes or p-values for differential expression.

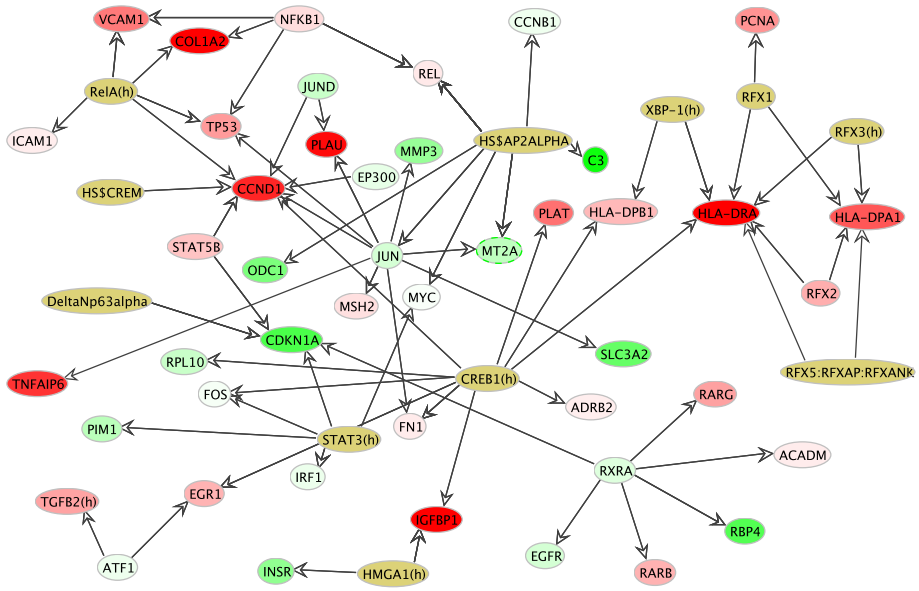

Figure 6.33: All identified relevant transcription factors and their target genes in osteoarthritis. Relevant transcription factors were identified using the rank score based on p-values for differential expression or fold change values. All transcription factors with a p-value better than 0.05 are included in the graph. Node color corresponds to fold change values.

evaluation of transcription factors we find CCDN1 (cyclin D1), which induces cell-cycle progression from S1 to G phase, up-regulated and CDKN1A (cyclin dependent kinase inhibitor 1A), which inhibits the cell-cycle progression, down-regulated. Therefore, we could expect a faster progression through the cell cycle in arthritic chondrocytes and a higher rate of cell proliferation. This is in agreement with the finding of chondrocyte clusters in osteoarthritic cartilage tissue (Poole, 1997; Pfander et al., 2001; Quintavalla et al., 2005). The regulators of these cell-cycle genes include STAT3 and STAT5B as well as RELA, JUN and JUND. But as we have seen in the significant areas, there are other differentially regulated cell cycle-related genes. Furthermore, Gómez-Camarillo and Kouri (2005) observe a reduced rate of cell division in osteoarthritic tissue, contradicting our hypothesis. Figure 6.34 shows the cell cycle according to the KEGG database with fold changes from the osteoarthritis expression data. In some cases, the nodes from KEGG correspond to multiple genes. In that case, the median expression value was used. The most strongly differentially expressed genes (with the highest fold change values) are the ones observed earlier: CDKN1A, CCND1, and GADD45B. Additionally, WEE1, which is supposed to act as a negative regulator of entry into mitosis, is also down-regulated. This again strengthens the hypothesis that there should be a higher cell proliferation in osteoarthritic chondrocytes. It is also in contradiction to Gómez-Camarillo and Kouri (2005) who hypothesize that the phosphorylation state of the mitosis-promoting factor (MTF) complex, consisting of CDK-1 and Cyclin-B, could be a reason why the chondrocytes do not enter mitosis. WEE1 is

one of the proteins that can phosphorylate the MTF complex. Its down-regulation should promote the transition into mitosis. All in all, the role of disruptions of the cell cycle in osteoarthritis is much debated. The presence of some cell-cycle markers and the finding of chondrocyte clusters in osteoarthritis seems to imply that chondrocytes proliferate faster in diseased than in healthy tissue. In the study of Gómez-Camarillo and Kouri (2005) on the other hand, no proliferation of osteoarthritic chondrocytes could be detected.

Cell-cell adhesion is another process that appears regulated according to the transcription factor data. Genes relevant for this process include VCAM1, ICAM1, PLAU and PLAT, their regulators are NFκB (RELA and NFKB1), JUN, JUND and CREB1.

The transcription factor data generally point to the same processes that have been identified earlier using significant area search. Much fewer differentially expressed genes are accounted for, but on the other hand a hypothesis about the involved transcription factors is provided. When more comprehensive data on transcription factor binding becomes available, we expect that more differentially expressed genes will be represented in the generated hypotheses and that there will be a stronger signal for the relevant transcription factors.

Discussion

Osteoarthritis is a complex disease with a complex phenotype. One important part of that phenotype is the destruction of the extracellular matrix, which is believed to be governed by an imbalance between anabolic and catabolic activities of the chondrocytes. The chondrocytes themselves also show specific phenotypes, e.g. a differentiation towards hypertrophic chondrocytes or fibroblasts. There are also spatial differences in the cartilage phenotype (Sandell and Aigner, 2001).

Taking into account the complexity of the disease, it cannot be expected that a complete picture of the disease processes in osteoarthritis can be gained from gene expression data alone. Nevertheless, with the network-based methods presented here, one can quickly identify processes that appear to be relevant for the disease. Considering the expression data in a proper context can assist in the formulation of biological hypotheses, as could be seen in the discussion of the cell cycle regulation in osteoarthritis. Sometimes, it is also possible to find proteins which could constitute a link between relevant processes, although cross-talk between pathways is in general not well understood. An example for such a link is CREBBP, which could play a role in several arthritis-related pathways.

Still, these results represent only hypotheses and ultimately, their validity can only be tested experimentally.

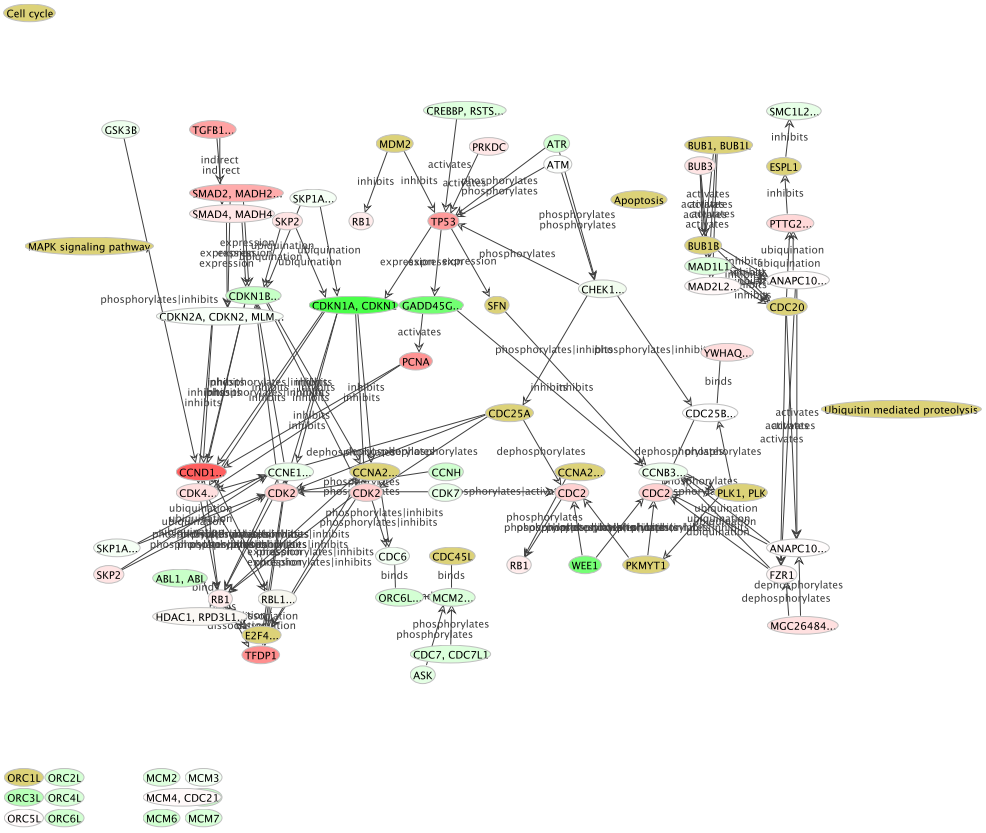

Figure 6.34: Expression of cell cycle genes in osteoarthritis. Many cell cycle related genes are differentially expressed in osteoarthritis. While it is conspicuous that CCND1, which is associated with cell cycle progression, is up-regulated and its antagonist CDKN1A is down-regulated, no clear general pattern emerges.

Chapter 7

Conclusions and Future Work

7.1 Achievements and limitations

Today, expression data are collected routinely on a genome-wide scale. A single Affymetrix GeneChip microarray costs only a few hundred Dollars. There are easy-to-use software tools for simple analyses of microarray data.

The ever-growing biological knowledge is documented in hundreds of scientific journals and in indexing databases for these journals like Pubmed. Some of this knowledge also becomes formalized in databases about genes, proteins, pathways, genotypes and phenotypes, and other biological topics.

Still, in most cases, the researcher performing a microarray analysis is left alone with lists of differentially expressed genes and only his own background knowledge to guide the interpretation. Thus, the integration of high-throughput data with biological knowledge is an important task that can significantly improve the interpretation of the data.

As clustering of samples is performed routinely for the analysis of expression data, but the results are often hard to interpret, we developed unsupervised decision trees that take Gene Ontology annotations into account during the clustering process. Although in many cases standard clustering methods with a succeeding over-representation analysis yields similar results, the unsupervised decision trees can constitute an interesting complimentary method.

Most parts of this thesis considered networks as a source of biological knowledge. As a general tool for handling biological networks and expression data, ToPNet was developed with many methods for interactive exploration and visualization of the data. ToPNet also provides the basis to this work's main contribution, which consists of a general framework to query networks with associated expression data and functional annotations, the *pathway query language* and the *pathway search* algorithm.

What was achieved through the *pathway query language*? With *pathway queries* it is possible to formulate quite complex hypotheses and assess these with different scoring methods. But in our applications, especially with the osteoarthritis data, it turned out that the results are often not as specific as we would like. For instance, we can list transcription

factors that are likely to be relevant for the observed expression data, but we cannot figure out how these transcription factors are regulated and what their regulation has to do with osteoarthritis. If our goal is to get from mere observation of changes in expression patterns to an explanation of these observations, we have just made the first step. There is still a long way to go, and that way might eventually lead to complete dynamical models of pathways and networks as they are suggested by today's systems biologists. The advantage of methods relying on less detailed models like *pathway queries* together with the suggested scoring schemes is that available knowledge can be incorporated at different levels of detail and completeness. In order to build a complete dynamical model of a network, all interactions and the corresponding parameters have to be determined experimentally, a process that is at least costly and time-consuming and in most cases simply infeasible with today's methods. With *pathway queries*, we can make use of background knowledge that is far less detailed. We can use *pathway queries* to define interesting contexts for expression data. The search algorithm and the scoring functions provide us with instances that appear significant with respect to that context. In principle, we could assess complex hypotheses, but at the moment such hypotheses are often not supported by the available network data. There is still a large gap between the detailed knowledge that can be found in the literature and the information available in biological network databases. There are at least two reasons for that gap:

1. Community efforts to build detailed network databases are just starting (e.g. Reactome[1]).

2. For many important details and subtleties of biological results there is no standard formalization available.

3. It is often difficult to estimate to what extend experimental results can be transfered to other situations.

The efforts to create comprehensive network databases could be supported by text mining or by journal policies that require a formalized deposition of main results in public databases for a publication. Unfortunately, text mining approaches so far fail to extract detailed information from the scientific literature.

For the second and third problems the question of knowledge representation has to be reconsidered. A representation is desirable that can accommodate the need for detail for an in-depth representation of biological facts as well as generalization in order to transfer specific results to other contexts. Such an approach would require careful modeling of orthology, family and other relationships that could allow to transfer properties from one object to another. When interpreting experimental data from human, it will often be desirable to include background knowledge available from experiments on model organisms like mouse or rat. But transferring knowledge from one organism, cell line, or condition to another is not only a problem of representation. Due to the complexity of many biological systems there is no general rule when such a transfer is valid. If for instance the activation

[1] `http://www.reactome.org`

of a transcription factor induces the expression of a target gene in one tissue, this might not be the case in another tissue. It is a crucial issue to generalize experimental results as far as possible, but the difficulty of this inductive step is one of the reasons why biology is such a complex science. For working with *pathway queries* we suggest to generalize and transfer facts liberally, so that as much knowledge as possible can be utilized for the generation of instances. The scoring methods and manual inspection must then determine the relevant instances.

In summary, the *pathway query language* allows to make use of background knowledge in the form of networks at different levels of detail. The more detailed the available networks are, the more specific *pathway queries* can be developed and analyzed. *Pathway queries* are no substitute for detailed dynamical models, but they represent an efficient way of querying available knowledge and putting it into context with experimental data.

7.2 Future challenges

Advances in biology and especially in bioinformatics are often driven by the technology of biological measurements. Novel experimental methods often allow addressing new biological questions and produce new types of data. Therefore, novel algorithmic techniques will often be required as well. Thus, if we would like to know what awaits bioinformatics in the coming years, we have to take a look at developments in biotechnology.

But not only biotechnology but also new understandings in biology can lead to new challenges. An example for new insights in biology that raised new problems is the understanding of a gene. Not long ago it was believed that in most cases one gene would correspond to one protein. While alternative splicing is still not well understood today, it is generally believed that it does play a fundamental role in the generation of the great functional diversity in the human proteome.

The analysis of biological data like sequences or mRNA expression in order to answer specific research questions is at the core of bioinformatics. But besides this targeted data analysis, the integration of data may allow to address new questions, which we tried to demonstrate in this thesis. This kind of data integration is another field where many challenges lie ahead. Especially for industrial applications of bioinformatics methods it will be crucial to provide data from many sources together with algorithms working on these data.

In the following, challenges arising from these different aspects will be discussed.

7.2.1 New generation of microarrays

Due to improvements in lithography technology, the main provider of high density oligonucleotide microarrays, Affymetrix, can print more and more probes on a single array. The next generation of Affymetrix microarrays will include the All-Exon arrays and Tiling arrays. The All-Exon array type has recently become available and the Tiling arrays will be available soon. Both types have several million probes on a single microarray. The

All-Exon array has a probe set (usually consisting of four probes) for every exon supported by an Ensembl or RefSeq transcript sequence and, in addition, many putative exons derived by gene prediction software and other methods. This technology makes it possible to discover differential splicing events, i.e. genes that are differently spliced in two experimental conditions. Splicing is believed to have great influence on the functional diversity of proteins, and differential splicing therefore represents a regulatory mechanism that for the first time can be investigated on a large scale. The final goal of an analysis with exon arrays is to estimate the expression level not only for genes, but for all possible transcripts.

Tiling arrays represent the complete genome sequence except repetitive elements at a certain resolution, and thus allow for the identification of previously unknown transcribed elements. Possible applications of tiling arrays are discussed in Bertone et al. (2005).

7.2.2 Proteomics and metabolomics

Mass spectrometry (MS) is currently the key technology for proteomics and metabolomics. Advances in related technologies drive the development in proteomics and metabolomics. Glinski and Weckwerth (2005) demonstrate this connection for studies in the area of plant physiology. They also describe what can be achieved with state-of-the-art MS technology. Fiehn (2002) provides an overview on metabolomic analysis methods, their differences and terminology. In addition, some approaches for mining metabolite data and modeling the metabolic behavior of an organism are described.

Today, models of the metabolism of many organisms are available at a quite detailed level. The stoichiometry of most relevant reactions as well as the participating enzymes are known. One important challenge is the integration of transcriptomic data with metabolomic data. Such integrated models will foster the understanding of the interplay between metabolism and regulation of gene expression.

7.2.3 MicroRNA and epigenetics

Regulation through microRNAs and DNA methylation constitute regulatory mechanisms that receive increasing attention in biological research. While expression regulation through protein-DNA interaction by transcription factors is in principle quite well understood, the importance of those two mechanisms has only recently become obvious. Large-scale measurements for microRNAs will soon become available as they can be performed with microarrays, for instance with the tiling arrays described above. So far, no technology for large-scale measurements of DNA methylation states has been developed.

These two new regulatory mechanisms add a new dimension to the process of gene and protein regulation. Since microRNA targets might be easy to identify by sequence methods and their concentration can be measured with microarrays, it can be hoped that their understanding will develop quickly and their influence can soon be accounted for in regulatory models.

It is also possible to induce gene silencing with artificially introduced RNAs, which is then called RNA interference (RNAi). Using so-called cell microarrays, genome-wide loss-

of-function studies have become feasible with RNAi (Wheeler et al., 2005). The interfering RNAs are spotted on a microarray and cell cultures are grown. The cultures can then be analyzed for different phenotypes such as cell morphology, cell viability, or the expression of a reporter gene.

7.2.4 Data integration

As mentioned above, an important challenge for future research is the improvement of data integration. The current situation is characterized by data of different quality and level of detail. These differences have to be accommodated for and, in addition, data from different experimental approaches and different species should be integrated into algorithmic approaches that aim at improving the understanding of certain biological processes or diseases.

Integration of data from different species is of particular importance. On the sequence level, comparisons between different species has become a standard procedure for many research questions as sequence conservation can often provide valuable information about the functional significance of certain features. An example for that approach was presented in 6.2.2, where we used sequence conservation to find functional transcription factor binding sites. An interesting question for future research is how far this comparative approach can be extended to other types of data such as networks or even expression data.

7.3 Final remarks

Most parts of this thesis were concerned with the integration of different data types, especially gene expression data and biological networks. As larger and better curated databases of networks will certainly evolve, it is important to develop algorithms that can work with these new data. At the moment, two kinds of methods are prevalent: The first one comprises simple querying mechanisms, usually based on keyword matching, as they can be found in almost any web interface for a biological database. The second one covers simulation approaches, where very detailed models are needed, which are available only for very few biological systems. *Pathway queries* were developed as an attempt to provide a method that lies between these two approaches. They provide capabilities for complex queries, taking network structure into account, as well as for basic reasoning using statistical scoring methods. As the name suggests, *pathway queries* are still mainly a querying mechanism. Further developments should improve the reasoning capabilities, possibly taking the dynamics of biological systems into account. Ideas for such developments could be borrowed from simulation methods as well as classical reasoning systems. With such an improved querying and reasoning system, it will be possible to integrate existing and emerging biological databases much more efficiently into the task of interpreting newly generated data.

Appendix A

XML Schema and stylesheet of the Pathway Query Language

A.1 Schema definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
        xmlns:pw="http://bio.informatik.uni-muenchen.de/Pathways"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns="http://www.w3.org/1999/xhtml"
        elementFormDefault="qualified"
        targetNamespace="http://bio.informatik.uni-muenchen.de/Pathways"
        version="1.0"
        xml:lang="en">

  <xs:annotation>
    <xs:documentation xml:lang="en">Schema for pathway queries.
    A pathway query describes a template for biological networks with annotations and
    experimental data.
    <p>Florian Sohler, Ralf Zimmer: <cite>Identifying active transcription factors and
        kinases from expression data using Pathway Queries.</cite>
        Bioinformatics. 2005; 21(Suppl. 2):ii115-ii122.
        </p><p>
    Florian Sohler, Daniel Hanisch, Ralf Zimmer: <cite>
          New methods for joint analysis of biological networks and expression data.</cite>
          Bioinformatics. 2004 Jul 1;20(10):1517-21.
        </p>
    </xs:documentation>
  </xs:annotation>

  <xs:element name="TheNet" type="pw:Net">
    <xs:annotation>
```

```
    <xs:documentation xml:lang="en">
    The root element for a pathway query is TheNet.
    </xs:documentation>
  </xs:annotation>
  <xs:key name="SubnetKey">
  <xs:annotation>
    <xs:documentation xml:lang="en">Names of subnet elements must be unique.
    They are referred to in connection elements.</xs:documentation>
  </xs:annotation>
    <xs:selector xpath=".//pw:Subnet"/>
    <xs:field xpath="@name"/>
  </xs:key>
  <xs:keyref name="FromRef" refer="pw:SubnetKey">
    <xs:selector xpath=".//pw:Connection"/>
    <xs:field xpath="pw:ConnectFrom"/>
  </xs:keyref>
  <xs:keyref name="ToRef" refer="pw:SubnetKey">
    <xs:selector xpath=".//pw:Connection"/>
    <xs:field xpath="pw:ConnectTo"/>
  </xs:keyref>
  <!--unique name="ConnectionKey">
    <selector xpath=".//pw:Connection"/>
       <field xpath="@name"/>
  </unique-->
</xs:element>

<xs:complexType name="Net">
  <xs:annotation>
    <xs:documentation xml:lang="en">The type Net defines a network template.
    It specifies subnetworks or single nodes in the subnet element and connections
    (corresponding to paths in instances of the template) in the connection elements.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="Subnet" type="pw:Net"/>
      <xs:choice maxOccurs="unbounded" minOccurs="0">
        <xs:element name="Connection" type="pw:ConnectionType"/>
        <xs:element name="VirtualConnection" type="pw:VirtualConnectionType"/>
      </xs:choice>
    </xs:sequence>
    <xs:element name="PathwayNode" type="pw:PathwayNodeType"/>
  </xs:choice>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute default="false" name="multiple" type="xs:boolean"/>
```

```
  <xs:attribute name="multipleMin" type="xs:nonNegativeInteger"/>
  <xs:attribute name="layer" type="xs:positiveInteger" use="optional"/>
</xs:complexType>

<xs:complexType name="ConnectionType">
  <xs:sequence>
    <xs:element maxOccurs="1" minOccurs="1" name="ConnectFrom" type="xs:string"/>
    <xs:element maxOccurs="1" minOccurs="1" name="ConnectTo" type="xs:string"/>
    <xs:element maxOccurs="1" minOccurs="0" name="PlaceQuery" type="pw:QueryType"/>
    <xs:element maxOccurs="1" minOccurs="0" name="TransitionQuery" type="pw:QueryType"/>

    <xs:element maxOccurs="1" minOccurs="0" name="Scoring" type="pw:ScoringType"/>
  </xs:sequence>
  <xs:attribute default="0" name="minEdges" type="xs:nonNegativeInteger"/>
  <xs:attribute default="1" name="maxEdges" type="xs:positiveInteger"/>
  <xs:attribute default="false" name="intersectionAllowed" type="xs:boolean"/>
  <xs:attribute default="false" name="undirected" type="xs:boolean"/>
  <xs:attribute name="layer" type="xs:nonNegativeInteger" use="optional"/>
</xs:complexType>

<xs:complexType name="VirtualConnectionType">
  <xs:sequence>
    <xs:element maxOccurs="1" minOccurs="1" name="ConnectFrom" type="xs:string"/>
    <xs:element maxOccurs="1" minOccurs="1" name="ConnectTo" type="xs:string"/>
    <xs:choice>
      <xs:element name="Comparison" type="xs:string"/>
      <xs:sequence>
        <xs:element minOccurs="0" name="FromDynamicQuery" type="pw:DynamicQueryType"/>
        <xs:element minOccurs="0" name="ToDynamicQuery" type="pw:DynamicQueryType"/>
      </xs:sequence>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DynamicQueryType">
  <xs:choice>
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="1" name="BasicQuery" type="pw:BasicQueryType"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="DynamicParameter">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="MapName" type="xs:string"/>
            <xs:element name="Node">
              <xs:simpleType>
                < xs:restriction  base="xs:string">
```

```
                    <xs:enumeration value="from"/>
                    <xs:enumeration value="to"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:sequence>
        <xs:element maxOccurs="1" minOccurs="1" name="DynamicQuery1"
          type="pw:DynamicQueryType"/>
        <xs:element maxOccurs="1" minOccurs="1" name="Operator"
          type="pw:OperatorType"/>
        <xs:element maxOccurs="1" minOccurs="1" name="DynamicQuery2"
          type="pw:DynamicQueryType"/>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="PathwayNodeType">
      <xs:annotation>
      <xs:documentation xml:lang="en">Defines a node of the network template by specifying
       restrictions  on node annotations.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="1" name="Query" type="pw:QueryType"/>
      <xs:element maxOccurs="1" minOccurs="0" name="Scoring" type="pw:ScoringType"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"/>
    <xs:attribute default="1000000000" name="maxVertices" type="xs:positiveInteger"/>
  </xs:complexType>

  <xs:complexType name="QueryType">
    <xs:annotation>
      <xs:documentation xml:lang="en">
      Combines BasicQuery elements using boolean operators.
      </xs:documentation>
    </xs:annotation>

    <xs:choice>
      <xs:element maxOccurs="1" minOccurs="1" name="BasicQuery" type="pw:BasicQueryType"/>
      <xs:sequence>
        <xs:element maxOccurs="1" minOccurs="1" name="Query1" type="pw:QueryType"/>
            <xs:element maxOccurs="1" minOccurs="1" name="Operator" type="pw:OperatorType"/>
```

```
      <xs:element maxOccurs="1" minOccurs="1" name="Query2" type="pw:QueryType"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>

<xs:simpleType name="OperatorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="and"/>
    <xs:enumeration value="or"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="BasicQueryType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Defines a query on one annotation type, e.g.
    GO molecular function = transcription factor activity .</xs:documentation>
  </xs:annotation>

  <xs:sequence>
    <xs:element maxOccurs="1" minOccurs="1" name="MapName" type="xs:string"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="Parameter" type="xs:string"/>
    <xs:element maxOccurs="1" minOccurs="1" name="Operator" type="pw:MapOperatorType"/>
    <xs:element maxOccurs="1" minOccurs="1" name="Value" type="xs:string"/>
  </xs:sequence>
  <xs:attribute default="false" name="Negated" type="xs:boolean"/>
</xs:complexType>

<xs:simpleType name="MapOperatorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="lt"/>
    <xs:enumeration value="gt"/>
    <xs:enumeration value="eq"/>
    <xs:enumeration value="like"/>
    <xs:enumeration value="gteq"/>
    <xs:enumeration value="lteq"/>
    <xs:enumeration value="isnull"/>
  </xs:restriction>
</xs:simpleType>


<xs:complexType name="ScoringType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Defines different scoring methods that are used to score
    instances of the network template.</xs:documentation>
  </xs:annotation>
```

```
    <xs:choice>
      <xs:element name="P−Value" type="pw:PValType"/>
      <xs:element name="AdditiveScore" type="pw:AdditiveScoreType"/>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="PValType">
    <xs:annotation>
      <xs:documentation xml:lang="en">Defines scoring types that result in a p−value.
      The MapName scoring type simply refers to a ToPNet data map and gets its
      result by applying the data map to the nodes of the instance.
      The RankScore should only be applied to PathwayNodes with
      multiplicity one. It specifies a data map to rank all nodes of the search network.
      From these ranks significance values will be computed.
      The FETScore works similar to the RankScore, but uses Fisher exact test to
      compute p−values. A Query is performed on
      all nodes in the search graph. The significance of the overlap with the instance nodes
      for the given PathwayNode is computed.</xs:documentation>
    </xs:annotation>

    <xs:choice>
      <xs:element name="MapName" type="xs:string"/>
      <xs:element name="MultiNode">
                <xs:complexType/>
      </xs:element>
      <xs:element name="RankScore">
        <xs:complexType>
          <xs:attribute name="MapName" type="xs:string"/>
          <xs:attribute name="tail" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="high"/>
                <xs:enumeration value="low"/>
                <xs:enumeration value="both"/>
              </xs:restriction>
                    </xs:simpleType>
          </xs:attribute>
        </xs:complexType>
      </xs:element>
      <xs:element name="FETScore">
        <xs:complexType>
          <xs:sequence>
                <xs:element name="Query" type="pw:QueryType"/>
          </xs:sequence>
```

```
          <xs:attribute name="relativeToResult" type="xs:boolean" default="false"/>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="AdditiveScoreType">
    <xs:choice>
      <xs:element name="MapName" type="xs:string"/>
    </xs:choice>
  </xs:complexType>
</xs:schema>
```

A.2 Stylesheet definition

```
TransitionQuery    {
        display: block;
}

Query    {
        display: block;
}

Query2:after     {
        content: " )";
}

Query1:before    {
        content: "( ";
}

Scoring     {
        display: block;
        margin-top: 10pt;
        margin-left: 15pt;
}

P-Value > MultiNode:before {
        content: "MultiNode";
}

P-Value > FETScore:before {
        content: "Fisher exact test";
}
```

```
P−Value > MapName:before {
        content: "Map ";
}

Scoring:before    {
        content: "Scoring Method: ";
}

ConnectTo   {
        color: green;
}

TheNet:before    {
        content: attr(name);
        font−size−adjust: 0;
        font−size: 20pt;
}

Subnet:before    {
        content: attr(name) ": ";
        color: green;
}

ConnectFrom   {
        color: green;
}

ConnectFrom:after    {
        content: " −> ";
}

Connection:before    {
        color: rgb(153, 0, 0);
        content: "Connection (" attr(maxEdges) " steps): ";
        margin−top: 0pt;
}

BasicQuery > Operator   {
        font−style:  italic ;
        margin−left: 5pt;
        margin−right: 5pt;
}

Query > Operator, Query1 > Operator, Query2 > Operator {
        font−style: normal;
```

```
        font-weight: bold;
        text-decoration: underline;
        margin-left: 5pt;
        margin-right: 5pt;
}

Value   {
}

Subnet   {
        display: list-item;
         list-style-position: outside;
        text-indent: 0pt;
        margin-top: 10pt;
        margin-bottom: 10pt;
        margin-left: 0.9893993cm;
}

MapName {
}

Connection   {
        display: list-item;
         list-style-position: outside;
        text-indent: 0pt;
        margin-bottom: 10pt;
        margin-left: 0.9893993cm;
        margin-top: 0cm;
}

TheNet   {
}

PathwayNode  {
}
```

Bibliography

Adams, M., Kelley, J., Gocayne, J., Dubnick, M., Polymeropoulos, M., Xiao, H., Merril, C., Wu, A., Olde, B., Moreno, R., et al. (1991). Complementary dna sequencing: expressed sequence tags and human genome project. *Science*, 252:1651–1656.

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 207–216, Washington D.C.

Al-Shahrour, F., Diaz-Uriarte, R., and Dopazo, J. (2004). Fatigo: a web tool for finding significant associations of gene ontology terms with groups of genes. 20(4):578–580.

Apanovitch, D. M., Slep, K. C., Sigler, P. B., and Dohlman, H. G. (1998). Sst2 is a gtpase-activating protein for gpa1: Purification and characterization of a cognate rgs-g protein pair in yeast. *Biochemistry*, 37(14):4815–4822.

Arbeitman, M. N., Furlong, E. E. M., Imam, F., Johnson, E., Null, B. H., Baker, B. S., Krasnow, M. A., Scott, M. P., Davis, R. W., and White, K. P. (2002). Gene expression during the life cycle of drosophila melanogaster. *Science*, 297(5590):2270–2275.

Ayala, G., Dai, H., Ittmann, M., Li, R., Powell, M., Frolov, A., Wheeler, T., Thompson, T., and Rowley, D. (2004). Growth and survival mechanisms associated with perineural invasion in prostate cancer. *Cancer Research*, 64(17):6082–6090.

Baetz, K., Moffat, J., Haynes, J., Chang, M., and Andrews, B. (2001). Transcriptional coregulation by the cell integrity mitogen-activated protein kinase slt2 and the cell cycle regulator swi4. *Mol. Cell. Biol.*, 21(19):6515–6528.

Baral, C., Chancellor, K., Tran, N., Tran, N., Joy, A., and Berens, M. (2004). A knowledge based approach for representing and reasoning about signaling networks. *Bioinformatics*, 20(Suppl. 1):i15–22.

Bardwell, L. (2004). A walk-through of the yeast mating pheromone response pathway. *Peptides*, 25(9):1465–1476.

Bardwell, L., Cook, J. G., Zhu-Shimoni, J. X., Voora, D., and Thorner, J. (1998). Differential regulation of transcription: repression by unactivated mitogen-activated protein

kinase kss1 requires the dig1 and dig2 proteins. *Proc Natl Acad Sci U S A*, 95(26):15400–5.

Bartel, D. P. (2004). Microrнas: Genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–297.

Basak, J. and Krishnapuram, R. (2005). Interpretable hierarchical clustering by constructing and unsupervised decision tree. 17(1):121–132.

Bautz, E. and Reilly, E. (1966). Gene-specific messenger rna: isolation by the deletion method. *Science*, 151(708):328–330.

Bell, A. C., West, A. G., and Felsenfeld, G. (2001). Insulators and boundaries: Versatile regulatory elements in the eukaryotic genome. *Science*, 291(5503):447–450.

Bellot, P. and El-Bèze, M. (2000). Clustering by means of unsupervised decision trees or hierarchical and k-means-like algortithm. In *RIAO'2000 Conference Proceedings*, volume 1, pages 344–363.

Berman, B. P., Nibu, Y., Pfeiffer, B. D., Tomancak, P., Celniker, S. E., Levine, M., Rubin, G. M., and Eisen, M. B. (2002). Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the drosophila genome. *PNAS*, 99(2):757–762.

Bertone, P., Gerstein, M., and Snyder, M. (2005). Applications of dna tiling arrays to experimental genome annotation and regulatory pathway discovery. *Chromosome Research*, 13(3):259–274.

Blackwood, E. M. and Kadonaga, J. T. (1998). Going the distance: A current view of enhancer action. *Science*, 281(5373):60–63.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., and Schneider, M. (2003). The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucl. Acids Res.*, 31(1):365–370.

Bolstad, B., Irizarry, R., Astrand, M., and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193.

Borgelt, C. and Kruse, R. (2002). Induction of association rules: Apriori implementation. In *Proceedings of the 14th Conference on Computational Statistics*, Berlin, Germany.

Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh. ACM.

Breitling, R., Amtmann, A., and Herzyk, P. (2004). Iterative group analysis (iGA): a simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, 5:5–34.

Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, Manuel, J., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS*, 97(1):262–267.

Buchwalter, G., Gross, C., and Wasylyk, B. (2004). Ets ternary complex transcription factors. *Gene*, 324:1–14.

Bustin, S. (2000). Absolute quantification of mrna using real-time reverse transcription polymerase chain reaction assays. *J Mol Endocrinol*, 25(2):169–193.

Casella, G. and Berger, R. L. (2002). *Statistical inference*. Duxbury, Pacific Grove, 2. edition.

Catron, K. M., Brickwood, J. R., Shang, C., Li, Y., Shannon, M. F., and Parks, T. P. (1998). Cooperative binding and synergistic activation by rela and c/ebpbeta on the intercellular adhesion molecule-1 promoter. *Cell Growth Differ*, 9(11):949–59. 1044-9523 Journal Article.

Chen, L.-F. and Greene, W. C. (2004). Shaping the nuclear action of nf-κb. *Nature Reviews Molecular Cell Biology*, 5:392–401.

Cherry, J., Adler, C., Ball, C., Chervitz, S., Dwight, S., Hester, E., Jia, Y., Juvik, G., Roe, T., Schroeder, M., Weng, S., and Botstein, D. (1998). Sgd: Saccharomyces genome database. *Nucl. Acids Res.*, 26(1):73–79.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., and Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science*, 282(5389):699–705.

Civil, A., Rensink, I., Aarden, L. A., and Verweij, C. L. (1999). Functional disparity of distinct cd28 response elements toward mitogenic responses. *J Biol Chem*, 274(48):34369–74. 0021-9258 Journal Article.

Cleveland, W. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Amer. Stat. Assoc.*, 74:829–836.

Coe, B. (2003). You want ketchup with your dna chips? an overview of expression microarrays. *BioTeach Online Journal*, 1:89–94.

Creighton, C. and Hanash, S. (2003). Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86.

Daignan-Fornier, B. and Fink, G. (1992). Coregulation of purine and histidine biosynthesis by the transcriptional activators bas1 and bas2. *Proceedings of the National Academy of Sciences*, 89:6746–50.

Daraselia, N., Yuryev, A., Egorov, S., Novichkova, S., Nikitin, A., and Mazo, I. (2004). Extracting human protein interactions from medline using a full-sentence parser. *Bioinformatics*, 20(5):604–611.

de Nobel, H., van Den Ende, H., and Klis, F. M. (2000). Cell wall maintenance in fungi. *Trends Microbiol*, 8(8):344–5. 0966-842x Journal Article Review Review, Tutorial.

DeRisi, J. L., Iyer, V. R., and Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–685.

Dolan, J. W., Kirkman, C., and Fields, S. (1989). The yeast ste12 protein binds to the dna sequence mediating pheromone induction. *Proc Natl Acad Sci U S A*, 86(15):5703–7.

Draghici, S., Khatri, P., Martins, R. P., Ostermeier, G., and Krawetz, S. A. (2003). Global functional profiling of gene expression. *Genomics*, 81(2):98–104.

Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P., and Trent, J. M. (1999). Expression profiling using cdna microarrays. *Nature Genetics*, 21 Suppl 1:10–14.

Dulos, J., van der Vleuten, M. A. J., Kavelaars, A., Heijnen, C. J., and Boots, A. M. (2005). Cyp7b expression and activity in fibroblast-like synoviocytes from patients with rheumatoid arthritis: Regulation by proinflammatory cytokines. *Arthritis & Rheumatism*, 52(3):770–778.

Edwards, D. (2003). Non-linear normalization and background correction in one-channel cdna microarray studies. *Bioinformatics*, 19(7):825–833.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868.

Fiehn, O. (2002). Metabolomics – the link between genotypes and phenotypes. *Plant Molecular Biology*, 48(1-2):155–171.

Fields, S. and Song, O.-K. (1989). A novel genetic system to detect protein?protein interactions. *Nature*, 340:245–246.

Fields, S. and Sternglanz, R. (1994). The two-hybrid system: an assay for protein-protein interactions. *Trends in Genetics*, 10(8):286–292.

Fisher, R. (1932). *Statistical methods for research workers*. Oliver and Boyd, London, 4th edition.

Fraley, C. and Raftery, A. (2002). Mclust:software for model-based clustering, density estimation and discriminat analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631.

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. *Journal of Computational Biology*, 7(3-4):601–620.

Friese, M. A., Jones, E. Y., and Fugger, L. (2005). Mhc ii molecules in inflammatory diseases: interplay of qualities and quantities. *Trends in Immunology*, 26(11):559–561.

Fruchtermann, T. M. J. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software – Practice and Experience*, 21:1129–1164.

Fundel, K., Kffner, R., Aigner, T., and Zimmer, R. (2005). Data processing effects on the interpretation of microarray gene expression experiments. In Torda, A., Kurtz, S., and Rarey, M., editors, *German Conference on Bioinformatics (GCB) 2005*, volume P-71 of *GI Lecture Notes in Informatics*, pages 77–91.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Gansner, E., Koutsofios, E., and North, S. (2002). *Drawing graphs with dot*. AT&T.

Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability : a guide to the theory of NP-completeness*. W. H. Freeman, San Francisco. Michael R. Garey, David S. Johnson. ill. ; 24 cm. A Series of books in the mathematical sciences Includes indexes.

Gat-Viks, I., Tanay, A., Raijman, D., and Shamir, R. (2005). The factor graph network model for biological systems. In *Research in Computational Molecular Biology : 9th Annual International Conference, RECOMB 2005, Cambridge, MA, USA, MAY 14-18, 2005, Proceedings*, volume 3500 of *Lecture Notes in Bioinformatics*. Springer.

Gebauer, M., Saas, J., Sohler, F., Haag, J., Sder, S., Pieper, M., Bartnik, E., Beninga, J., Zimmer, R., and Aigner, T. (2005). Comparison of the chondrosarcoma cell line sw1353 with primary human adult articular chondrocytes regarding their gene expression profile and their reactivity to il-1β. *Osteoarthritis and Cartilage*, 13(8):697–708.

Gentle, J. E., Hrdle, W., and Mori, Y., editors (2004). *Handbook of computational statistics*, chapter III.5. Springer-Verlag, Heidelberg.

Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Li, F. L. C., Maechler, M., Rossini, A., Sawitzki, G., Smith, C., Smyth, G., Tierney,

L., Yang, J., and Zhang, J. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80.

Gentleman, R., Carey, V., Dudoit, S., Irizarry, R., Huber, W., and Smyth, G., editors (2005). *LIMMA: linear models for microarray data. In: Bioinformatics and computational biology solutions using R and Bioconductor*, chapter 23. Springer, New York.

Georgii, E., Richter, L., Ruckert, U., and Kramer, S. (2005). Analyzing microarray data using quantitative association rules. *Bioinformatics*, 21(Suppl. 2):ii123–129.

Glinski, M. and Weckwerth, W. (2005). The role of mass spectrometry in plant systems biology. *Mass Spectrometry Reviews*. Article online in advance of print.

Gómez-Camarillo, M. A. and Kouri, J. B. (2005). Ontogeny of rat chondrocyte proliferation: studies in embryo, adult and osteoarthritic (oa) cartilage. *Cell Research*, 15(2):99–104.

Grosu, P., Townsend, J. P., Hartl, D. L., and Cavalieri, D. (2002). Pathway processor: A tool for integrating whole-genome expression results into metabolic networks. *Genome Res.*, 12(7):1121–1126.

Guetsova, M. L., Lecoq, K., and Daignan-Fornier, B. (1997). The isolation and characterization of saccharomyces cerevisiae mutants that constitutively express purine biosynthetic genes. *Genetics*, 147(2):383–97.

Hanisch, D. (2004). *New analysis methods for gene expression data via construction and incorporation of biological networks*. PhD thesis, Ludwig-Maximilians-Universität München.

Hanisch, D., Fluck, J., Mevissen, H. T., and Zimmer, R. (2003). Playing biology's name game: identifying protein names in scientific text. *Pac Symp Biocomput*, pages 403–14.

Hanisch, D., Sohler, F., and Zimmer, R. (2004). Topnet–an application for interactive analysis of expression data and biological networks. *Bioinformatics*, 20(9):1470–1471.

Hanisch, D., Zien, A., Zimmer, R., and Lengauer, T. (2002). Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18 Suppl 1:S145–S154.

Hardiman, G. (2002). Microarray technologies – an overview. *Pharmacogenomics*, 3(3):293–297.

Hochberg, Y. (1988). A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75:800–802.

Hollenhorst, P. C., Pietz, G., and Fox, C. A. (2001). Mechanisms controlling differential promoter-occupancy by the yeast forkhead proteins fkh1p and fkh2p: implications for regulating the cell cycle and differentiation. *Genes Dev.*, 15(18):2445–2456.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70.

Holton, W. T. (2003). Evaluation of mapping consistency between public gene and protein databases.

Hommel, G. (1988). A stagewise rejective multiple test procedure based on a modified bonferroni test. *Biometrika*, 75:383–386.

Honda, K., Yamada, T., Hayashida, Y., Idogawa, M., Sato, S., Hasegawa, F., Ino, Y., Ono, M., and Hirohashi, S. (2005). Actinin-4 increases cell motility and promotes lymph node metastasis of colorectal cancer. *Gastroenterology*, 128(1):51–62.

Horn, P. J. and Peterson, C. L. (2002). Molecular biology: Chromatin higher order folding–wrapping up transcription. *Science*, 297(5588):1824–1827.

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.

Hwang-Shum, J. J., Hagen, D. C., Jarvis, E. E., Westby, C. A., and Sprague, G. F., J. (1991). Relative contributions of mcm1 and ste12 to transcriptional activation of a- and alpha-specific genes from saccharomyces cerevisiae. *Mol Gen Genet*, 227(2):197–204.

Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18 Suppl 1:S233–40.

Iragne, F., Barre, A., Goffard, N., and de Daruvar, A. (2004). Aliasserver: a web server to handle multiple aliases used to refer to proteins. *Bioinformatics*, 20(14):2331–2332.

Kanehisa, M. (1996). Toward pathway engineering: a new database of genetic and molecular pathways. *Science & Technology Japan*, 59:34–38.

Karakos, D., Khudanpur, S., Eisner, J., and Priebe, C. E. (2005). Unsupervised classification via decision trees: an information-theoretic perspective. In *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE.

Kaufman, L. and Rouosseeuw, P. (1990). *Finding groups in data: an introduction to cluster analysis*, chapter 5. Wiley, New York.

Kawashima, T., Murata, K., Akira, S., Tonozuka, Y., Minoshima, Y., Feng, S., Kumagai, H., Tsuruga, H., Ikeda, Y., Asano, S., Nosaka, T., and Kitamura, T. (2001). Stat5 induces macrophage differentiation of m1 leukemia cells through activation of il-6 production mediated by nf-kappab p65. *J Immunol*, 167(7):3652–60. 0022-1767 Journal Article.

Khatri, P. and Draghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595.

Kim, C. and Falkow, S. (2003). Significance analysis of lexical bias in microarray data. *BMC Bioinformatics*, 4(12).

Kohonen, T. (1988). *Self-Organization and Associative Memory*. Springer-Verlag, New York.

Kooperberg, C., Fazzio, T. G., Delrow, J. J., and Tsukiyama, T. (2002). Improved background correction for spotted dna microarrays. *Journal of Computational Biology*, 9(1):55–66.

Krauthammer, M., Rzhetsky, A., Morozov, P., and Friedman, C. (2000). Using blast for identifying gene and protein names in journal articles. *Gene*, 259(1-2):245–252.

Krishnan, V. and Westhead, D. (2003). A comparative study of machine-learning methods to predict the effects of single nucleotide polymorphisms on protein function. *Bioinformatics*, 19(17):2199–2209.

Lance, E., Kimura, L., and C.N.Manibog (1993). The expression of major histocompatibility antigens on human articular chondrocytes. *Clinical Orthopedics and Related Research*, 291:266–282.

Lapointe, J., Li, C., Higgins, J., v. d. Rijn, M., Blair, E., Montgomery, K., Ferrari, M., Egevad, L., Rayford, W., Bergerheim, U., Ekman, P., DeMarzo, A., Tibshirani, R., D.Botstein, Brown, P., Brooks, J., and Pollack, J. (2004). Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *PNAS*, 101(3):811–816.

Larsson, C. (2006). Protein kinase c and the regulation of the actin cytoskeleton. *Cellular Signalling*, 18(3):276–284.

Leaman, D., Chen, P. Y., Fak, J., Yalcin, A., Pearce, M., Unnerstall, U., Marks, D. S., Sander, C., Tuschl, T., and Gaul, U. (2005). Antisense-mediated depletion reveals essential and specific functions of microrna in drosophila development. *Cell*, 121(7):1097–1108.

Lee, N. H. (2005). Genomic approaches for reconstructing gene networks. *Pharmacogenomics*, 6(3):245–258.

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

Leser, U. (2005). A query language for biological networks. *Bioinformatics*, 21(Suppl. 2):ii33–39.

Lönnstedt, I. and Speed, T. (2002). Replicated micorarray data. *Statistica Sinica*, 12:31–46.

Lottaz, C. and Spang, R. (2005). Molecular decomposition of complex clinical phenotypes using biologically structured analysis of microarray data. *Bioinformatics*, 21(9):1971–1978.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE transactions on computational biology and bioinformatics*, 1(1):24–45.

Maglott, D., Ostell, J., Pruitt, K. D., and Tatusova, T. (2005). Entrez gene: gene-centered information at ncbi. *Nucl. Acids Res.*, 33(Suppl. 1):D54–58.

Malemud, C. J., Islam, N., and Haqqi, T. (2003). Pathophysiological mechanisms in osteoarthritis lead to novel therapeutic strategies. *Cells Tissues Organs*, 174:34–48.

Markowetz, F., Bloch, J., and Spang, R. (2005). Non-transcriptional pathway features reconstructed from secondary effects of rna interference. *Bioinformatics*, 21(21):4026–4032.

Marshall, E. (2004). Getting the noise out of gene arrays. *Science*, 306(5696):630–631.

Martel-Pelletier, J., Welsch, D. J., and Pelletier, J.-P. (2001). Metalloproteases and inhibitors in arthritic diseases. *Best Practice and Research in Clinical Rheumatology*, 15(5):805–829.

Matsuno, H., Doi, A., Nagasaki, M., and Miyano, S. (2000). Hybrid petri net representation of gene regulatory network. *Proc. Pacific Symposium on Biocomputing*, 5:341–352.

Michielis, S., Koscielny, S., and Hill, C. (2005). Prediction of cancer outcome with microarrays: a multiple random validation study. *The Lancet*, 365:488–492.

Millward-Sadler, S. J. and Salter, D. M. (2004). Integrin-dependent signal cascades in chondrocyte mechanotransduction. *Annals of Biomedical Engineering*, 32(3):435–446.

Muerhoff, A. S., Leary, T. P., Desai, S. M., and Mushahwar, I. K. (1997). Amplification and subtraction methods and their application to the discovery of novel human viruses. *Journal of Medical Virology*, 53(1):96–103.

Nagasaki, M., Doi, A., Matsuno, H., and Miyano, S. (2004). A versatile petri net based architecture for modeling and simulation of complex biological processes. *Genome Informatics*, 15(1):180–197.

Natarajan, K., Meyer, M., Jackson, B., Slade, D., Roberts, C., Hinnebusch, A., and Marton, M. (2001). Transcriptional profiling shows that gcn4 is a master regulator of gene expression during amino acid starvation in yeast. *Molecular and Cellular Biology*, 21(13):4347–4368.

Orlando, V. (2000). Mapping chromosomal proteins in vivo by formaldehyde-crosslinked-chromatin immunoprecipitation. *Trends in Biochemical Sciences*, 25(3):99–104.

Palenchar, P., Kouranov, A., Lejay, L., and Coruzzi, G. (2004). Genome-wide patterns of carbon and nitrogen regulation of gene expression validate the combined carbon and nitrogen (CN)-signaling hypothesis in plants. *Genome Biology*, 5(11).

Parker, R. and Barnes, N. (1999). mrna: detection by in situ and northern hybridization. *Methods in Molecular Biology*, 106:247–283.

Pavlidis, P. (2005). Erminej – gene ontology analysis for microarray data, v2.0.4. http://microarray.genomecenter.columbia.edu/ermineJ.

Pavlidis, P., Lewis, D., and Noble, W. (2002). Exploring gene expression data with class scores. *Pac. Symp. Biocomput.*, pages 474–485.

Peters, J. H., Loredo, G. A., and Benton, H. P. (2002). Is osteoarthritis a 'fibronectin-integrin imbalance disorder'? *Osteoarthritis and Cartilage*, 10(11):831–835.

Petricoin III, E., Hackett, J., Lesko, L., Puri, R., Gutman, S., Chumakov, K., Woodcock, J., Feigal, Jr., D., Zoon, K., and Sistare, F. (2002). Medical applications of microarray technologies: a regulatory science perspective. *Nature Genetics*, 32 Suppl 2:474–479.

Pfander, D., Krtje, D., Weseloh, G., and Swoboda, B. (2001). Zellproliferation im humanen arthrotischen gelenkknorpel. *Zeitschrift für Orthopädie und ihre Grenzgebiete*, 139:375–381.

Poole, C. (1997). Articular cartilage chondrons: form, function and failure. *Journal of Anatomy*, 191:1–13.

Quackenbush, J. (2002). Microarray data normalization and transformation. *Nature Genetics*, 32 Suppl 2:496–501. There's more in that NG supplement.

Quinlan, J. (1993). *C4.5: programs for machine learning*, chapter 2. Morgan Kaufmann.

Quintavalla, J., Kumar, C., Daouti, S., Slosberg, E., and Uziel-Fusi, S. (2005). Chondrocyte cluster formation in agarose cultures as a functional assay to identify genes expressed in osteoarthritis. *Journal of Cellular Physiology*, 204(2):560–566.

R Development Core Team (2004). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rajagopalan, D. (2003). A comparison of statistical methods for analysis of high density oligonucleotide array data. *Bioinformatics*, 19(12):1469–1476.

Ren, B., Robert, F., Wyrick, J. J., Aparicio, O., Jennings, E. G., Simon, I., Zeitlinger, J., Schreiber, J., Hannett, N., Kanin, E., Volkert, T. L., Wilson, C. J., Bell, S. P., and Young, R. A. (2000). Genome-Wide Location and Function of DNA Binding Proteins. *Science*, 290(5500):2306–2309.

Reveille, J. (2005). Genetic studies in the rheumatic diseases: present status and implications for the future. *Journal of Rheumatology*, 32(Supplement 72):10–13.

Rigaut, G., Shevchenko, A., Rutz, B., Wilm, M., Mann, M., and S'eraphin, B. (1999). A generic protein purification method for protein complex characterization and proteome exploration. *Nature Biotechnology*, 17:1030–1032.

Riyazi, N., Spee, J., Huizinga, T. W. J., Schreuder, G. M. T., de Vries, R. R. P., Dekker, F. W., and Kloppenburg, M. (2003). HLA class II is associated with distal interphalangeal osteoarthritis. *Ann Rheum Dis*, 62(3):227–230.

Rohatgi, V. K. and Saleh, A. K. M. E. (2001). *An introduction to probability and statistics*. Wiley series in probability and statistics. Texts and references section. Wiley, New York, 2nd edition.

Rossi, S., Graner, E., Febbo, P., Weinstein, L., Bhattacharya, N., Onody, T., Bubley, G., Balk, S., and Loda, M. (2003). Fatty acid synthase expression defines distinct molecular signatures in prostate cancer. *Molecular Cancer Research*, 1(10):707–715.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and applied mathematics*, 20:53–65.

Sandell, L. J. and Aigner, T. (2001). Articular cartilage and changes in arthritis: Cell biology of osteoarthritis. *Arthritis Research and Therapy*, 3(2):107–113.

Scarano, M. I., Strazzullo, M., Matarazzo, M. R., and D'Esposito, M. (2005). Dna methylation 40 years later: Its role in human health and disease. *Journal of cellular physiology*, 204:21–35.

Schacherer, F., Choi, C., Gtze, U., Krull, M., Pistor, S., and Wingender, E. (2001). The transpath signal transduction database: a knowledge base on signal transduction networks. *Bioinformatics*, 17:1.

Schena, M. (2000). *Microarray Biochip Technology*. Westborough, MA: BioTechniques Press.

Schuster, S., Dandekar, T., and Fell, D. A. (1999). Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology*, 17(2):53–60.

Schwarz, R., Musch, P., von Kamp, A., Engels, B., Schirmer, H., Schuster, S., and Dandekar, T. (2005). Yana – a software tool for analyzing flux modes, gene-expression and enzyme activities. *BMC Bioinformatics*, 6(135).

Sherwood, P. W., Tsang, S. V., and Osley, M. A. (1993). Characterization of hir1 and hir2, two genes required for regulation of histone gene transcription in saccharomyces cerevisiae. *Molecular Cell Biology*.

Slonim, D. K. (2002). From patterns to pathways: gene expression data analysis comes of age. *Nature Genetics*, 32 Suppl 2:502–508.

Smalheiser, N. R. and Swanson, D. R. (1998). Using arrowsmith: a computer-assisted approach to formulating and assessing scientific hypotheses. *Computer Methods and Programs in Biomedicine*, 57(3):149–153.

Smyth, G. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(3).

Sohler, F., Hanisch, D., and Zimmer, R. (2004). New methods for joint analysis of biological networks and expression data. *Bioinformatics*, 20(10):1517–1521.

Sohler, F. and Zimmer, R. (2005). Identifying active transcription factors and kinases from expression data using pathway queries. *Bioinformatics*, 21(Suppl. 2):ii115–122.

Southern, E. (2001). Dna microarrays. history and overview. *Methods in Molecular Biology*, 170:1–15.

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):3273–3297.

Spiecker, M., Darius, H., and Liao, J. K. (2000). A functional role of i kappa b-epsilon in endothelial cell activation. *J Immunol*, 164(6):3316–22. 0022-1767 Journal Article.

Srinivasan, P. and Libbus, B. (2004). Mining medline for implicit links between dietary substances and diseases. *Bioinformatics*, 20(Suppl. 1):i290–296.

Steffen, M., Petti, A., Aach, J., D'haeseleer, P., and Church, G. (2002). Automated modelling of signal transduction networks. *BMC Bioinformatics*, 3(1):34.

Stoll, D., Templin, M., Bachmann, J., and Joos, T. (2005). Protein microarrays: applications and future challenges. *Curr Opin Drug Discov Devel*, 8:239–252.

Storey, J. and Tibshirani, R. (2003). Statistical significance for genomewide studies. *PNAS*, 100(16):9440–9445.

Stormo, G. D. (2000). DNA binding sites: representation and discovery . *Bioinformatics*, 16(1):16–23.

Stoughton, R. B. (2005). Applications of dna microarrays in biology. *Annual Review of Biochemistry*. Advance publication.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *PNAS*, 96(6):2907–2912.

Tanay, A. and Shamir, R. (2001). Computational expansion of genetic networks. *Bioinformatics*, 17(Suppl. 1):270–278.

Tedford, K., Kim, S., Sa, D., Stevens, K., and Tyers, M. (1997). Regulation of the mating pheromone and invasive growth responses in yeast by two map kinase substrates. *Current Biology*, 7:228–38.

The Gene Ontology Consortium (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29.

The Gene Ontology Consortium (2001). Creating the gene ontology resource: design and implementation. *Genome Research*, 11:1425.

Tian, L., Greenberg, S. A., Kong, S. W., Altschuler, J., Kohane, I. S., and Park, P. J. (2005). Discovering statistically significant pathways in expression profiling studies. *PNAS*, 102(38):13544–13549.

Törönen, P., Kolehmainen, M., Wong, G., and Castrn, E. (1999). Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451(2):142–146.

Tran, N., Baral, C., Nagaraj, V. J., and Joshi, L. (2005). Knowledge-based framework for hypothesis formation in biochemical networks. *Bioinformatics*, 21(Suppl. 2):ii213–219.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. (2001). Missing value estimation methods for dna microarrays. 17(6):520–525.

Turner, S., Ricci, A., Petropoulos, H., Genereaux, J., Skerjanc, I., and Brandl, C. (2002). The e2 ubiquitin conjugase rad6 is required for the argr/mcm1 repression of arg1 transcription. *Molecular and Cellular Biology*.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98(9):5116–5121.

Urnov, F. D. (2003). Chromatin remodeling as a guide to transcriptional regulatory networks in mammals. *Journal of Cellular Biochemistry*, 88(4):684–694.

Vapnik, V. N. (1998). *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York.

Velculescu, V., Zhang, L., Vogelstein, B., and Kinzler, K. (1995). Serial analysis of gene expression. *Science*, 270:484–487.

Venkatasubbarao, S. (2004). Microarrays - status and prospects. *Trends in Biotechnology*, 22(12):630–637.

Vincenti, M. P. and Brinckerhoff, C. E. (2002). Transcriptional regulation of collagenase (mmp-1, mmp-13) genes in arthritis: integration of complex signaling pathways for the recruitment of gene-specific transcription factors. *Arthritis Research*, 4(3):157–164.

Voit, E. O. and Radivoyevitch, T. (2000). Biochemical systems analysis of genome-wide expression data. *Bioinformatics*, 16(11):1023–1037.

Wagner, A. (2001). How to reconstruct a large genetic network from n gene perturbations in fewer than n2 easy steps. *Bioinformatics*, 17(12):1183–1197.

Watson, J. D. and Crick, F. H. (1953). The structure of dna. *Cold Spring Harb Symp Quant Biol*, 18:123–131.

Wheeler, D. B., Carpenter, A. E., and Sabatini, D. M. (2005). Cell microarrays and rna interference chip away at gene function. *Nature Genetics*, 37(Suppl. 1):S25–S30.

Wingender, E., Chen, X., Hehl, R., Karas, H., Liebich, I., Matys, V., Meinhardt, T., Pruss, M., Reuter, I., and Schacherer, F. (2000). Transfac: an integrated system for gene expression regulation. *Nucleic Acids Res*, 28(1):316–9.

Xenarios, I., Rice, D., Salwinski, L., Baron, M., Marcotte, E., and Eisenberg, D. (2000). Dip: The database of interacting proteins. *Nucleic Acids Research*, 28:289–91.

Xu, Z., Dziarski, R., Wang, Q., Swartz, K., Sakamoto, K. M., and Gupta, D. (2001). Bacterial peptidoglycan-induced tnf-alpha transcription is mediated through the transcription factors egr-1, elk-1, and nf-kappab. *J Immunol*, 167(12):6975–82. 0022-1767 Journal Article.

Yang, Y., Buckley, M., and Speed, T. (2001). Analysis of cdna microarray images. *Briefings in Bioinformatics*, 2(4).

Yang, Y., Dudoit, S., Luu, P., Lin, D., Peng, V., Ngai, J., and Speed, T. (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4).

Yeang, C.-H. and Jaakola, T. (2003). Physical network models and multi-source data integration. In *The Seventh Annual International Conference on Research in Computational Molecular Biology*.

Zien, A., Kuffner, R., Zimmer, R., and Lengauer, T. (2000). Analysis of gene expression data with pathway scores. *Proceedings of the International Conference of Intelligent Systems Molecular Biology*, 8:407–417.

Zupan, B., Bratko, I., Demsar, J., Juvan, P., Curk, T., Borstnik, U., Beck, J. R., Halter, J., Kuspa, A., and Shaulsky, G. (2003). Genepath: a system for inference of genetic networks and proposal of genetic experiments. *Artificial Intelligence in Medicine*, 29(1-2):107–130.

Acknowledgements

There are many people who supported me in various ways during the years I have been working on this thesis, and I would like to express my gratitude towards all of them.

First of all, I would like to thank my advisor Ralf Zimmer for giving me the opportunity to work in the field of gene expression analysis and write my thesis on this exciting topic. In many interesting discussions he contributed new ideas and helped me building a coherent body of work.

The bioinformatics group at the LMU was like a second family during long hours of work. There was always someone willing to discuss a bioinformatics problem or simply chat over a cup of coffee. In particular, I would like to thank Joannis Apostolakis. He always had an open ear for my problems and a mind full of interesting ideas.

There are three people whose contribution I would like to emphasize: Working with Jan Gewehr on the Drosophila data was very productive and we always had a good time. Henning Redestig invented with me the unsupervised decision trees, without him the corresponding chapter would not exist. Daniel Ziemek's dissertation paved the way for my own work, and we spent a lot of time coding together, making ToPNet a really useful tool.

I am grateful to Martin Vigron who reviewed the thesis, and Hans-Peter Kriegel and Volker Heun for agreeing to participate in the thesis committee.

Finally, I would like to thank Conny Donner. She gave me support and kept me motivated in times when I made little progress and finishing my thesis seemed far away.

Lebenslauf

Persönliche Daten

Name	Florian Sohler
Anschrift	Birkenwerderstraße 37 13439 Berlin
Geburtsdatum	04. April 1975
Geburtsort	Dortmund
Staatsangehörigkeit	Deutsch

Schulische und akademische Ausbildung

1981 - 1985	Grundschule: Marienschule in Paderborn
1985 - 1994	Gymnasium Theodorianum, Paderborn (Abiturnote 1,2)
1995 - 1999	Rheinische Friedrich-Wilhelms-Universität Bonn
1997	Vordiplom Informatik (Note: Sehr gut)
1998	Vordiplom Mathematik (Note: Gut)
Aug. 1999 - Aug. 2000	University of Wisconsin Madison Vollstipendium der Universität Bonn Abschluss Master of Science Titel der Masterarbeit: *Promoter prediction in E.coli: an empirical evaluation of different probabilistic methods*
Nov. 2000 - Sep. 2001	Wissenschaftlicher Mitarbeiter an der GMD in St. Augustin
Okt. 2001 - Jan. 2006	Wissenschaftlicher Mitarbeiter an der Ludwig-Maximilians-Universität München

Berufliche Tätigkeiten

Sep. 1997 - Feb. 1998	Programmiertätigkeit als studentische Hilfskraft am Lehrstuhl für Neuroinformatik der Universität Bonn
Okt. 1998 - Feb. 1999	Tutorentätigkeit begleitend zur Vorlesung Lineare Algebra I
Apr. 2003 - Nov. 2004	Externer Mitarbeiter auf Teilzeitbasis bei der Firma BioSolveIT
seit Feb. 2006	Angestellter der ComConsult GmbH, Aachen, tätig als Bioinformatiker bei der Schering AG, Berlin

Publikationen

2006 Henning Redestig, Dirk Redsilber, Florian Sohler, Joachim Selbig:
Unsupervised Decision Trees Based on Gene Ontology Terms for the Interpretation of Gene Expression Data. Submitted.

Henning Redestig[1], Florian Sohler[1], Ralf Zimmer, Joachim Selbig:
Unsupervised Decision Trees Based on Gene Ontology Terms for the Interpretation of Gene Expression Data.
[1]Authors contributed equally
Proceedings of the GfKl 2006, in press.

2005 Florian Sohler, Ralf Zimmer:
Identifying active transcription factors and kinases from expression data using Pathway Queries. Bioinformatics 21, Suppl. 2: ii115-122.

M. Szugat, D. Güttler, K. Fundel, F. Sohler, R. Zimmer:
Web Servicing the Biological Office. Bioinformatics 21, Suppl. 2: ii268-269.

Florian Sohler: Generating contexts for expression data using Pathway Queries: In Principles and Practice of Semantic Web Reasoning F. Fages, S. Soliman, (Eds.). Lecture Notes in Computer Science, Vol. 3703:160-162, ISBN:3-540-28793-0.

M. Gebauer, J. Saas, F. Sohler, J. Haag, S. Söder, M. Pieper, E. Bartnik, J. Beninga, R. Zimmer, T. Aigner:
Comparison of the Chondrosarcoma Cell Line SW1353 with Primary Human Adult Articular Chondrocytes Regarding their Gene Expression Profile and their Reactivity to IL-1β. Osteoarthritis and Cartilage, 13(8):697-708.

2004 Thomas Aigner, Eckart Bartnik, Florian Sohler, Ralf Zimmer:
Functional Genomics of Osteoarthritis: On the Way to Evaluate Disease Hypotheses. Clin Orthop. 2004 Oct; 1(427S):S138-S143.

Florian Sohler, Daniel Hanisch, Ralf Zimmer:
New methods for joint analysis of biological networks and expression data. Bioinformatics. 2004 Jul 1;20(10):1517-21.

Daniel Hanisch, Florian Sohler, Ralf Zimmer:
ToPNet - an application for interactive analysis of expression data and biological networks. Bioinformatics 20(9):1470-1.