CoryneRegNet

A reference database and analysis platform for corynebacterial gene regulatory networks

Jan Baumbach

21. Januar 2008

Dissertationsschrift zur Erlangung des Grades eines Doktors der Naturwissenschaften an der Technischen Fakultät der Universität Bielefeld

Acknowledgment

This PhD thesis could not have been written without the support of a lot of people.

First of all, I would like to thank Prof. Sven Rahmann, Dr. Andreas Tauch, Prof. Jens Stoye, and Prof. Alfred Pühler for supervising, for a lot of very helpful discussions, and simply all their support at all levels.

Furthermore, I would like to express thanks to Prof. Ralf Hofestädt for leading the examination board, both Prof. Ralf Hofestädt and Dr. Jacob Köhler for helping with my application for the International NRW Graduate School in Bioinformatics and Genome Research and for their advises with other strategic decisions. I wish to acknowledge Dr. Thoralf Töpel for his survey and for participating in the examination board.

In particular I express many thanks to my office neighbor Tobias Wittkop for joint cooperations, very helpful discussions, and for listening to my problems.

For the very effective and successful cooperation, many thanks go to Karina Brinkrolf, who performed the sister project to CoryneRegNet at the biological, wet lab side.

Certainly, Dr. Dirk Evers, Silke Kölsch, and the whole Graduate School is gratefully acknowledged not just for financial support but also for all their help with organization and strategic decissions.

Especially Ralf Nolte, but also the whole CeBiTec computer support team is acknowledged for technical support.

I am grateful to Heiko Neuweger, Dr. Michael Dondrup, Dr. Jörn Kalinowski, Dr. Alexander Goesmann, and Dr. Andrea Hüser for helpful discussions.

In the framework of this thesis, I supervised several Bachelor theses and wish to thank the following students for their help with CoryneRegNet: Jochen Weile, Jessica Schneider, and Katrin Rademacher.

I thank Oliver Röttger, Sven Sandow, Sita Lange, Tobias Wittkop, Josch Pauling, and my dad for proof reading of the manuscript.

This work would not have been possible without the support of my best friends. Hence, I wish to thank Alexander, Aurelie, Burkhard, Christian, Hannes, Josch, Oliver, Pascal, Rainer, Stefan, and Tobias.

Of course I am very grateful to my whole family (especially my grandparents) and to my girlfriend Birte.

Last but not least, I wish to thank my parents exceedingly. I know that I owe them very much.

Contents

Summary

Detailed information on DNA-binding transcription factors (the key players in the regulation of gene expression) and on transcriptional regulatory interactions of microorganisms deduced from literature-derived knowledge, computer predictions, and global DNA microarray hybridization experiments, has opened the way for the genome-wide analysis of transcriptional regulatory networks. The large-scale reconstruction of these networks allows the *in silico* analysis of cell behavior in response to changing environmental conditions. Here, we mainly focus on the gene regulatory interactions of corynebacteria, which are relevant in biotechnological production processes and human medicine.

In the framework of this thesis, we aim to a user-oriented software platform that supports (i) the integration of existing knowledge, (ii) visualization capabilities, (iii) the generation of novel hypotheses, and (iv) the possibility to share post-processed data with others.

Until now, there is no online database available that provides well-structured data on corynebacterial gene regulatory networks. Five related systems, which are specialized for other species are analyzed regarding their advantages and disadvantages. Not one of these databases provide all the necessary methods to satisfactorily support the above mentioned data processing tasks. None of the systems provide sufficient data exchange methods, statistically sound transcription factor (TF) binding site (TFBM) predictions, or comparative network analyses. Just one online database provides network visualization capabilities, none an appropriate homology detection. All platforms contribute to the basic requirements (i) data integration, (ii) raw data access, and (iii) graphical genome browsing. Nevertheless, data access is often difficult and the generation of novel hypotheses is not adequately addressed.

In this thesis, we present the corynebacterial reference database and analysis platform CoryneRegNet, which outperforms the other approaches at several levels. It offers a novel data structure to overcome typical data integration problems. First, CoryneRegNet was designed for the non-pathogenic soil bacterium *Corynebacterium glutamicum*. By using our ontology-based back-end, it could easily be extended by gene regulatory data on *Corynebacterium efficiens*, the human pathogens *Corynebacterium diphtheriae*, *Corynebacterium jeikeium*, *Mycobacterium tuberculosis*, and the model organism *Escherichia coli* K-12. Similar to the other platforms, CoryneRegNet provides a web-based user interface to access the database content. Aside from this and in contrast to the related platforms, CoryneRegNet offers a structured data exchange method for subsequent analyses by means of a SOAP-based Web Service server. With PoSSuMsearch, we integrated a fast method for TFBM predictions, which also provides statistically sound significance values for putative hits. In contrast to all related platforms, the GraphVis feature allows (i) the visualization of reconstructed gene regulatory networks as graphs, (ii) the interspecies comparison of these graphs, and (iii) the projection of gene expression data onto these graphs. Unlike the other systems, CoryneRegNet is directly connected to a genome annotation system that

provides additional up to date information for selected genes and proteins. Contrary to the related platforms, CoryneRegNet also supports (i) an appropriate homology detection, (ii) a method to analyze regulatory networks in the context of gene expression studies to predict putative contradictions, and (iii) a fast method for the automatic readjustment of often inaccurately determined TFBMs.

With CoryneRegNet, biological data on transcriptional gene regulations in microorganisms can easily be utilized for the generation of novel hypotheses and further bioinformatics analyses. We demonstrate these applicabilities by means of four application cases, which can not be directly addressed with other existing platforms.

CoryneRegNet is a comprehensive system for the integrated analysis of procaryotic gene regulatory networks. It is a versatile systems biology platform to support the efficient and large-scale analysis of transcriptional regulation of gene expression in microorganisms. CoryneRegNet is publicly available at `http://www.coryneregnet.de` or via the CoryneCenter portal at `http://www.corynecenter.de`.

1 Introduction

1.1 Background & motivation

Microorganisms continuously have to handle changing environmental conditions to maintain their functional homeostasis and to overcome stress situations with detrimental consequences for growth and survival [87]. Therefore, they evolved mechanisms to sense alterations within their environmental surroundings and developed molecular strategies co-ordinated by complex transcriptional regulatory networks to manage unfavorable conditions. The complexity of such regulatory networks results from the interaction of numerous transcription units consisting of a transcription factor (TF) and a defined set of regulated target genes [125]. The most important components of these units are apparently the DNA-binding transcription factors. They are responsible for sensing environmental and intracellular signals to control cellular reproduction and growth [4–6]. Depending on the growth conditions of a bacterial cell certain fractions of the total set of transcription factors are operating [110]. Some of them only control the expression of a single gene whereas others organize the activation or repression of numerous target genes [125].

Transcription factors include a DNA-binding domain that possesses a secondary structure to recognize the operator sequences of regulated genes [100] (refer to Figure 1.1). These sequences are more or less conserved (refer to Figure 1.2). In the following we denote such a nucleotide sequence as transcription factor binding motif (TFBM or shortly BM). The by far most widely used model to describe a set of BMs for a given TF is a position frequency matrix (PFM). PFMs can be converted to position weight matrices (PWMs), also called position-specific score matrices (PSSMs) by taking log-odds (see e.g. [107, 121]). In turn the PWMs may be used to scan the upstream sequences of putative target genes in order to predict novel TF-DNA interactions *in silico*. In Section 3.3 and Section 3.4 we give more detailed information on PFMs, PWMs, and PWM-based TFBM predictions.

The availability of whole genome sequences provides the opportunity to define the total set of DNA-binding transcription factors of an organism [20, 102]. This is a first step not only in understanding the regulatory complexity of a certain bacterial cell but also for reconstructing the global connectivity of a regulatory network to theoretically describe and deduce gene expression pattern of a microorganism [60]. From a set of complete genome sequences it has been deduced that large genomes include more transcription factors per gene than small genomes [24]. The increase of genomic complexity is thus associated with a more complex regulation of gene expression since the additional genetic information has to be integrated into the existing regulatory network basically operating in a bacterial cell. The transcriptional regulatory network of *Escherichia coli* K-12 so far is one of the best characterized regulatory systems of a single cell. The total number of about 320 transcriptional regulators of *E. coli* were classified into eight distinct regulatory modules with defined physiological functions [110]. Additional bioinformatics studies suggested a

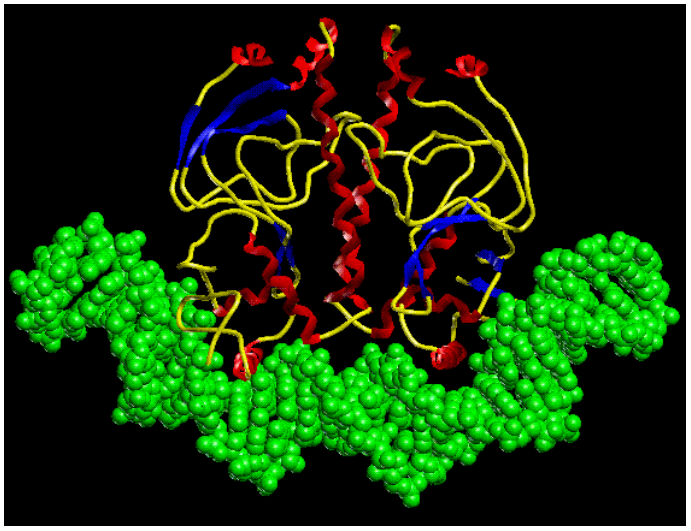


Figure 1.1: The secondary structure of the human transcription CAP-DNA complex (1ber). Taken from http://gibk26.bse.kyutech.ac.jp/jouhou/image/dna-protein/rna/rna.html.

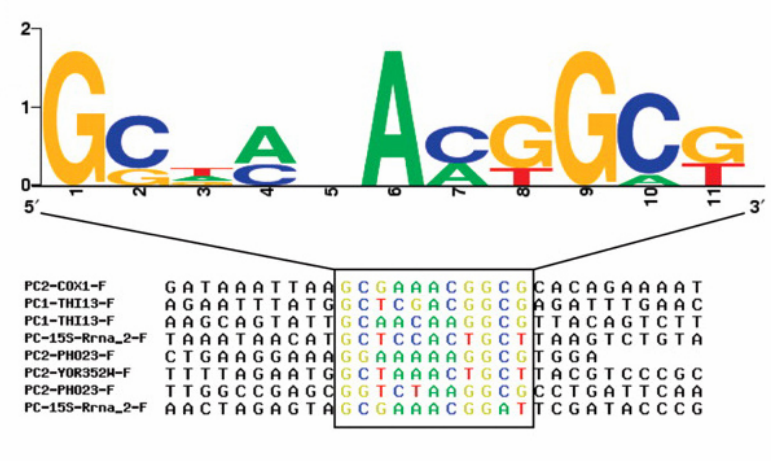


Figure 1.2: The transcription factor binding sites of a regulator can slightly vary for different target genes. Taken from http://weblogo.berkeley.edu.

hierarchical and modular structure of the regulatory network, excluding circular feedback loops on transcriptional level for this organism [8, 83, 84].

The genus *Corynebacterium* comprises a number of human pathogens, like *Corynebacterium diphtheriae* and *Corynebacterium jeikeium*, as well as the non-pathogenic soil bacteria *Corynebacterium glutamicum* and *Corynebacterium efficiens* that are widely used in biotechnological production processes of food and feed additives [47, 59]. Because of their relevance in biotechnology and medicine the genome sequences of *C. glutamicum* ATCC 13032, *C. efficiens* YS-314, *C. diphtheriae* NCTC 13129, and *C. jeikeium* K411 have recently been determined [25, 67, 97, 124]. First comparative analyses revealed a high-level conservation of orthologous genes in these genome sequences, indicating that the corynebacterial species have rarely undergone genome rearrangements and thus largely retained their ancestral genome structure [94]. An initial step in understanding the transcriptional regulatory machinery of corynebacteria was the bioinformatics identification of the encoded transcription factors [20]. A collection of 127 DNA-binding transcription factors was detected in the genome sequence of *C. glutamicum*, whereas 103 regulators were identified

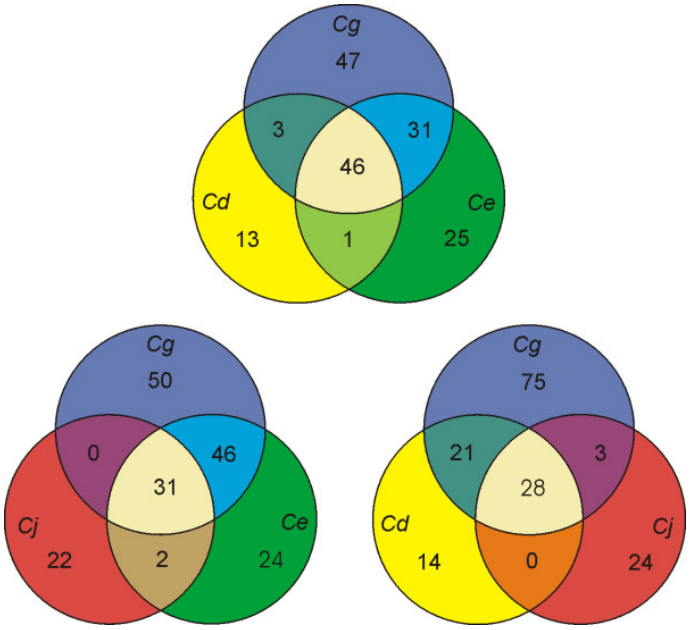


Figure 1.3: A comparison of the content of transcription factors in four completely sequenced corynebacterial genomes. The Venn diagrams show the number of shared and organism-specific genes among the genomes. Abbreviations: Cg, *C. glutamicum*; Ce, *C. efficiens*; Cd, *C. diphtheriae*; Cj, *C. jeikeium*. Taken from [20].

in *C. efficiens*, 63 in *C. diphtheriae* and 55 in *C. jeikeium*. The relation between these numbers agrees well with the assumption that the quantity of transcription factors of an organism is correlated to the genome size and the environmental surrounding a bacterial cell is exposed to [24]. Accordingly, the physiological versatility of *C. glutamicum* results in a considerably higher number of transcriptional regulators, and in consequence in a more complex regulatory network by integrating and co-ordinating additional regulatory subnetworks. According to amino acid comparisons and protein structure predictions the repertoire of DNA-binding transcription factors of *C. glutamicum*, *C. efficiens*, *C. diphtheriae*, and *C. jeikeium* were further on divided into 25 families of regulatory proteins. A common set of only 28 regulators was encoded by all of the four genome sequences and thus presumably includes the core set of DNA-binding transcription factors of these bacteria [20] (refer to Figure 1.3). Despite the progress in bioinformatics prediction of transcription factors, the reconstruction of regulatory networks is generally hindered by the relatively low level of evolutionary conservation of other molecular network components, for instance of the cognate operator sequence (binding motif) of a DNA-binding transcription factor. However, developments in DNA microarray technology have allowed the generation of genome-wide data sets experimentally characterizing the regulatory networks of corynebacteria [61, 75, 111].

1.2 Aims

The ambition of this post-genomic approach is first to decipher and reconstruct the transcriptional regulatory network of *C. glutamicum* as a model organism and subsequently the networks of *C. diphtheriae*, *C. efficiens*, and *C. jeikeium*. The goal is to develop an online available database and analysis platform for corynebacterial gene regulatory networks that

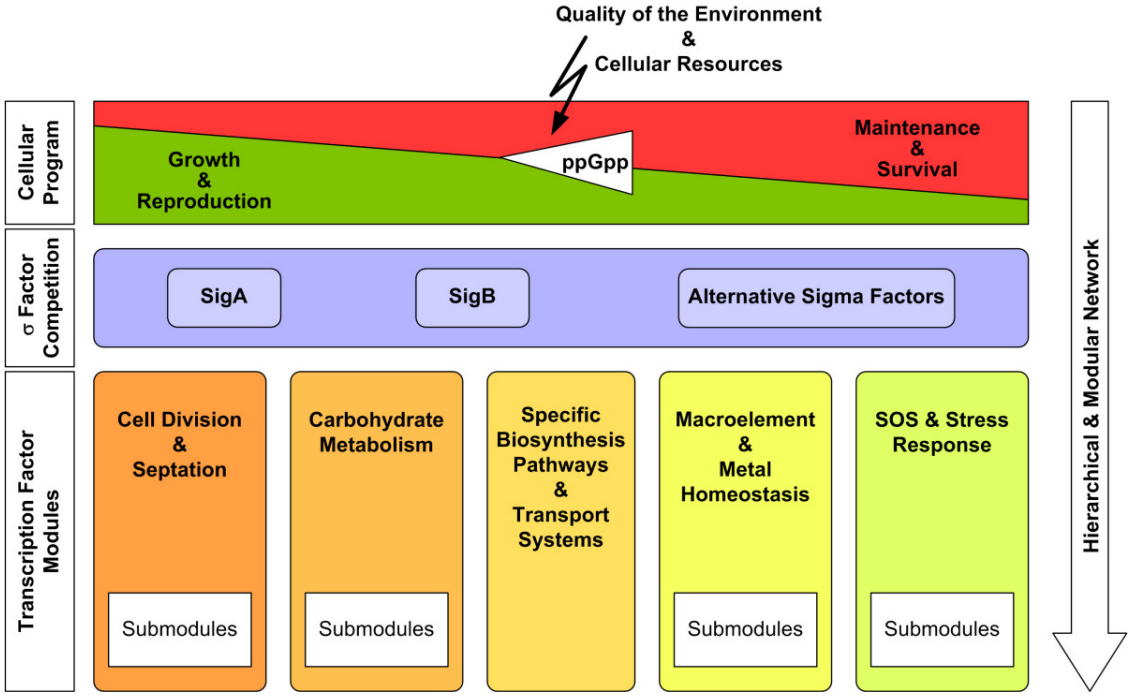


Figure 1.4: The biological concept of gene regulation in *C. glutamicum*. The model presents the hierarchical and modular network structure of transcriptional regulatory interactions. It consists of five distinct transcription factor modules and a module containing the main and alternative sigma factors involved in differential gene expression by sigma factor competition. A top level regulator is the hyperphosphorylated guanosine nucleotide ppGpp, involved in sensing the quality of the environment and the cellular resources. The amount of ppGpp determines the cellular program and the role of sigma factor competition in global regulation of gene expression. Taken from [11].

reflects the biological concept of these organisms, as suggested in [11] (refer to Figure 1.4). The platform shall provide a solid basis for further regulatory network studies in the field of systems biology.

In other words, within this thesis, we aim to a user-oriented software platform that supports (i) the integration of existing knowledge, (ii) visualization capabilities, (iii) the generation of novel hypotheses, and (iv) the possibility to share post-processed data with others. To address these points, we developed CoryneRegNet, an ontology-based data warehouse and analysis platform of corynebacterial transcription factors and gene regulatory networks.

1.3 Structure

In Chapter 2, we first introduce and compare related systems with respect to the database content and to analysis features. We show that none of the existing platforms provide (i) data on corynebacteria and (ii) sufficient data analysis capabilities. We discuss why another platform is necessary. Subsequently, we describe which further visualization and functional capabilities are required for (i) an integrated systems biology analysis of existing knowledge, (ii) for the *in silico* generation and evaluation of new hypotheses, and (iii) for the potential to share data with other platforms in a well-structured manner.

Afterwards, we describe how we addressed these requirements and how we implemented

them in the systems biology platform CoryneRegNet in Chapter 3.

Here we first present the data integration procedure (Section 3.1). We start with the system architecture and give technical details on the used libraries (Section 3.1.1). The novel ontology-based data structure that helps to overcome the typical data integration problems of the other related platforms is described in Section 3.1.2. An introduction to Web Services and how CoryneRegNet and the systems biology community profits from that technology is given in Section 3.1.3. Here we show how post-processed data can be shared with other platforms in a well-structured way.

We explain the visualization functionalities of CoryneRegNet in Section 3.2 and this is subdivided into two parts. First, the main web interface is described in Section 3.2.1. Similar online representations are also provided by related systems, excluding the interfaces to specific data analysis features, which are not offered by the others. In the second part, the network visualization toolkit GraphVis is introduced (Section 3.2.2). A similar graph visualization and analysis capability is not provided by other platforms.

In Section 3.3 we briefly introduce the fast and statistically sound TFBM prediction software PoSSuMsearch and describe how it has been included into CoryneRegNet's back-end and front-end. With its integration, we provide an easy-to-use interface that helps to generate novel hypotheses.

Since an accurate determination of TFBMs is essential for further predictions, in Section 3.4 we present a method for the automatic readjustment of TFBMs. On top of the PoSSuMsearch software, our approach provides faster and more accurate solutions than existing platforms.

How one can use gene expression studies to detect putative contradictions or inconsistencies in gene regulatory networks is described in Section 3.5. The corresponding COMA feature is explained in detail. We discuss how the systems biology community can profit from this functionality that is also not supported by related platforms.

A prerequisite for knowledge transfer from one model organism to other organisms is a fast and accurate genome-scale homology prediction. In order to integrate such data into CoryneRegNet, we developed a novel strategy for protein homology detection solely based on the amino acid sequences. In Section 3.6 we present the software FORCE that heuristically solves the weighted graph cluster editing problem. Subsequently, we demonstrate its application to huge datasets and how it has been integrated into CoryneRegNet. We show that our approach outperforms the most popular protein clustering tools.

Integrated systems biology platforms like CoryneRegNet or related systems are growing software projects. CoryneRegNet was also subjected to continuous improvement. In Section 3.7 we summarize the change in database content and the development of CoryneRegNet from the first release 1.0 to the current version 4.0. The ontology-based data structure and the generic system architecture of CoryneRegNet helped to keep negative effects that arouse from typical extension problems to a minimum.

Chapter 4 summarizes the results of this thesis. We discuss how CoryneRegNet meets the requirements analyzed in Chapter 2. In Section 4.1 we demonstrate the key features of CoryneRegNet by means of four application cases, which can not be directly addressed with other existing platforms: (i) the reconstruction of the stress response of *C. glutamicum* at transcriptional level in Section 4.1.1, (ii) the knowledge transfer from *C. glutamicum* to *C. diptheriae* regarding the well-studied regulator DtxR in Section 4.1.2, (iii) the compar-

ison of the regulatory LexA network of *C. glutamicum* and *E. coli* in Section 4.1.3, and (iv) the study of the transcriptional response of *C. glutamicum* to two different feeding conditions by using gene expression data in Section 4.1.4.

We conclude that CoryneRegNet is a comprehensive systems biology platform for the storage, visualization, reconstruction, and analysis of procaryotic gene regulatory networks in Chapter 5. It outperforms other systems with related aims.

1.4 Availability

CoryneRegNet is publicly available at `http://www.coryneregnet.de`. A documentation on how to develop a CoryneRegNet Web Service client is also available at the web site. CoryneCenter can be entered via the portal web site `http://www.corynecenter.de`. The protein cluster software FORCE including the source code and all used datasets can be downloaded from `http://gi.cebitec.uni-bielefeld.de/comet/force/`. The transcription factor binding motif reannotation web server MoRAine is online available at `https://www.cebitec.uni-bielefeld.de/groups/gi/software/coryneregnet/moraine/`.

1.5 Publications & cooperations

The database itself, some basic visualization features, and a description of the back-end and the data import process has been published in [11] (CoryneRegNet 1.0). Following this, the database content has been extended considerably and several data analysis features have been developed and integrated. The subsequent releases 2.0, 3.0, and 4.0 are published in [10, 12, 14]. The articles [10, 11] are 'highly accessed' since publication at the BioMed Central (BMC) web site. Finally, CoryneRegNet has been interconnected with GenDB and EMMA. The comprehensive data analysis platform CoryneCenter was established and published in [96]. The ontology-based database back-end was initially developed for the ONDEX system, which is published in [71], while the large-scale protein cluster detection software FORCE was published in [108,131]. A manuscript, which describes the MoRAine web server has been submitted for publication [15].

This integrated systems biology project has partially been performed in cooperation with other scientists at the Center for Biotechnology, Bielefeld. Karina Brinkrolf worked on the sister project of CoryneRegNet and performed wet lab experiments. FORCE has been developed in joint cooperation with Tobias Wittkop. A detailed description of all cooperations is given in the Appendix in Section A on page 96.

2 Requirements and related work

In the past years, several approaches for the storage, analysis, and reconstruction of gene regulatory interactions have been implemented and established. This chapter gives a brief introduction to platforms that are related to our aims. We discuss the advantages and disadvantages of these systems. From this analysis, the requirements for CoryneRegNet are deduced.

2.1 Related platforms

2.1.1 RegulonDB

RegulonDB is an internationally recognized and established reference database for the procaryotic model organism *Escherichia coli* K-12. The provided amount of manually curated and experimentally validated knowledge on the gene regulatory network and the operon organization of *E. coli* is the largest currently available for any organism. The current release 5.0 is synchronized with a second *E. coli* reference database: EcoCyc [70], where the same information is offered. All data is gathered manually by the RegulonDB curation team. Starting point is a list of publication abstracts, obtained from PubMed and filtered by pertinent keywords. The manually extracted data on transcriptional regulatory interactions is stored in an Oracle DBMS and continuously checked for inconsistencies.

An online interface allows querying the database content. For a gene of interest, all annotated data is presented (gene product, position in the genome, molecular weight, functional classification, and access to the corresponding gene/protein sequence). Furthermore, all known gene regulatory interactions are given along with a list of co-regulated genes. Moreover, a visualization of the genomic context is offered, including the operon organization, binding sites, promoters, and terminators (genome browser). A network display tool allows a simple, circular, and graph-based visualization of the immediate neighbors of the gene of interest within the global regulatory network (refer to Figure 2.4 on page 19). Moreover, RegulonDB integrates the Nebulon-tool [66], which predicts sets of functionally related genes (clusters of potentially homologous proteins). It is based on the co-occurrence of genes within bacterial operons. RegulonDB is publicly available at `http://regulondb.ccg.unam.mx/`. It provides no data exchange methods, but a download of tab-delimited flat-files [114].

2.1.2 MtbRegList

MtbRegList is a database dedicated to the gene regulation of the human pathogenic bacterium *Mycobacterium tuberculosis*. Its back-end stores 121 predicted and experimentally validated 'regulatory DNA motifs' (transcription factor binding motifs) along with a frequently updated genome annotation from the TubercuList database [23].

As for RegulonDB, an online interface allows querying the database content. For a gene of interest, the gene annotation is available (gene product, position in the genome, and access to the corresponding gene/protein sequence); obtained from TubercuList and NCBI [127]. Furthermore, for all genes the web interface provides hyperlinks to corresponding COG database entries [123], where possible. Also, similar to RegulonDB, one can graphically navigate the genomic context, given a gene or a genetic region as a starting point (genome browser). A network visualization is not supported. A method to search for TFBMs stored in the database exists. The user can enter a 'signature' (similar to a regular expression) to retrieve a list of potentially matching motifs. MtbRegList is publicly available at `http://pages.usherbrooke.ca/gaudreau/MtbRegList/www/index.php`. It also does not provide data exchange methods, but search results can be downloaded in XML or tab-delimited text format [65].

2.1.3 PRODORIC

The database PRODORIC generally aims to the storage and analysis of procaryotic gene regulations. Similar to RegulonDB and MtbRegList, all data is gathered by analyses of scientific literature and subsequently stored in a database, which is based on the TRANSFAC database (see Section 2.1.5), but is extended to specific procaryotic characteristics. PRODORIC includes all NCBI genome annotations of procaryotic organisms even if no information on any transcriptional regulation is available. Mainly *Bacillus subtilis*, *E. coli*, and *Pseudomonas aeruginosa* are supported.

As the other systems, PRODORIC also provides a web interface for querying the database content, and moreover it allows to execute analysis features. Aside from the integrated NCBI data, PRODORIC supports links to COG and to SWISS-PROT [7]. For a gene of interest, the gene annotation is available (gene product, position in the genome, and access to the corresponding gene/protein sequence), along with its regulators (the TFs that control the gene) the corresponding TFBMs are available, which are subsequently used to construct position weight matrices (PWMs). In total, PRODORIC covers regulatory information on six organisms (with 2517 TFBMs and 149 PWMs). The PWMs can subsequently be used as input for the integrated TFBM matching software Virtual Footprint [92] to predict further TF-DNA interactions. As in RegulonDB, a genome browser provides a graphical representation of the genomic context at sequence level. A network visualization is not supported. PRODORIC is publicly available at `http://www.prodoric.de`. As the other systems, it does not provide data exchange methods. Not even data download as flat-files is supported [93].

2.1.4 DBTBS

DBTBS is the database of transcriptional regulation in *Bacillus subtilis*. It is essentially a compilation of transcription factors with their regulated genes as well as their recognition sequences (TFBMs), which were experimentally characterized and reported in the literature. Annotated genes are linked to the Japanese BSORF database [44]. DBTBS also supports the prediction of putative TFBMs within a given input sequence by using PWMs and consensus patterns. Furthermore, DBTBS contributes to comparative genomics by detecting potentially orthologous transcription factors in other procaryotic genomes. It

provides a genome browser but no network visualization capabilities. DBTBS is publicly available at `http://dbtbs.hgc.jp`. Again it does not provide data exchange methods. Data download as flat-files is also not supported [64, 85].

2.1.5 TRANSFAC

Although TRANSFAC is a commercial platform, which is distributed by BIOBASE (`http://www.biobase.de`) and focuses on eucaryotic organisms (human, mouse, *Arabidopsis thaliana*, *Drosophila melanogaster*, and *Saccharomyces cerevisiae*), it shall be introduced here because it provides similar data analysis functionalities as the above mentioned systems.

The data is manually curated and gathered by analyzing scientific literature. TRANSFAC stores TFs, genes, and TFBMs. The entries presented at the web site are hyperlinked to corresponding databases (NCBI, DATF: Database of *Arabidopsis* Transcription Factors [56], Drosophila DNase I Footprint Database [18], and FlyBase [37]). For a given TF, all genes that are under direct transcriptional control of the TF are presented along with the corresponding TFBMs. These motifs are subsequently used to construct PWMs. As for PRODORIC, the PWMs can subsequently be used as input for the integrated motif matching software features MATCH [69], and P-MATCH: [27]. TRANSFAC does not provide a network visualization. Unlike the afore mentioned systems, a genome browser is also not directly integrated. TRANSFAC is available online at `http://www.gene-regulation.com` under a commercial license. Again, this system does not provide any data exchange methods [88, 129, 130].

2.1.6 Summary

Here we summarize the related platforms by means of a compacted view on the database content and on the analysis features.

Database content

The afore mentioned databases store data on gene regulations for the following organisms:

- RegulonDB: *Escherichia coli* K-12
- MtbRegList: *Mycobacterium tuberculosis* H37Rv
- PRODORIC: *Bacillus subtilis*, *Escherichia coli* K-12, *Pseudomonas aeruginosa* ATCC 15692, and *Pseudomonas aeruginosa* PAO1
- DBTBS: *Bacillus subtilis*
- TRANSFAC: *Homo sapiens* (human), *Mus musculus* (mouse), *Arabidopsis thaliana*, *Drosophila melanogaster* (fruit fly), and *Saccharomyces cerevisiae* (yeast)

Note that we exclude those organisms from consideration where just a few gene regulatory interactions are available (this solely effects PRODORIC). Not one of the systems provide data on that species this work aims to: corynebacteria and closely related microorganisms.

Data analysis features

Table 2.1 compares and summarizes the main data analysis and visualization features of the related platforms. We will use this summary to compile a list of necessary and desirable functionalities for our system in the next section.

Table 2.1: Summary and comparison of related database systems and analysis platforms, which are dedicated to gene regulation. Note that raw data access in MtbRegList is granted by means of XML, but separate for each entry.

Feature	RegulonDB	MtbRegList	PRODORIC	DBTBS	TRANSFAC
Genome browser	+	+	+	+	
Network visualization	+				
Network analysis					
Raw data access	+	(+)			+
Data exchange methods					
BM prediction		+	+	+	+

2.2 Requirement analysis

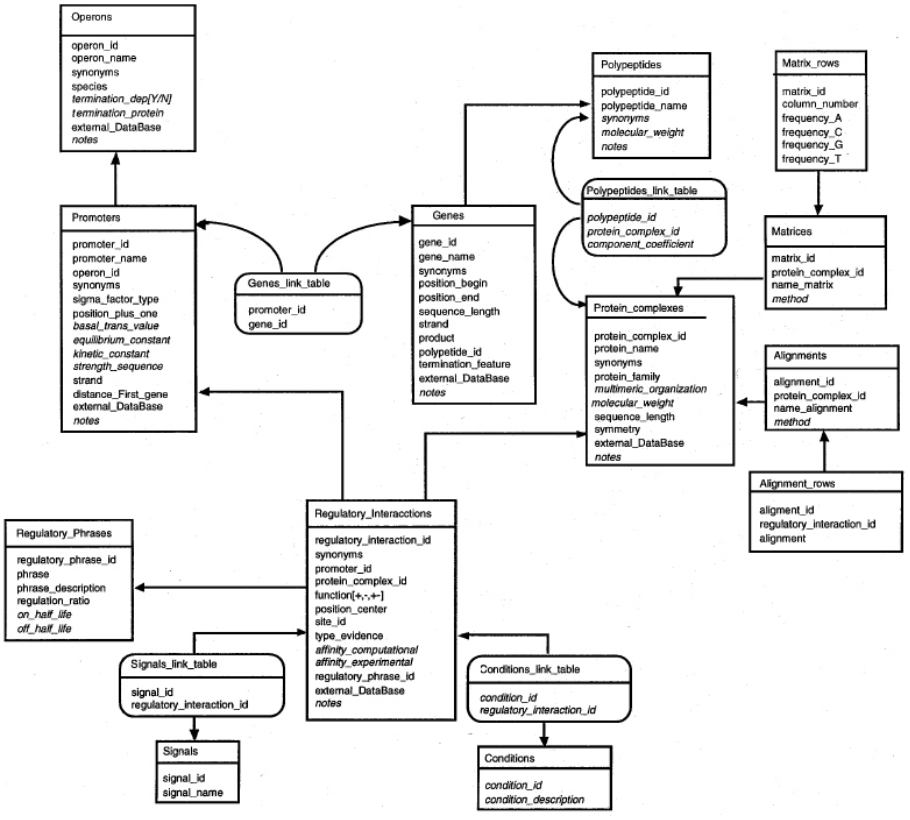


Figure 2.1: Overview of the data structure of the RegulonDB 1.0 database back-end. Taken from [62]

All the databases use special purpose data structures at the back-end level. Figure 2.1 shows an overview of the data structure of the first release of RegulonDB [62]. The continuously growing amount of available data and novel models of gene regulation have caused

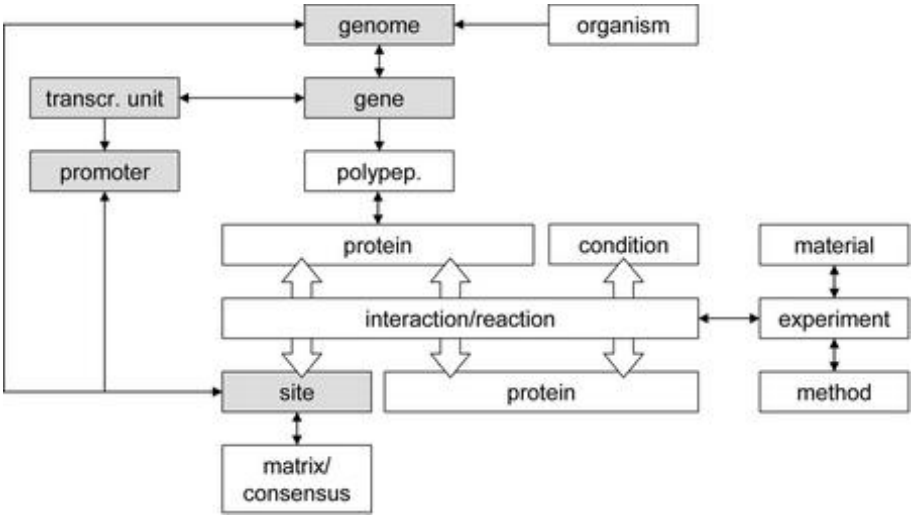


Figure 2.2: Overview of the data structure of the PRODORIC database back-end. Taken from [93].

.

the necessity to adapt these structures. Between 1998 and 2006, the data structure of RegulonDB has been changed and extended several times in order to fit the novel knowledge into the specialized data structure. These stepwise changes are reported in [113–117]. The back-end organization of the PRODORIC database is shown in Figure 2.2.

Both the schemata of RegulonDB and PRODORIC illustrate the high specialization of the data structures. Even though both systems are specialized for the same purpose, namely modeling procaryotic gene regulation, two circumstances effect the transfer of real-world models to *in silico models*. (i) The database designers act on slightly different assumptions about the nature of the real-world features that have to be modeled. This partially depends on the topic-specific know-how at that point of time when the data structures are formulated. (ii) When new information and novel experimental results on the modeled concepts are available the designers have to adapt the database back-end in a way to fit the new concepts into the schemata. This mainly entails modifications in both the import process of the data into the database and the querying procedure. Furthermore, the front-end has to be altered in order to query, display, and analyze the novel information. To overcome these problems, we propose to use an ontology-based data structure that is explained in Section 3.1.2 on page 23.

From the summary in Table 2.1 (page 16) we can gather direct impressions of the basic functionalities, not including obvious standard capabilities:

- Genome browser: A genome browser visualizes the genomic context of a gene of interest, ideally along with known sequence features (TFBMs, gene start/stop positions, etc.). All the afore mentioned databases make use of an own implementation that is specialized for the specific database structure. A graphical comparison of genome browsers (including the CoryneRegNet visualization described in Section 3.2.1) is given in Figure 2.3. All except for the PRODORIC viewer also visualize TFBMs and the operon structure. The browsers of RegulonDB and MtbRegList support image maps, which allow the user to directly navigate to sequence features by clicking on them. TRANSFAC does not provide a genome browser. We address this point in Section 3.2.1 on page 27.

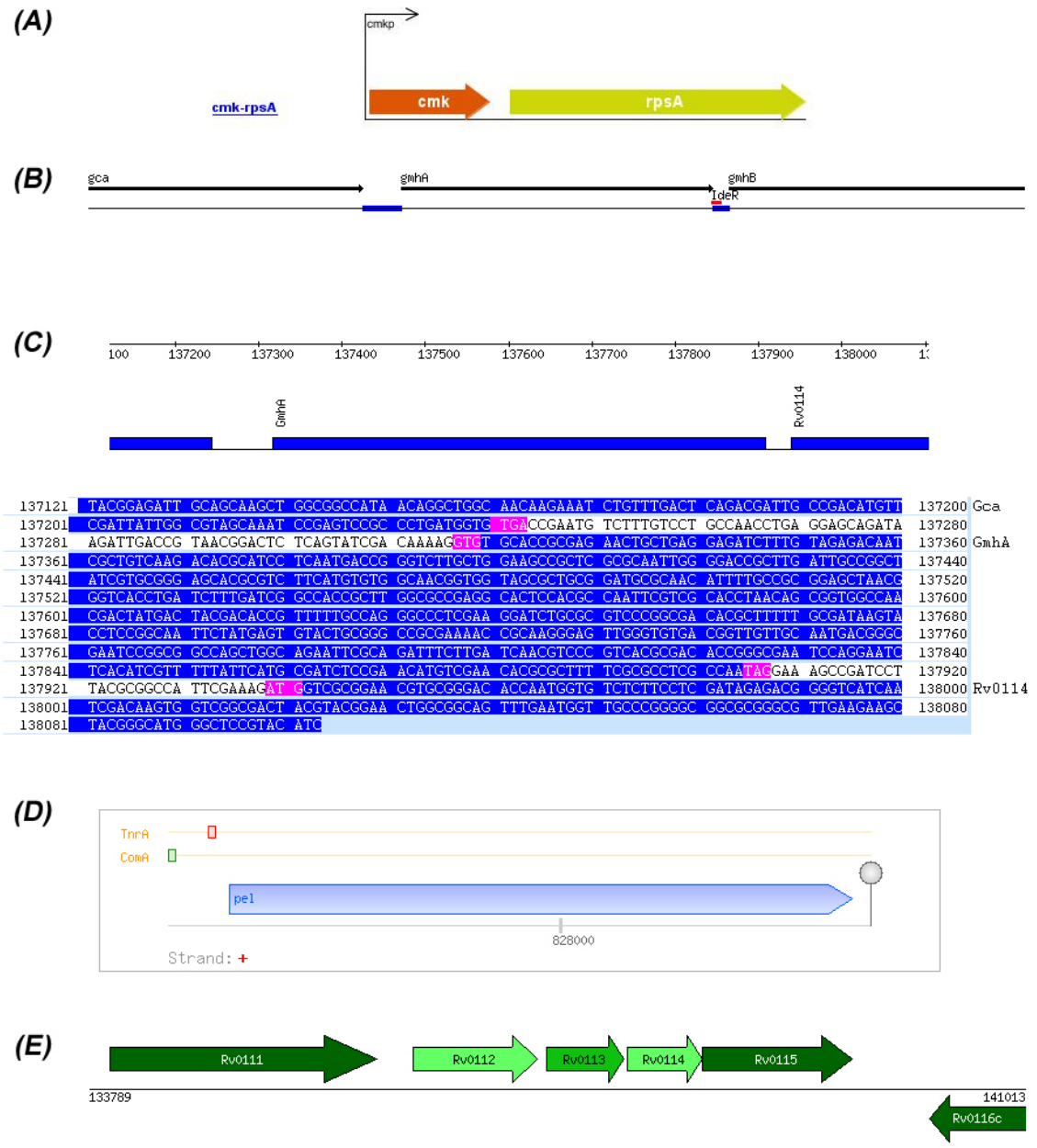


Figure 2.3: This figure illustrates the genome browsers of RegulonDB (A), MtbRegList (B), PRODORIC (C), DBTBS (D), and CoryneRegNet (E). The screenshots have been taken from the corresponding web sites (see text). Displayed genes: *cmk-rpsA* of *E. coli* in (A); *gmhA* of *M. tuberculosis* in (B), (C), and (E); *pel* of *B. subtilis* in (D).

- Network visualization: If we consider genes as nodes and transcriptional interactions between genes as labeled, directed edges, one can imagine networks (graphs). These can be visualized by utilizing adequate graph layout algorithms. Figure 2.4 shows the network visualization of RegulonDB, the only platform that supports such a capability. Unfortunately, the graph layout can not be changed (e.g. to reflect the hierarchical network structure). Furthermore, the genes (nodes) in the graph can not be navigated by clicking on them. In Section 3.2.1 (page 27) we present a considerably improved graph visualization feature.

- Raw data access: The provided data normally is stored using a relational database

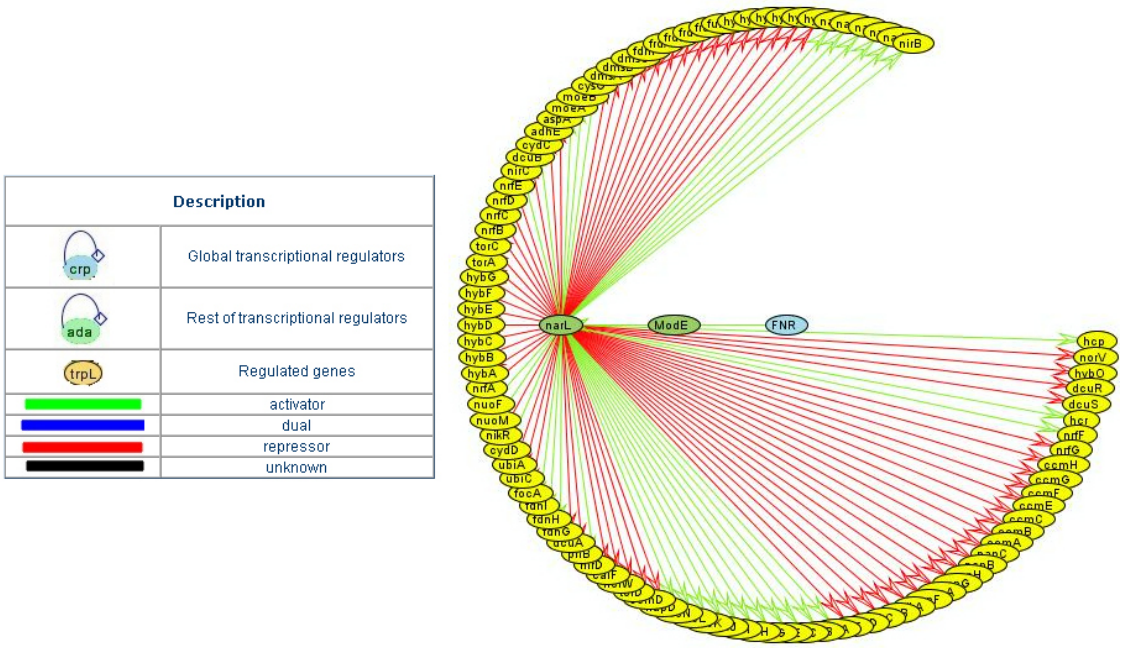


Figure 2.4: This screenshot shows the gene regulatory network of the nitrate regulator NarL in *E. coli* visualized with RegulonDB.

management system (e.g. MySQL, PostgreSQL). This supports an efficient data processing but hinders the direct access of external users. Therefore, all datasets are extracted in (more or less) regular time intervals and stored as tab-delimited flat files, which are subsequently offered for download at the web site. We contribute to this point in Section 3.1.3 on page 25.

- BM prediction: The prediction of putative TFBMs allows (i) the identification of further gene regulatory interactions of a TF of interest, and thus (ii) the transfer of regulatory networks of a model organism to other closely related species. A variety of tools have been developed in the last years to attack this problem (for example Virtual Footprint and MATCH, mentioned in the previous section). These model BMs of a TF as PWMs and subsequently use them to scan the upstream sequences of putative target genes for matching sequence motifs. Although this is an important feature, all related platforms lack (i) a statistically sound but fast method and (ii) a method that helps to validate and readjust the often imprecise determined TFBMs. We present the integration of an improved BM prediction in Section 3.3 (page 35) and show its benefits by means of an application case in Section 4.1.2 (page 72). A suitable automatic BM reannotation method is presented and discussed in Section 3.4 (page 37).

Aside the functions provided by the already established systems, we consider the following features as valuable extensions to gene regulatory databases:

- Data exchange methods: Usually, interconnections between different data sources are realized by HTML-links to other web pages or by regular, manual downloads and a subsequent integration of the corresponding data. This is both time-consuming and error-prone. By integrating SOAP-based Web Services, the data can be post-processed easily and hence be presented in a different way, and, most importantly, it

is always up to date. Using Web Services, the user would not even recognize that the data is downloaded from another service. A publicly available Web Service server offers methods to query data from the database. Neither direct access to the DBMS server is necessary, nor any knowledge about the data structure in the back-end. All API information is provided as so-called Web Service Definition Language (WSDL)) files. In Section 3.1.3 on page 25 the benefits of a Web Service integration into CoryneRegNet are discussed in detail. We demonstrate these benefits in Section 4.1.4 on page 77 by means of an application example.

- Network analysis: Bacterial gene regulatory networks normally show a hierarchical structure that is mostly conserved between closely related species [4, 6, 82]. Hence, graph comparison capabilities (for both known and predicted networks) are a highly desirable feature. Such a function assists scientists with the cross-species knowledge transfer and hence with the identification of novel promising targets for wet lab analyses. On top of that, the projection of gene expression levels (measured e.g. with DNA microarrays) onto graphs helps to gain a first overview of experimental data. Simple contradictions or inconsistencies in the context of known or predicted gene regulatory networks could become obvious within seconds. We address this point in Section 3.2.2 (page 32). The network comparison capabilities as well as the projection of microarray results onto graphs are described in detail.

As consequences or enhancements of the afore mentioned requirements, we also consider the following points as being necessary:

- Homology detection: In order to provide graph comparison functionality at the front-end side, it is necessary to have a mapping between homologous proteins/genes. The simplest way would be the integration of corresponding data from the COG or the SCOP [3] databases. Unfortunately, both standard repositories omit data on corynebacteria and on specific mycobacterial strains. Hence, either the installation of an existing software or the development of a novel, easy-to-integrate method is desirable. In Section 3.6 (page 48) we address this point and present a novel homology detection method that outperforms other popular approaches.

- Contradictions/inconsistencies in gene expression experiments: Assume we have given a (possibly incomplete) gene regulatory network and the operon organization of an organism. Further, we have given a potentially genome-wide transcriptional expression study (e.g. a microarray experiment). Now, one can scan the experimental results for contradictions in the relative gene expression levels concerning (i) operons and (ii) the known regulatory network. For example, one could imagine that a repressing transcription factor is upregulated, but one of its target genes is not downregulated (as expected); that would be a hint for further (hidden or unknown) transcriptional regulatory relationships. In Section 3.5 on page 45 we present the COMA feature, which addresses this requirement.

3 CoryneRegNet

In this chapter, we describe CoryneRegNet, an ontology-based data warehouse that satisfies the requirements for a corynebacterial gene regulatory database and analysis platform that were outlined in the previous chapter.

3.1 Data integration

A prerequisite to systems biology is the integration of heterogeneous experimental and annotation data, which is stored in numerous life-science databases. Efficient data handling and integration is encumbered by a wide range of problems. Frequently, the explicit specification of data that should be integrated during future research and database development is not available beforehand. At present, most databases are implemented on relational database management systems; they store data by using specialized data structures, which leads to two problems during data processing.

1. Attribute names are often not self-explanatory and equivalent attributes have different names in different databases. Therefore, problems with attribute values might occur when using, for instance, unequal units in different data sources [76].

2. The main problem, however, affects the querying procedure, since it requires detailed semantic knowledge about the content of specific database tables. When extending an embedded database by new data, the data structure of the back-end needs to be changed, which usually affects the import procedures and the front-end applicability as well as the stability and integrity of the whole system [103].

In the case of the below mentioned systems GenDB and EMMA, a special interface called BRIDGE [51, 52] has been developed, to overcome the data exchange problems. A more general and more widely used and accepted technique is the application of SOAP-based Web Services. These have recently been implemented e.g. by the European Bioinformatics Institute [104] and BRENDA [9].

For CoryneRegNet, the starting point of data integration is a collection of different flat files that store the genome annotations, predicted operons, gene regulations and the membership of transcription factors to their families. In order to supply the researchers' need for structured information and appropriate visualization along with knowledge recombination for usage in further analyses and integrated easy-to-use bioinformatics methods, such as TFBM prediction, an important task is to find a way to import existing data into a single data scheme, with respect to expandability of the data repository by data of unknown and unspecified structure. Our approach to address the mentioned challenges converts all data sources into a common ontology-based, graph-like data structure, denoted integrated ontologies (also recently used in the ONDEX system [71]).

In the following, we first describe the overall system architecture of CoryneRegNet. Subsequently, we contribute to the special data structure of the database back-end and the benefits from integrating the genome annotation system GenDB [90] and the microarray storage and analysis platform EMMA [36, 79] into CoryneRegNet by using SOAP-based Web Services. Last, we describe how the systems biology community can profit from the integrated Web Service server.

3.1.1 System architecture

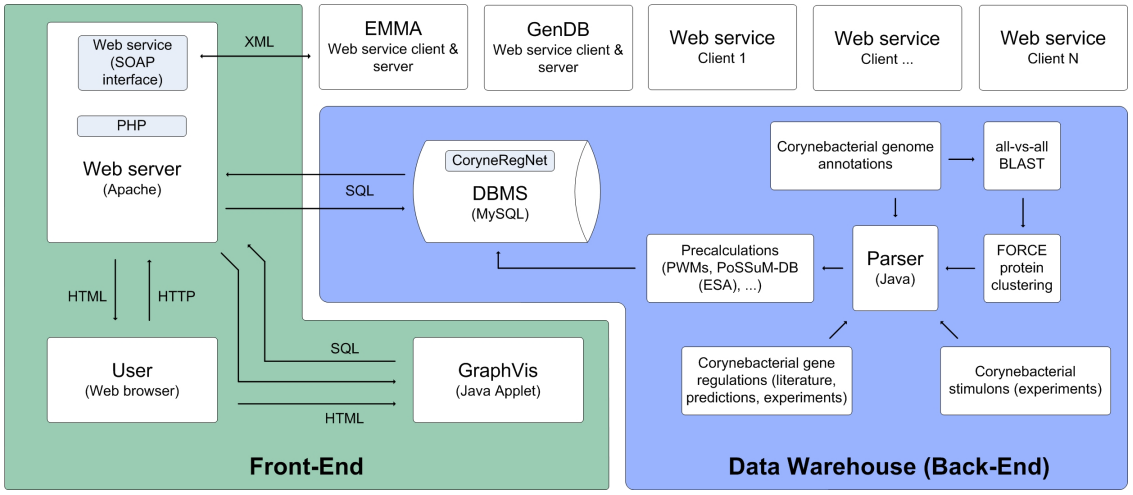


Figure 3.1: The system architecture of CoryneRegNet. Since it is a data warehouse, all time-consuming calculations are performed at data warehousing. The results are then transformed into an ontology-based data structure and imported to the MySQL database server (Back-End). An Apache web server processes the user requests, queries the database, and constructs the corresponding web pages. It further provides the SOAP-based Web Services for GenDB, EMMA, and other applications and also queries them as a client. A Java Applet is used for network visualization and analysis (Front-End).

Figure 3.1 illustrates the system architecture of CoryneRegNet, which is designed as a web-based software environment that is publicly available. The complete genome sequences of all integrated microorganisms along with the genome annotations have been downloaded from NCBI in GenBank format and imported into CoryneRegNet. Furthermore, biological data relevant to transcriptional regulations were imported into the database as derived from literature knowledge (included in the database as PubMed link), computer predictions, and experimental studies. The data import process was realized by running a parser that was implemented in Java. The parser software additionally integrates the imported data into a single ontology-based data structure and converts it into a relational data model. The output are tab-delimited flat-files that in turn are input files for the MySQL built-in import procedure and finally used to fill the CoryneRegNet database.

Since CoryneRegNet is a data warehouse, all time-consuming calculations are regularly performed at import process. First, the upstream region of each gene is extracted and stored in a separate table of the database and additionally in a flat file in FASTA format for integration with the PoSSuMsearch software. The CoryneRegNet import program includes all-vs.-all BLAST [2] results for all gene and protein sequences in the database using an E-value (expected number of higher scoring hits in random sequences) threshold

of 10^{-6} for genes and 10^{-10} for proteins. This calculation has to be performed just once when a new genome annotation is added and is also used as input for the protein cluster prediction software FORCE. The results are re-used in later import procedures, in contrast to the PoSSuMsearch suffix array, which has to be re-created during every data warehousing process because the upstream sequences that are used in this pre-calculation step depend on the imported operon tables. In addition, all PWMs are recalculated.

For the web front-end PHP 5 (`http://www.php.net`) is used. It runs on an Apache server 2.0.49 (`http://www.apache.org`), which queries the database management system MySQL 4.1.9 (`http://www.mysql.org`). All diagram graphics are created with JpGraph 1.20.3 (`http://www.aditus.nu/jpgraph`) and GD Graphics Library 2.0 (`http://www.boutell.com/gd`). Operon information for corynebacterial genomes is based on the VIMSS operon prediction [106]. The sequence logo painter is implemented in Java 5 (`http://java.sun.com`), as is the graph-based network visualization tool GraphVis that uses an academic license version of the yFiles Java graph library (`http://www.yworks.com`). Transcription factor binding sites are modeled by position weight matrices and are reannotated by means of MoRAine (`https://www.cebitec.uni-bielefeld.de/groups/gi/software/coryneregnet/moraine/`, refer to Section 3.4, page 37). The PWM matching tool PoSSuMsearch [16] (also refer to Section 3.3, page 35) can be downloaded separately from `http://bibiserv.techfak.uni-bielefeld.de/possumsearch`. Protein clusters are obtained by using the FORCE heuristic (`http://gi.cebitec.uni-bielefeld.de/comet/force/`, also refer to Section 3.6, page 48). CoryneRegNet further provides a SOAP-based Web Service server and also queries the GenDB/EMMA services as a client by using the NuSOAP library for PHP (`http://sourceforge.net/projects/nusoap`). The corresponding WSDL files, example scripts etc. are available at `http://www.CoryneCenter.de`. The entire system was developed and runs on servers configured with Solaris 9/Sun OS 5.9. CoryneRegNet itself is tested and runs under Windows XP/Vista, Linux and Solaris OS 5.9 with Internet Explorer 6.0+, Mozilla Firefox 1.5+, or Opera 8.0+. In order to use the GraphVis features, Java 5+ has to be installed and configured.

3.1.2 Ontology-based data structure

Generally, any kind of biological data can be considered as an ontology, which consists of concepts that are linked through relations. Accordingly, the goal was to integrate heterogeneous data related to transcriptional regulation into a database in such a way that they fit into a single ontology-based data structure. In principle, technical and semantic data integration can be performed during data import. If a mechanism exists that ensures the correct semantics of the relations, then different data sources from different levels of biological hierarchy can be integrated into the same database scheme.

An ontology-based data structure consists of concepts that are linked through relations. The integrated data can be regarded as a set of structured and named concepts, whereas the data sources are so-called controlled vocabularies (CVs) [72]. During the data warehousing (import) process, CoryneRegNet creates a dataset concept for each biological entity (genes, proteins, transcription factors, etc.) and a dataset relation for each linkage between two concepts (refer to *from_concept* and *to_concept* in Figure 3.2). All concepts and relations are typed, using concept classes and relation types that are organized internally as

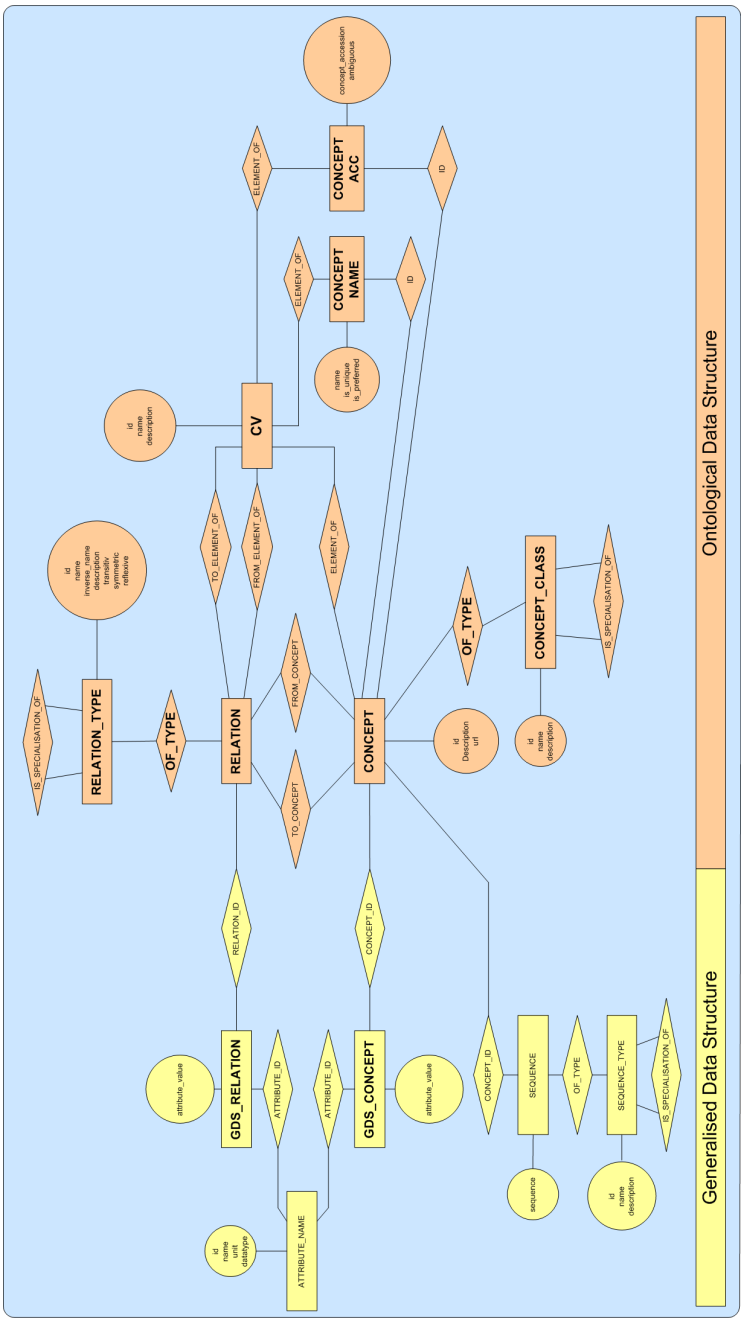


Figure 3.2: Entity Relationship (ER) diagram representing the data structure used for the construction of CoryneRegNet. The ER was implemented in the DBMS MySQL and is divided into two main parts: the generalized data structure (*GDS*) and the ontology-based data structure. Rectangles represent entities, rhombi represent relations between two entities, and circles represent attributes of entities. The entities *Concept* and *Relation*, which are the main components of the ontology-based data structure, are located in the center of the ER diagram. They store all essential data on genes, proteins, etc. as well as every linkage between them. They are typed (*Concept_class* and *Relation_type*) and link to the controlled vocabulary (*CV*) they have been extracted from. Furthermore, they link to their generalized attributes (*GDS_relation* and *GDS_concept*) and to associated sequences (entity *Sequence*). Alternative names and accessions are stored in the tables of the entities *Concept_name* and *Concept_accession*.

specialization trees. Furthermore, the database back-end stores the CV from which the concepts and relations are extracted and the unique accessions and names they have in the source databases. Additionally, attached attributes are saved in a generalized data structure (GDS) to overcome the earlier mentioned attribute handling problem with values from heterogeneous data sources. Figure 3.2 shows the Entity Relationship diagram of CoryneRegNet, which was implemented in MySQL and which is similar to that of the ONDEX (ONtological inDEXing) [71] data structure.

Using an ontological structure mainly impacts the querying procedure, which remains unchanged when extending the data repository by new data integrated from various data sources. For example for CoryneRegNet 2.0, we added three corynebacterial genome annotations to the database back-end. But the process still is very similar to the first release, which contained only a single genome annotation. To include the data, the import manager creates a concept for every biological entity and a relation for every connection between two concepts. In addition, we formulate a new concept class *organism* and a new relation type *belongs_to_organism*. Consequently, each biological concept (gene, protein, etc.) is linked to the concept *organism* by a relation of type *belongs_to_organism*. The result is a database that contains more information but appears unchanged to all querying front-end programs. One example: When using the release 1.0 front-end (designed for a single organism) with the release 2.0 database content (four organisms) to display all genes having 'DtxR' in their names list, the output is a simple list of four genes (one for each corynebacterial species), as the release 1.0 front-end does not distinguish between the species. The important point is that the front-end can access the extended database.

3.1.3 Web Services

Generally, Web Services can be defined as software interfaces that interact via a network connection using XML-based messages. These either contain queries (function calls) or the corresponding results. While the transfer is usually performed using HTTP, the SOAP protocol can be used to describe the message structure. The description of an entire Web Service is conducted by using the Web Service Description Language (WSDL). Any software that is written in a programming language, which offers a SOAP interface can retrieve data directly from that service. Such a program can internally handle all queried data as if the data would be stored in local data structures and memory. Hence, using SOAP-based Web Services, the end-user of an integrated platform even does not recognize that the data is obtained from another system. More general information on Web Services and SOAP can be found in [32, 128].

Most interconnections of biological online databases are still realized by using HTML-links to other web pages or by regular, manual downloads and a subsequent integration of the corresponding data. The introduction of Web Services has opened the way to overcome this workaround and to directly integrate, combine, and visualize appropriate data where it is most expedient. The major advantages of Web Services in automatic biological knowledge processing are:

1. No flat files need to be provided by the distributed platforms, so no extra parsers need to be written.

2. All data is stored in the distributed systems and not copied into a local repository. Storage requirements are decreased but data transfer costs may increase.

3. No updates or adjustments of the federated database scheme are necessary.

4. The repositories do not need to be actively synchronized.

Recently, the access to biomedical Web Services has been published for a growing number of online resources. Some popular examples are: several databases and data analysis services of the EBI, the BRENDA database, KEGG [68], OLS (Ontology Lookup Service) [33], and PathwayExplorer [91].

However, with the existing Web Services no microarray experiments or information on microbial gene regulatory networks can be accessed. Moreover, providing methods for retrieving the necessary data is only the first step towards a successful integrated analysis. The next step, which for the biologist may be more important, is to build a tool on top of this, which allows for convenient data mining and provides suitable visualizations of the integrated data sets. No application is yet known to us that uses Web Services to retrieve data for an integrated analysis of microarray experiments and gene regulatory networks coupled to the genome annotation, and provides a user interface for interactively viewing, browsing, and analyzing the data at the same time.

In the following, we briefly describe the GenDB and the EMMA Web Services. We describe how we use SOAP to integrate the two data sources with CoryneRegNet to provide new analysis methods and how CoryneRegNet profits from the combined power of all three systems in one platform, which is called CoryneCenter.

Since both GenDB and EMMA are implemented in Perl, they provide servers utilizing the SOAP::Lite library (`http://www.soaplite.com`).

Client for GenDB and EMMA

CoryneRegNet benefits on several aspects from the direct connection to GenDB and EMMA:

- For a gene of interest, more accurate and up to date annotation data from GenDB is displayed in the detailed view of a gene (EC numbers for enzymes, Gene Ontology numbers, etc.).

- Using the GenDB Web Service, all target genes of a transcription factor are linked to KEGG pathways and a list of regulated pathways is presented. This allows insights into the general nature of a transcription factor.

- The build-in network visualization Applet GraphVis now features the projection of stimulon data (gene expression levels) extracted from EMMA to the size of the concerned nodes, which represent the genes.

Usually, this kind of interconnections are realized by utilizing HTML-links or by regular, manual downloads of the corresponding data. Using Web Services the user even does not recognize that the data is downloaded from another service. The data is always up to date and it can be post-processed much more easily. As for all distributed platforms, the disadvantage is the dependence on a working internet connection.

Server

The publicly available Web Service server offers several methods to query data from CoryneRegNet. Neither direct access to the MySQL server is necessary, nor any knowledge about the data structure in the back-end. All API information is provided as WSDL file. The WSDL file for the interface of CoryneRegNet is automatically generated on demand by that PHP/SOAP script, which also implements the Web Service functions. After a requirements analysis with biologists, who use CoryneRegNet, we decided to provide the following methods:

- getOrganisms: Compares a given string to all organism names in the database and returns the unique organism identifier for all matches.
- getTfGeneIDs: Returns all identifiers of genes that code for transcription factors, for a given organism ID.
- getGeneID: Genes can have ambiguous IDs in different databases and most of them are additionally stored in CoryneRegNet. This methods returns unique internal gene IDs, given an ambiguous one.
- regulates: For a given gene G, all genes that are regulated by G are returned, including additional information (evidence, regulation type, PubMedID, etc.).
- isRegulatedBy: For a given gene G, all genes that regulate G are returned, including additional information.
- getOperonByGeneID: Returns operon information for a given gene ID.

Now, it is possible to retrieve the most important data from CoryneRegNet directly from any software that is written in a programming language, which offers a SOAP interface. Such a program can internally handle all queried data as if the data would be stored in local data structures and memory.

A detailed documentation (with examples) on how to implement a CoryneRegNet Web Service client is available at the CoryneRegNet and CoryneCenter web sites.

3.2 Visualization

3.2.1 User interface

Web-based user interfaces to biological databases often support the following tasks: (i) browsing by listing or navigating through database entries, (ii) searching by identifying entries based on restrictions on the values of data fields within the database, (iii) visualizing by presenting a visual representation of the data, and (iv) querying by specifying a special search using a query building interface [49]. As well as other gene regulatory databases, such as PRODORIC, CoryneRegNet also emphasizes browsing, searching and visualizing. After login, the entry page of CoryneRegNet shows a statistical summary of the data currently integrated into the database and provides the possibility to browse the organisms (Figure 3.3). Alternatively, the user can start searching the database using

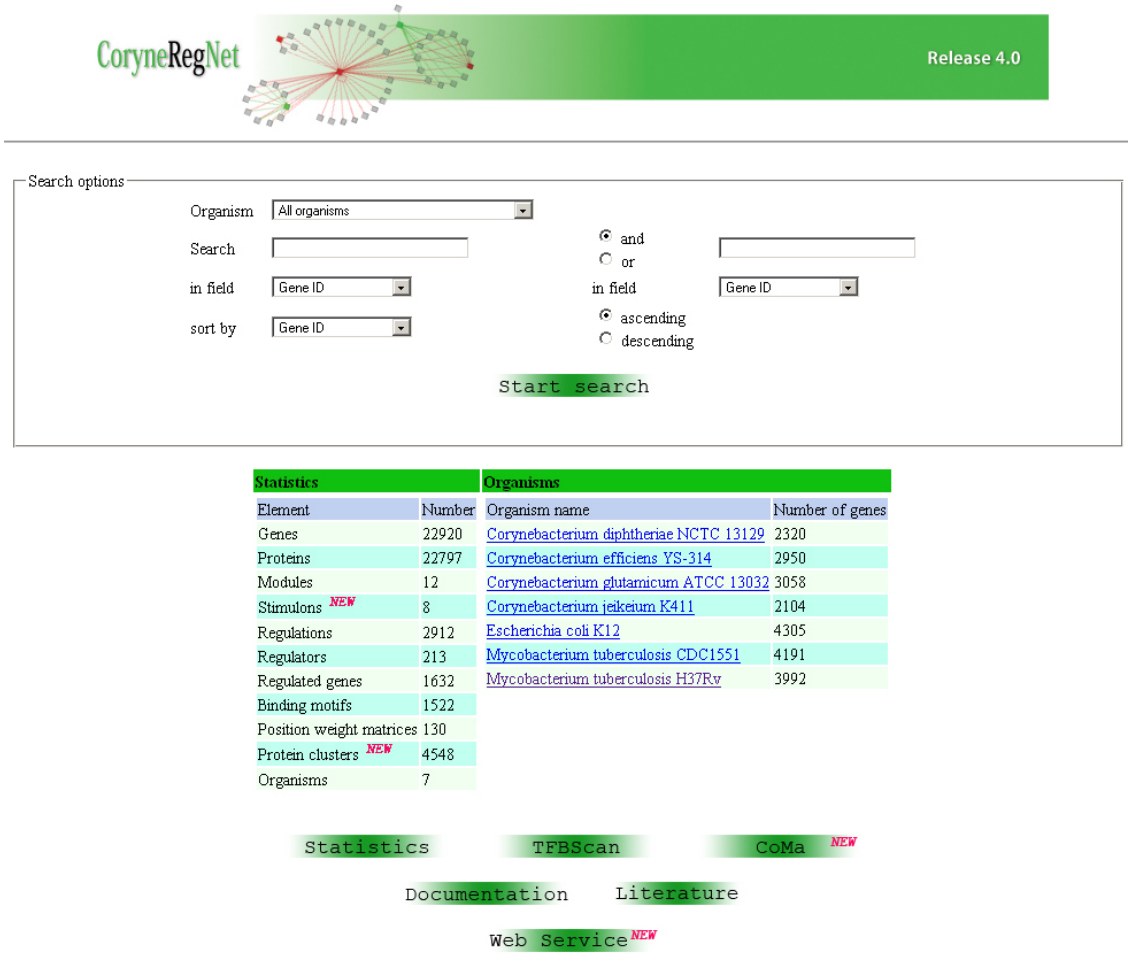


Figure 3.3: Screenshot of the CoryneRegNet main search form.

criteria that were obtained through a requirements analysis with potential users. The criteria are implemented following the typical search mask style of the other gene regulatory databases mentioned in Chapter 2. The search results are presented in a table-based style including gene and protein identifiers and names, the regulator type (if the specific protein is a transcription factor), the functional module the gene belongs to, and the transcriptional regulations the gene is involved in. The user may acquire additional information on specific elements by clicking on them. A typical detailed view of data regarding a transcription factor gene is presented in Figures 3.4, 3.5, and 3.6. It is possible to navigate to other entries of CoryneRegNet, to the genome annotation system GenDB and to the NCBI Entrez Gene database by following the respective links.

Statistics

Statistical analyses, which are performed on-the-fly during browsing the statistics pages of CoryneRegNet, are integrated as visualization into the web interface, thereby reflecting the sum total of the current database content as well as species-specific evaluations:

- Quantities of regulator types and families
- Distribution of the number of transcription factors regulating a gene
- Distribution of the number of co-regulating transcription factors

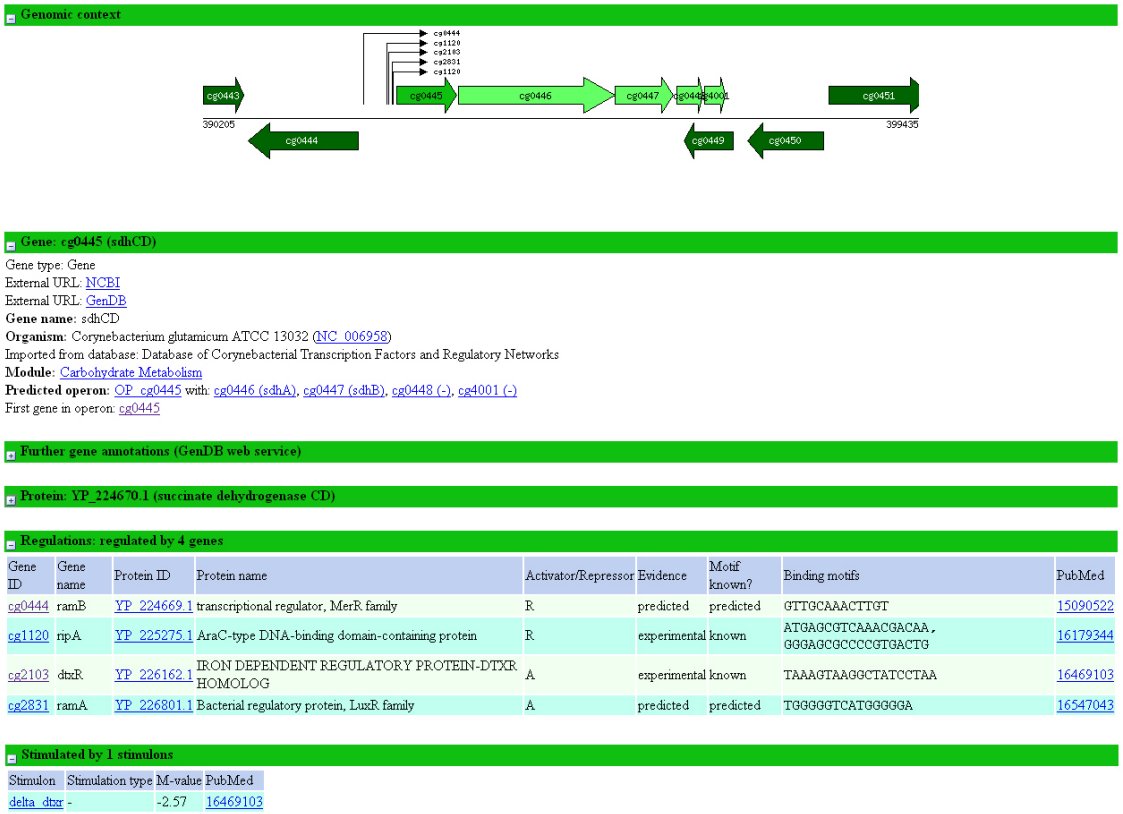


Figure 3.4: Screenshot of the CoryneRegNet details page (part 1) for the gene *sdhCD* (*cg0445*). Shown is the main part including the genome browser and essential information about the gene: operon organization, organism, gene regulations, further gene annotations available from GenDB, and known stimulons that effect the gene expression level of *sdhCD*.

- Number of co-regulators and regulations for each transcription factor
- Distribution of transcription factor binding site distances from the translational start of a gene, and
- Distribution of PWM lengths.

Similar analyses have been performed for *E. coli* and published e.g. in [5,102]. For example Figure 3.7 plots the distribution of the number of TFs vs. the number of genes they control. Babu et al. found similar results for *E. coli* [5]. As another example, Figure 3.8 plots the distances of the TFBMs from the target gene start positions. One can see that repressors tend to dock more closely to the gene start than activators.

Genome browser

The details page of CoryneRegNet also visualizes the selected gene region. When searching a gene, a genome viewer automatically generates a linear display on the top of the details page, showing the position of the selected gene within its genomic surrounding. When the selected gene is part of an operon, all members of this operon appear in their chromosomal arrangement and are specifically colored. TFBMs are shown in front of the selected gene or, if the gene is part of an operon, in front of the first gene of the corresponding transcription unit. An operon is clearly defined as a group of two or more genes that

Regulations: regulates 10 pathways (GenDB web service)

KEGG PathwayID/Link	KEGG Pathway name	Regulated genes and EC numbers
271	Methionine metabolism	(1): cg1996 (2.1.1.37)
230	Purine metabolism	(2): cg3114 (2.7.7.4), cg3115 (2.7.7.4)
920	Sulfur metabolism	(5): cg3114 (2.7.7.4), cg3119 (1.8.1.2), cg3115 (2.7.7.4), cg3118 (1.8.1.2), cg3116 (1.8.4.8)
720	Reductive carboxylate cycle (CO2 fixation)	(1): cg0446 (1.3.99.1)
450	Selenoamino acid metabolism	(4): cg3114 (2.7.7.4), cg3119 (1.8.1.2), cg3115 (2.7.7.4), cg3118 (1.8.1.2)
860	Porphyrin and chlorophyll metabolism	(1): cg2445 (1.14.99.3)
650	Butanoate metabolism	(1): cg0446 (1.3.99.1)
632	Benzoate degradation via CoA ligation	(1): cg0446 (1.3.99.1)
190	Oxidative phosphorylation	(1): cg0446 (1.3.99.1)
20	Citrate cycle (TCA cycle)	(1): cg0446 (1.3.99.1)

Stimulated by 1 stimulons

Attributes, position weight matrix and sequence logo

Attribute name	Attribute value
Gene start	1992314
Gene end/stop	1993000
Gene located at complementary strand?	false
Codon start	1

Nucleotide content

	relative	absolute
A	22.9 %	157
T	21 %	144
C	28.5 %	196
G	27.7 %	190

Attribute name	Attribute value
Regulator Type	DtxR
TF is an autoregulator?	false

Position Weight Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	3	8	26	5	0	5	10	28	5	0	3	4	22	20	4	2	1	23	19
T	26	22	2	3	1	20	18	4	1	1	7	28	6	2	2	6	29	9	7
C	1	2	3	2	0	6	1	0	3	11	20	1	5	2	26	25	1	0	3
G	3	1	2	23	32	2	4	1	24	21	3	0	0	9	1	0	2	1	4

Sequence logo

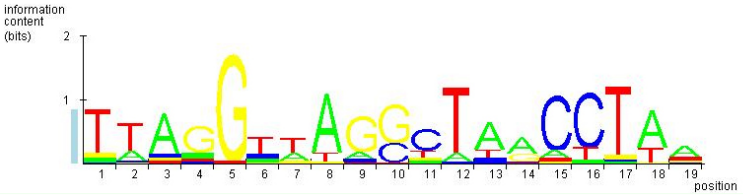


Figure 3.5: Screenshot of the CoryneRegNet details page (part 2) for the gene *dtxR* (*cg2103*), which encodes a repressor of *C. glutamicum* responsible for the regulation of the iron metabolism. Shown are that KEGG pathways that are predicted to be regulated (obtained by using the GenDB Web Service). Furthermore, some essential information about the gene itself is given followed by the PWM of the repressor and the sequence logo.

are transcribed as a polycistronic unit. The respective operon information of *E. coli* has been adopted from RegulonDB, whereas the VIMSS operon predictions [106] have been used for corynebacteria, but manually curated afterwards. The graphical design of the genome viewer uses an image map, allowing direct access to other gene details pages of CoryneRegNet by clicking on them (refer to the top of Figure 3.4).

Sequence logos

Sequence logos are a graphical method for displaying patterns in a set of aligned sequences [118] and, accordingly, provide suitable tools for the characterization of DNA-binding sequence motifs of transcriptional regulators [31, 112]. Since sequence logos display both significant residues and subtle sequence patterns, one can determine not only the consensus sequence for DNA binding of a transcriptional regulator but also the relative frequency of bases and the information content (measured in bits) at every position in a nucleotide sequence. To create sequence logos from the TFBMs gathered in CoryneReg-

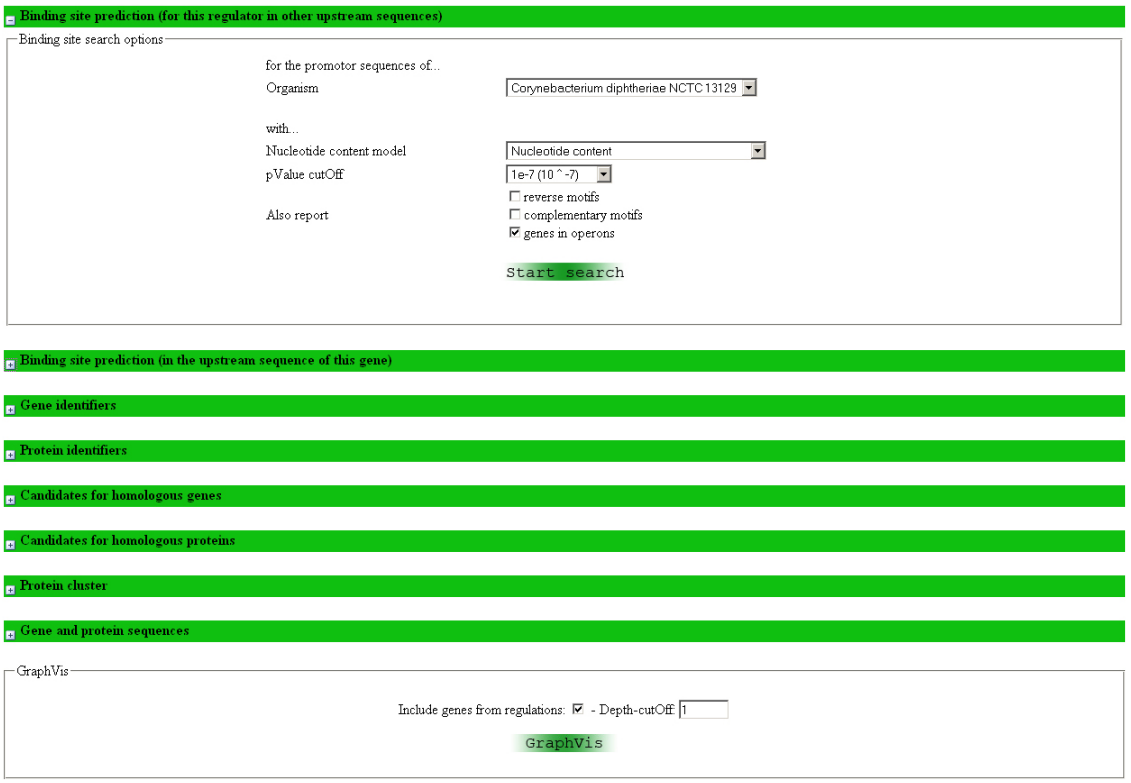


Figure 3.6: Screenshot of the CoryneRegNet details page (part 3) for the gene *dtxR* (*cg2103*). Shown is the binding site prediction (TFBScan) start form (top) and the start form for the GraphVis applet that visualizes the gene regulatory network of *dtxR* up to a certain depth threshold (bottom). Besides, one can optionally view further known gene/protein identifiers, candidates for homologous genes/proteins, the protein cluster *dtxR* is assigned to, and the nucleotide/amino acid sequences.

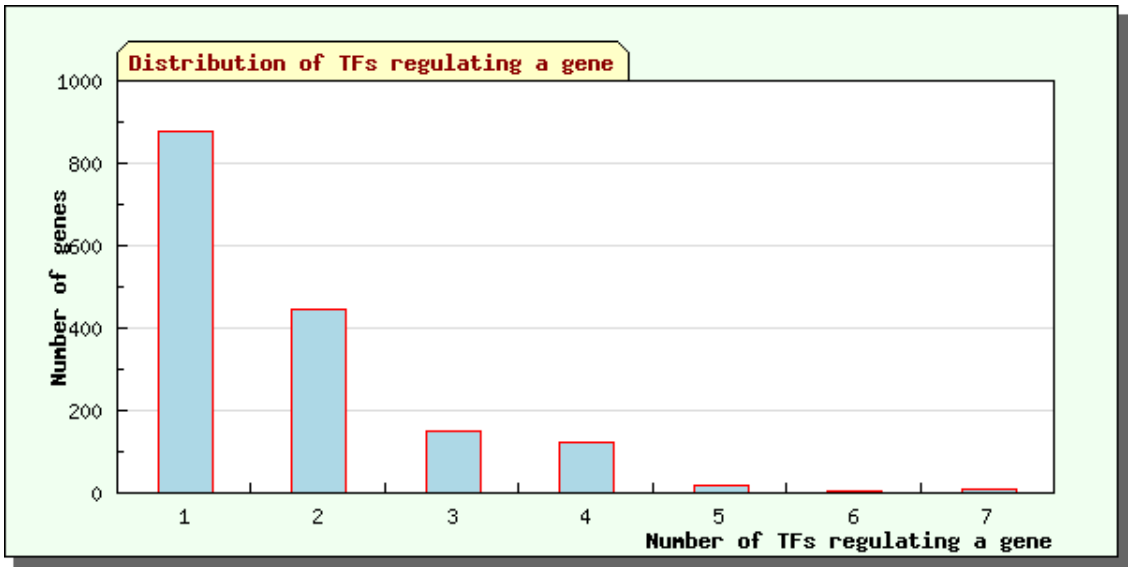


Figure 3.7: This figure plots the distribution of the number of transcription factors vs. the number of genes they control.

Net, the pre-calculated position frequency matrices (PFMs) are used for the computation. The PFMs are automatically updated during data warehousing as new TFBMs are integrated into the database. The logo geometry is stored in XML files generated by means

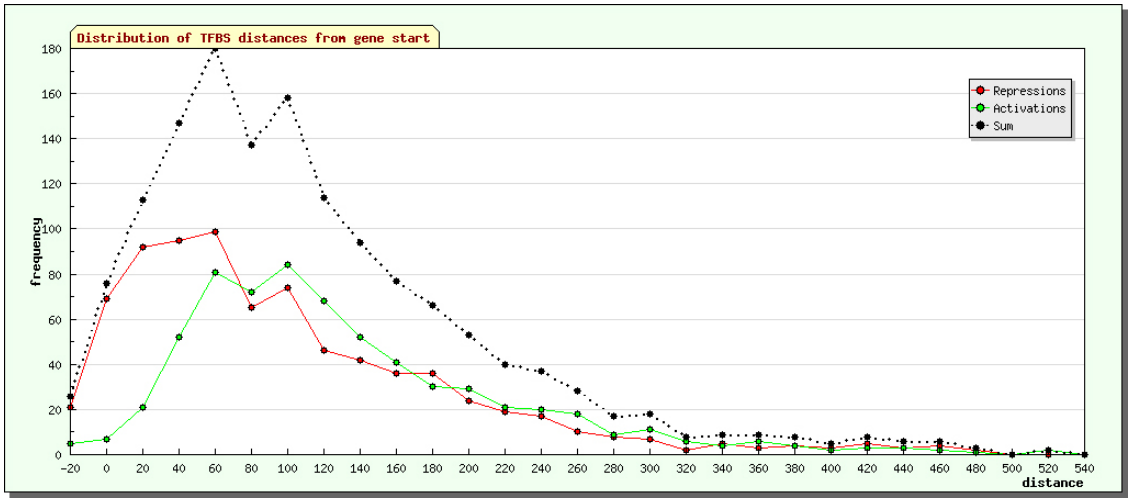


Figure 3.8: This figure plots the distances of the TFBMs from the target gene start positions. Note that some TFs can dock within the coding region of the gene they control (negative values at the x-axis). The red points represent the number of repressions for the respective distances, the green points for the activations. The black curve is the sum of both.

of JDOM (`http://www.jdom.org`), also allowing the graphical design of other logo types, such as hidden Markov model (HMM) logos [119]. In principle, the height of each character representing the DNA binding motif is made proportional to its frequency, and the characters are then stacked on top of each other for each position in the aligned nucleotide sequences [118]. Here, we use colored bars for underrepresented logo characters at each position of the motif to improve the visualization of the sequence logos. The height of an entire stack is proportional to the information content I_i of the motif at position i, and the letters are sorted so the most common one is on top of the stack. I_i is defined as the difference between the maximal nucleotide entropy E_{max} and the observed entropy E_{obs} in a certain column i of a PFM. If we assume a uniform distribution of the N nucleotides, the information content at position i and hence the stack height in the logo is as follows:

$$I_i = E_{max} - E_{obs} = 2 - (- \sum_{\sigma \in \{A,T,C,G\}} f_{\sigma i} \cdot \log_2 f_{\sigma i}) \quad \text{[bits]},$$

where $f_{\sigma i}$ is the frequency of nucleotide σ at position i. Note that for $N = 4$ nucleotides $E_{max} = \log_2 N = \log_2 4 = 2$ [bits]. In addition, the mean information content of the DNA binding motif is calculated and indicated graphically (refer to the examples in Figure 3.5 on page 30, and Figure 4.7 on page 77).

3.2.2 GraphVis

The user can visualize a transcriptional regulatory network at every navigation point using a result table or a detailed frame as starting point. The user has to define a graph depth cut-off and whether genes from hierarchical regulations should be included into the graph (refer to the *GraphVis* button in Figure 3.6). Graph construction starts with the selected set of genes, propagates through the regulatory network and adds more genes into the graph until the depth cut-off has been reached. The network visualization toolkit is a Java Applet. Due to security restrictions, it can not query the database server directly. Instead, it sends

its requests to a PHP script that redirects the query to the back-end and subsequently sends the results back to the applet. The user obtains the same details on genes, proteins and regulatory interactions as in the browser-based view of CoryneRegNet. The main advantage is the graphical overview of the reconstructed regulatory network, where nodes in the graph represent genes and edges represent regulatory relationships. The user can zoom into the graph, layout the graph by using different styles, remove selected elements from the graph or retrieve detailed information on selected genes. The user can extend the displayed graph by using an import wizard that provides a similar search mask as for the table-based web front-end. Furthermore, it is possible to visualize predicted transcriptional regulatory networks, and to compare them with evidenced graphs. Another practical aspect is that GraphVis also provides the interspecies comparison of regulatory networks.

In the following, we first introduce the comparative graph layouters that help with the interspecies network comparison, and afterwards we briefly describe the projection of experimental gene expression (i.e. microarray) results to visualized graphs.

Homology-based graph layouting

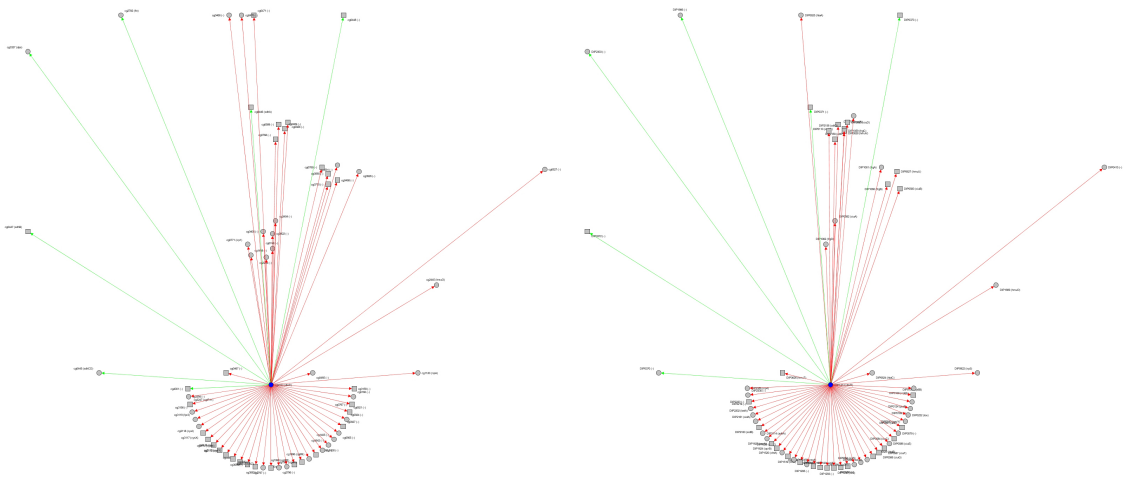


Figure 3.9: Visualization of the regulatory networks of DtxR of *C. glutamicum* (left side) and *C. diphtheriae* (right side) by using a force-based comparative graph layout.

In order to compare depth-1 gene regulatory networks, special graph layouters are necessary. Assume two sets of target genes $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ of two transcription factors t_A and t_B. Furthermore, we have given sequence-based similarities $s : A \times B \to \mathbb{R}$ between all pairs of genes. Consider a pair a, b as a homology $a \sim b$, if $s(a, b)$ exceeds a certain threshold. Let $H_A := \{a \in A \mid \exists b \in B, a \sim b\}$ be the set of those genes regulated by t_A that have at least one homology partner in B, and $H_B := \{b \in B \mid \exists a \in A, a \sim b\}$ respectively. We denote the set of those genes in A without any homology partners in B with $N_A := A \setminus H_A$, and $N_B := B \setminus H_B$ respectively.

The most simple layout style is illustrated in the application cases in Figure 4.6 (page 76), and Figure 4.8 (page 78). The genes of N_A and N_B are organized as semicircles on the left and the right side. Those genes from H_A and H_B are arranged clique-wise in the middle and connected by an undirected (black colored) edge that illustrates their potential homology.

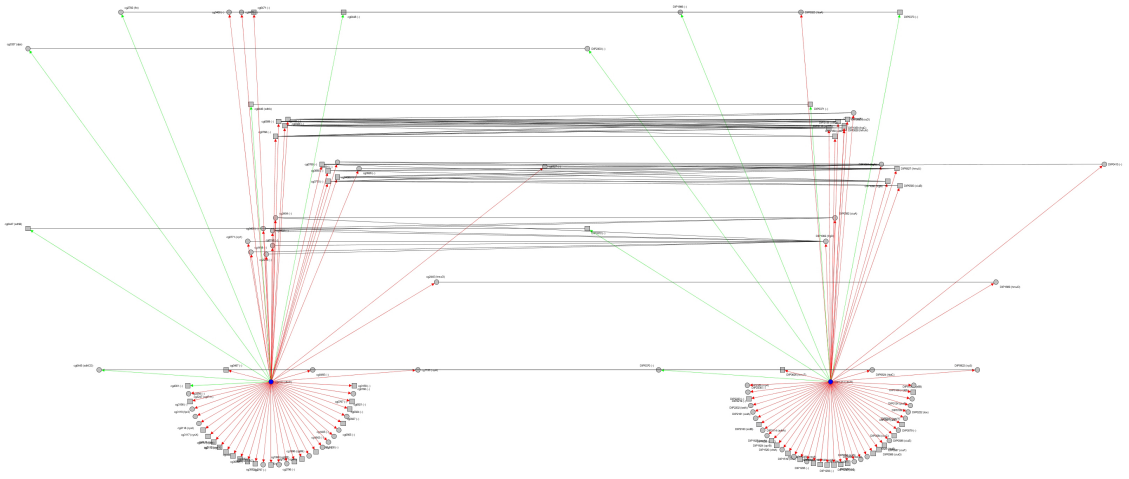


Figure 3.10: The same visualization as in Figure 3.9 but with homology edges (black).

A better visualization avoids homology edges. Such a graph layout is shown in Figures 3.9 and 3.10. The genes of N_A and N_B are organized as semicircles again but now below t_A and t_B (the blue nodes in the middle). Note that in Figure 3.10 all homology edges are horizontal. Just by the position of a node, the user can easily find its homology partners, even without the corresponding (black) edges (refer to Figure 3.9).

The main idea is, to position the cliques of genes of H_A to positions similar to those of their partners in H_B, relative to the root nodes t_A and t_B. In order to provide good graph layouts within reasonable response times, we implemented a heuristical solution that is based on physical intuition, and initially was introduced by Fruchterman and Reingold, and subsequently extended e.g. for BioLayout [46, 53].

The main idea of these layout algorithms is to arrange all nodes on a 2-dimensional plane to fit aesthetic criteria: e.g. an even node distribution in a frame and inherent symmetry reflection. The graphs' nodes are interpreted as magnets (or electrical charges of the same kind) and edges are replaced by rubber bands to form a physical system. The nodes are initially placed randomly, for example, and then left to the forces of the system, so that the magnetical repulsion and the bands' attraction forces on the nodes move the system to a minimal energy state. While a physical system provides the motivation for these algorithms, in the actual implementation the nodes need not move according to exact physical laws.

Expression level visualization

As already mentioned in Section 3.1.3, CoryneRegNet supports data exchange by utilizing Web Services. Gene expression data that is stored and managed in the EMMA system can be queried and visualized with GraphVis. Aside microarray data from EMMA, the user also has the following possibilities to provide data:

- TAB-delimited flat file
- MS Excel file.
- Usage of stimulon data from the CoryneRegNet database.

The GraphVis Applet first extracts the M-values (measure of differential gene expression) for all the genes of a given dataset. Subsequently, the node size s of all corresponding genes in a given visualized regulatory network is changed relative to the M-value m (with $-\infty < m < \infty$). This is done by setting $s = s_{old} \cdot (\mid m \mid +1)$, where s_{old} is the node size before the variation. Furthermore, the nodes of those genes with $m < 0$ ($m > 0$) are modified to have a red (green) dotted border. Figure 3.15 on page 46 and Figure 4.9 on page 79 illustrate this visualization style. It helps to find putative inconsistencies or even contradictions in visualized gene regulatory networks (also refer to the application case in Section 4.1.4 on page 77).

3.3 Binding site prediction

The typical user of CoryneRegNet looks for answers to many different questions, most of which cannot be anticipated. As new questions arise, the bioinformatics community is usually quick to develop algorithms and software packages to attack these problems. For CoryneRegNet, there is no need to re-invent the wheel; instead, our focus is to integrate the best available special-purpose tool for a particular task and merge the obtained results into the existing knowledge base. Here we describe the integration of the motif matching software PoSSuMsearch [16] from `http://bibiserv.techfak.uni-bielefeld.de/possumsearch/` that provides a fast and statistically sound method to detect transcription factor binding sites in a collection of DNA sequences. CoryneRegNet provides an easy-to-use interface to PoSSuMsearch, for example by the $TFBScan$ button on its title page.

In what follows, let $\Sigma := \{A, T, C, G\}$ be the DNA alphabet. There are many models to describe the DNA motif a particular transcription factor binds to. By far the most widely used one is a position frequency matrix (PFM). A PFM can be converted to a position weight matrix (PWM): For a motif of length m, a PWM is a $4 \times m$ matrix $S = (S_{c,i})_{c \in \Sigma, 1 \leq i \leq m}$ of real numbers (weights or scores). A PWM allows to assign a score $s(w)$ to any length-m DNA sequence window $w = (w_1, ..., w_m)$ by setting

$$s(w) = \sum_{i=1}^{m} S_{w_i}, i.$$

We say that the PWM S matches w if $s(w) \geq t$ for a suitably defined score threshold t. The idea is that the matches are good candidates for real TFBMs if we properly choose the scores S_{ij} (generally as log-odds scores between nucleotide distributions of true binding sites on the one hand and a background distribution on the other hand) and the threshold t (ideally based on statistical considerations of both type-I and type-II error; see e.g. [107]).

A typical use case would look as follows: Assume that it is known that a certain transcription factor regulates certain genes and that the binding sequences upstream of these genes are also known. From these sequences, we can build a PWM model and use it to look for further matches upstream of potentially regulated genes (e.g., those found to be co-differentially expressed in microarray analyses). We might also be interested in doing a genome-wide search for the motif, although without any contextual information, motif occurrences are generally not meaningful. The computational problem remains the same:

PWM Matching Problem For a given PWM S of length m, threshold t, and a DNA sequence $g = (g_1, ..., g_n)$, $n \geq m$, identify all positions i with $s(g_i, \ldots, g_{i+m-1}) \geq t$ and report their scores.

Two algorithms that solve this problem are given below, in order of increasing complexity and decreasing running time (on typical large-scale DNA data):

Sliding window scoring and variations For each starting position $i \in \{1, \ldots, n-m+1\}$, in increasing order, compute the window score $s_i := s(g_i, \ldots, g_{i+m-1})$ by summing m values and report (i, s_i) if $s_i \geq t$. The time complexity is obviously $O(mn)$. Time can be saved by stopping the evaluation of s_i before all m positions have been scored as soon as it becomes clear that the threshold t cannot be reached. This is referred to as lookahead scoring, which is most beneficial if the evaluation frequently stops after only one or two positions for each window. However, the first positions of the PWM are rarely the most discriminative ones, and more time can be saved by evaluating to positions not from left to right, but by a suitably determined permutation of the positions; a heuristic rule may be found in [132].

Enhanced suffix array search (ESAsearch) Instead of permuting the positions of the PWM, we may choose a different evaluation order of the text positions such that windows that share many prefix characters are evaluated together: Let $G_i := (g_i, ..., g_n)$ be the i-th suffix of g. Now order the suffixes lexicographically, i.e., find the permutation p of $\{1, ..., n\}$, called the suffix array of g, such that $G_{p(1)} < G_{p(2)} < \ldots < G_{p(n)}$. Additional tables, *lcp* and *skp*, that contain the longest common prefix (*lcp*) lengths between lexicographically adjacent suffixes, and the array position of the next smaller *lcp* length, respectively, are also created. Together with p, they form the enhanced suffix array. The enhanced suffix array needs to be pre-computed only once for the whole sequence content of the database and enables subsequent fast searches: Scoring sequence windows in lexicographic order with *lcp*-information allows to re-use partial prefix scores without recomputing them. For example, assume that sequence windows $w_1 = ACCAG$ and $w_2 = ACCAT$ are adjacent; their *lcp* length is 4. Knowing the partial score of the length-4 prefix of w_1, we only need to add the $T - score$ at position 5 to obtain the score for w_2. Using the *skp*-table, large parts of the text that can never reach the threshold because low prefix scores can be skipped in constant time.

PoSSuMsearch implements (non-permuted) lookahead scoring and ESAsearch; the latter one being generally fastest on long DNA sequences because of the ability to skip large parts of the sequence. CoryneRegNet contains 130 PWMs in the form of nucleotide frequency count matrices and 19 MB of upstream sequence data (1.8 MB alone for C. glutamicum). To our knowledge, PoSSuMsearch is the only available software package that is fast enough to provide interactive response times for large-scale PWM searches and at the same time integrates exact statistics: The score threshold t for matching is automatically computed based on the tolerable frequency of hits in random sequences (p-value) by an efficient and exact lazy-evaluation method [17].

Integration in CoryneRegNet PoSSuMsearch interacts with CoryneRegNet as follows: During data warehousing, the enhanced suffix array for each of the four corynebacteria

is created to allow the use of ESAsearch instead of a slower window-sliding algorithm. Before a search is started, CoryneRegNet computes the background distribution (nucleotide content frequencies) of the search space and the log-odds PWM on-the-fly and passes them to PoSSuMsearch via temporary files. PoSSuMsearch is executed using a system call from the PHP front-end. It temporarily creates tab-delimited flat files storing the matching results which are read by the font-end and deleted afterwards.

An application case and an evaluation is given in Section 4.1.2 on page 72. The user interface to PoSSuMsearch is available as *TFBScan* feature at the CoryneRegNet web site (refer to Figure 3.6 on page 31). The user has to choose a p-value threshold, a background model, and a target organism.

3.4 MoRAine - Binding site reannotation

Obviously, an important prerequisite for the construction of PWMs is an accurate annotation of TFBMs. The determination of TFBMs in wet lab experiments is time-consuming and error-prone. Nowadays, the position within the double-stranded DNA sequence to which a TF binds is determined by electrophoretic mobility shift assays (EMSA) [58], DNAse footprinting [48], ChIP-chip [122], or mutations of putative TFBMs and subsequent expression studies. All of these methods lack a precise identification that is accurate to one base pair (bp). Generally, TFs bind the double-stranded DNA and it is a matter of interpretation which strand of the DNA sequence is annotated (for example, the binding sequence $AGGCAT$ on the forward strand is equivalent to the sequence $ATGCCT$ on the reverse strand). Conceptually, this poses no problem, since given either motif, its reverse complement is easily computed. However, a practical problem occurs when a motif from either strand-based on approximate knowledge of its position is entered in a database and subsequently used blindly for PWM construction. This does happen in practice, especially for regulatory databases that integrate information from other sources, e.g., RegulonDB. Here all TFBMs are given $5' \rightarrow 3'$ (forward) relative to the target gene.

Since the stored motif is essentially chosen from a random strand, subsequently constructed PWMs may show a poor information content (e.g., a mixture of $AGGCAT$ and $ATGCCT$ instead of either motif) that consequently leads to bad binding motif predictions from the PWM.

In this section, we introduce MoRAine, an algorithm and software that assists with automatic TFBM reannotation. All motifs with experimental evidence underlying a PWM are evaluated with reference to their similarity to all other motifs. The goal is to reannotate the TFBMs by switching the strand and possibly shifting them a few positions in order to maximize the information content of the resulting adjusted PWM.

First, we give some definitions. Then we show that both methods implemented in MoRAine significantly increase the matrix quality by means of two examples calculated with the MoRAine web server version. In one example, we adjust the TFBMs of the regulator NarL of *E. coli*. We show that the corresponding sequence logo looks very similar to the manually reannotated, which is stored in the PRODORIC database. We discuss the same for the regulator MalT. Subsequently, we show that the PWMs resulting from the adjusted TFBMs significantly improve the prediction performance. MoRAine-adjusted PWMs increase the accuracy and decrease both the false negative and the false

positive rates. We finally introduce the MoRAine web server, an easy-to-use alternative for the computation of sequence logos, since it directly integrates PWM quality improvement. The stand-alone version of MoRAine can easily be included into a database back-end (i) as quality assurance and (ii) to additionally provide adjusted PWMs for subsequent TFBM predictions. Hence, we integrated MoRAine into CoryneRegNet.

Definitions

Again, in what follows, let $\Sigma := \{A,T,C,G\}$ be the DNA alphabet.

As already mentioned, the most widely used model to describe a set of TFBMs for a given TF is a position frequency matrix (PFM), defined as follows: Given a set of n TFBMs of length m over the alphabet Σ, a *position frequency matrix* $F = (f_{\sigma j})$ for a set of n TFBMs of length m is a $|\Sigma| \times m$ matrix, where $f_{\sigma j}$ is the frequency of symbol σ at position j.

Information content based sequence logos can be used to judge the PFM quality [31]. The *information content* I_j for column j of a PFM F is defined as

$$I_j := \log_2 |\Sigma| + \sum_{\sigma \in \Sigma} f_{\sigma j} \cdot \log_2 f_{\sigma j} \quad \text{[bits]}.$$

I_j reaches its maximum if and only if all symbols at position j agree; for $|\Sigma| = 4$, the maximal value is 2 bits (also refer to Section 3.2.1, page 30). The *mean information content* $I(F)$ for a whole frequency matrix F is

$$I(F) := \frac{1}{m} \sum_{j=1}^{m} I_j.$$

We use the mean information content as quality measure and denote it shortly with I if the matrix F is fixed.

3.4.1 Methods

Information content maximization

We start with a set of DNA sequences that extend l bp to the left and r bp to the right of the annotated TFBMs and set $m^+ := m + l + r$ to the length of the given sequences. Given a set of n sequences of length m^+, we first calculate the set M of every possible motif of length $m = m^+ - l - r$ derived by the operations *shift* and *switch* applied to every sequence. The operation *shift* provides every substring of length m for a given motif of length m^+, and the operation *switch* its reverse complements. This leads to a set S_i of $|S_i| = |M| = 2 \cdot (l + r + 1)$ motifs of length m for each input sequence i, with $i = 1, \ldots, n$.

The goal of this work is to find a set of motifs C that contains exactly one motif from each S_i and maximizes the information content of the corresponding frequency matrix F_C. We propose two heuristic algorithms (*cg* and *km*) based on clustering to find such a motif set C. Both utilize one of two similarity functions ($simC$ or $simS$).

As introduced above, the goal is to find a set of motifs C that contains exactly one motif from each S_i and maximizes the information content of the corresponding PFM F_C. Here, we describe two heuristic clustering algorithms to find such a motif set C.

Similarity Measures

For our clustering algorithm, we need a similarity function sim that measures the similarity between one motif and an existing cluster and thus helps to evaluate to which cluster of TFBMs a new TFBM is assigned. We use two different functions.

Motif-cluster similarity To measure the similarity between a single TFBM s and an existing non-empty cluster C', we calculate I for the frequency matrix constructed from all TFBMs of C', including s itself. We denote this function as $simC$.

Motif-seed similarity Following another strategy, each cluster is represented by a seed motif. Here we calculate I for the frequency matrix built from only the seed motif and the new TFBM. We denote this function as $simS$; it is faster to evaluate, but less accurate than $simC$.

These definitions apply only if the cluster C' to which a new motif s from a set S_i is to be assigned does not yet contain another motif from S_i. Otherwise, the similarity is set to $-\infty$; this ensures that each cluster contains only one motif from every set S_i.

Clustering strategies

The goal is to partition the set of motifs into $|M| = 2 \cdot (l+r+1)$ clusters, where each cluster contains exactly n motifs, one of each S_i ($i = 1, \ldots, n$) and thus is a putative solution. We describe two clustering strategies.

Variant of k-means with random seeds In this particular application, the number $|M|$ of clusters is known; so we use a variation of the k-means algorithm [57]. In the end, we pick the cluster with the highest mean information content I.

We start with a random set of $|M|$ (out of $n|M|$) motifs (the *seeds*) that form the initial clusters (see below for details on how to choose the initial seeds). Then, the following procedure is iterated until convergence: Each motif, in arbitrary but fixed order, is assigned to the cluster that maximizes the similarity ($simC$ or $simS$) value. This results in $|M|$ clusters, each consisting of n motifs. A new seed sequence is chosen for each cluster as the sequence that best represents the cluster. This continues until no more changes occur for the seed sequence set; see Algorithm 1 for details. This strategy can be repeated for different initial seeds and addition orders.

Cluster growing Since each motif of each S_i must be in a different cluster, each S_i is used in turn as a set of initial seeds. Subsequently, the other motifs are added to their most similar cluster, similarly to the first iteration of the km algorithm, but this procedure is not iterated. Finally, the best solution obtained from the n different starting configurations is reported (see Algorithm 2 for details).

Note that both clustering strategies (the km and cg) can be combined with both similarity functions ($simC$ and $simS$). The implications for running time and quality are discussed below.

Algorithm 1 Clustering with *km*

Input: *sim*, all sets S_i, with $i = 1, \ldots, n$, $|S_i| = |M|$
Output: Set of motifs C with maximal information content I

$oldseeds \leftarrow \{\}$
$seeds \leftarrow \{|M|$ arbitrary elements of $\bigcup_{i=1}^{n} S_i\}$
while $seeds \neq oldseeds$ **do**
    initialize clusters C_j, with $j = 1, \ldots, |M|$, with one seed per cluster
    $oldseeds \leftarrow seeds$
    for $i \leftarrow 1$ to n **do**
        for all motifs m in S_i **do**
            assign m to cluster C_j with maximal $sim(m, C_j)$ over $j = 1, \ldots, |M|$
    $seeds = \{\}$
    for all clusters C_j **do**
        find motif $m \in C_j$ with maximal $\sum_{m' \in C_j} sim_S(m, m')$
        add m to *seeds*
$C \leftarrow C_j$, with maximal $I(F_{C_j})$ over $j = 1, \ldots, |M|$
return $(C, I(F_C))$

Algorithm 2 Clustering with *cg*

Input: *sim*, all S_i, with $i = 1, \ldots, n$
Output: Set of motifs C with maximal information content I

$I_{best} \leftarrow 0$, $C_{best} \leftarrow \{\}$
for $i = 1$ to n **do**
    $seeds \leftarrow S_i$
    initialize clusters C_j, $j = 1, \ldots, |M|$, with one seed per cluster
    for each $k \neq i$ **do**
        for all motifs m in S_k **do**
            assign m to C_j with maximal $sim(m, C_j)$ over $j = 1, \ldots, |M|$
    $C \leftarrow C_j$, with maximal $I(F_{C_j})$ over $j = 1, \ldots, |M|$
    if $I(F_C) \geq I_{best}$ **then**
        $I_{best} \leftarrow I$, $C_{best} \leftarrow C$
return (C_{best}, I_{best})

3.4.2 Results

Information content improvement

We implemented MoRAine in JAVA. It is open source and can be downloaded at `https://www.cebitec.uni-bielefeld.de/groups/gi/software/coryneregnet/moraine/`.

Furthermore, MoRAine can be used as a web application. The user can copy and paste lists of TFBMs in FASTA format. Using such a list as input, the MoRAine web server calculates (i) the adjusted TFBMs and (ii) the corresponding sequence logos using the Berkley web logo library [31]. The adjusted TFBMs can be downloaded in FASTA format and used to build adjusted PFMs.

Figure 3.11 illustrates two example outputs of the MoRAine web server for the transcriptional regulators NarL and MalT of *E. coli*. One can see that the average information content is significantly improved. For NarL, we allowed to shift the motifs by at most one position to the left or to the right ($l = r = 1$). Therefore, we added one base pair to the left and to the right of the annotated TFBMs as flanking sequences. We provide both examples as application cases at the MoRAine web site.

The manually curated database of procaryotic transcriptional regulations PRODORIC

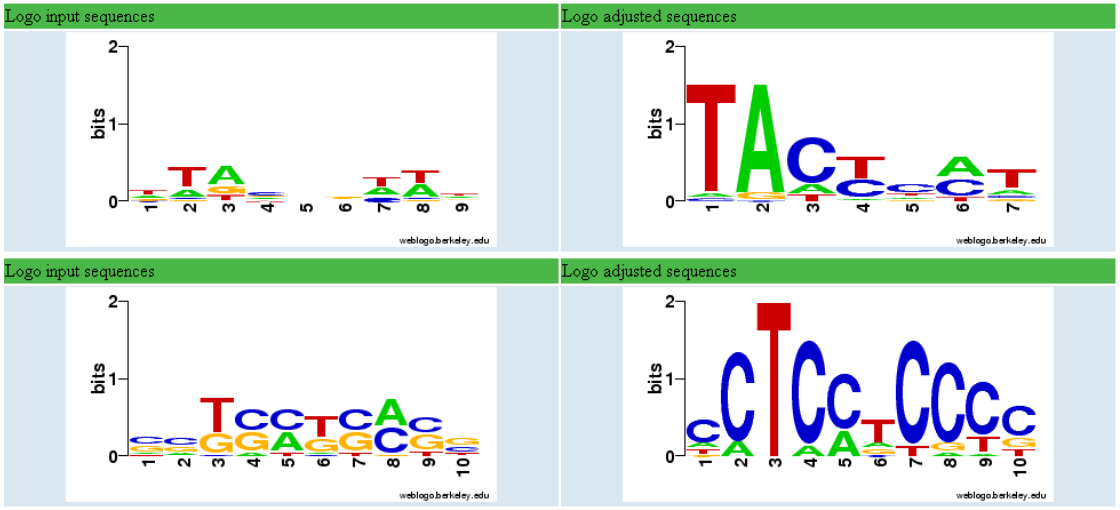


Figure 3.11: A comparison of the sequence logos constructed from the original database TFBMs (left side) and the adjusted TFBMs by using MoRAine (right side). The corresponding transcription factors are NarL (top) and MalT (bottom), both from *E. coli*. The 75 TFBMs for NarL and the 20 TFBMs for MalT have been extracted from RegulonDB. For NarL, we allowed to shift the motifs by at most one position to the left or to the right. The figure was taken as a screenshot from the MoRAine website.

Table 3.1: This table summarizes the information content improvement and the running times of MoRAine for different l- and r-values and all four search method/similarity function combinations.

	Difference (%)				Time (s)			
$l = r$	$cg/simC$	$cg/simS$	$km/simC$	$km/simS$	$cg/simC$	$cg/simS$	$km/simC$	$km/simS$
0	26.1	**27.0**	26.5	26.8	**0.6**	0.7	1.2	1.1
1	50.9	**54.4**	50.1	52.3	**0.7**	2.3	7.2	4.0
2	57.5	**63.6**	57.6	62.4	**0.8**	4.2	45.9	8.3
3	60.0	**69.5**	64.6	64.7	**1.0**	8.4	128.0	12.8
4	65.3	**70.1**	65.0	69.3	**1.1**	11.9	198.3	19.5
5	66.3	73.0	68.8	**73.3**	**1.3**	16.8	298.3	30.5
6	66.6	73.1	74.3	**74.9**	**1.8**	23.9	427.0	34.4
7	68.0	**78.7**	73.5	78.4	**2.0**	30.1	505.4	42.6

also provides TFBMs and sequence logos for NarL at `http://www.prodoric.de/matrix.php?matrix_acc=MX000003` and MalT at `http://www.prodoric.de/matrix.php?matrix_acc=MX000139`. As in most databases, also in RegulonDB, each TFBM is annotated in $5' \rightarrow 3'$ direction relative to the regulated target gene. Similar to our automated approach, the database annotators of PRODORIC improved the TFBMs annotations manually. They utilized the same operations to the TFBMs as MoRAine, namely *shift* and *switch*. Additionally, they removed or shortened TFBMs if necessary and beneficial. In the case of NarL both adjusted sequence logos look the same. In the case of MalT, the PRODORIC annotators choose (i) to shorten the motifs from 10 bps to 6 bps and (ii) to use the reverse complement TFBM sequences. We can reproduce this annotation by using the 10 bps TFBMs of MalT as input for MoRAine and set the user defined parameters $l = r = 2$.

An impression of how the running time scales with the number of input sequences is

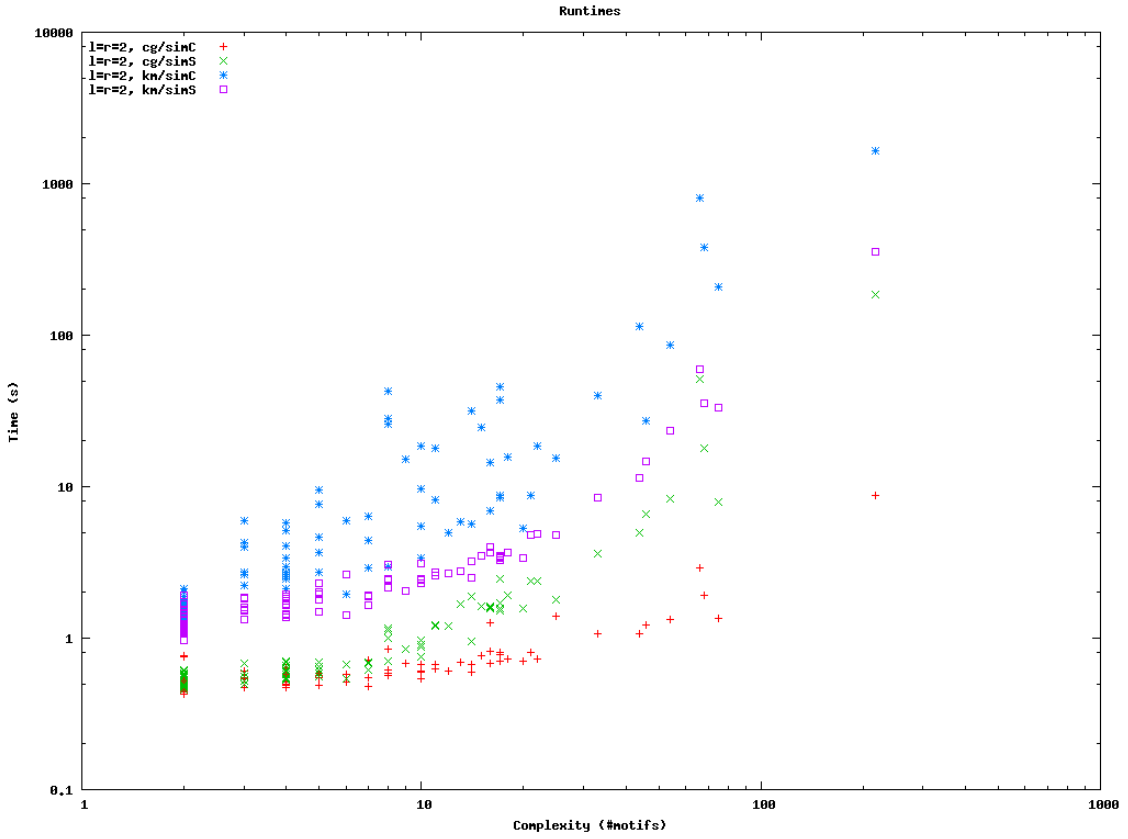


Figure 3.12: This plot illustrates the running times of MoRAine for different numbers of input TFBMs for $l = r = 2$. Note that both axes are log-scaled.

illustrated in Figure 3.12 (for $l = r = 2$). The fastest combination of search algorithm and similarity function is $(cg/simC)$. In order to decide which combination of search strategy and similarity function performs best in general, we compared the runtimes and the average improvement of the mean information content for several values of l and r. We used 1165 TFBMs of 85 transcription factors of *Escherichia coli* obtained from RegulonDB. The results are summarized in Table 3.1 The combination $(cg/simC)$ has the best runtime, but to gain the best information content improvement, one should use the combinations $(cg/simS)$ or $(km/simS)$. Figure B.1 (page 97) in Appendix Section B illustrates the relation between runtime and quality improvement for all combinations. In order to find good solutions shortly, it is recommendable to use the combination $(cg/simS)$, which often provides the best improvement and still has an appropriate runtime.

Adjusted TFBMs lead to better binding site predictions

As explained in Section 3.3 (page 35), PFMs and PWMs derived from TFBMs are often used to predict further TFBMs in a given set of DNA sequences, generally in sequences upstream of putatively regulated target genes or operons. In the following, the in Section 3.3 introduced software PoSSuMsearch is used to evaluate the prediction performance of (i) PWMs constructed from the original TFBMs extracted from the RegulonDB database and (ii) the MoRAine-adjusted PWMs. We show that by using MoRAine for preprocessing, the classification performance is significantly increased.

Datasets. We use the afore mentioned 1165 extracted TFBMs for 85 transcription factors from RegulonDB and construct 85 PWMs. Additionally, we obtained 3341 upstream sequences of all transcription units (TUs) of *E. coli* from CoryneRegNet. In CoryneRegNet, an upstream region is defined as that DNA sequence -560 to $+20$ bps upstream to the start codon of a TU (a gene, or an operon respectively). For every PWM we split these sequences into two sets: those with a known TFBM for the corresponding regulator (true positive) and those without a known TFBM, which we assume to be true negatives.

Classification performance. For each PWM, both forward and reverse strand of upstream sequences are used to predict TFBMs with PoSSuMsearch, using different p-value thresholds. For each threshold, we measure the fraction of false positives (FP := number of incorrectly predicted motifs in relation to all predicted motifs), false negatives (FN := number of not predicted motifs in relation to the number of all motifs in the reference list), and the accuracy (ACC := number of correctly predicted motifs in relation to all motifs in the reference list).

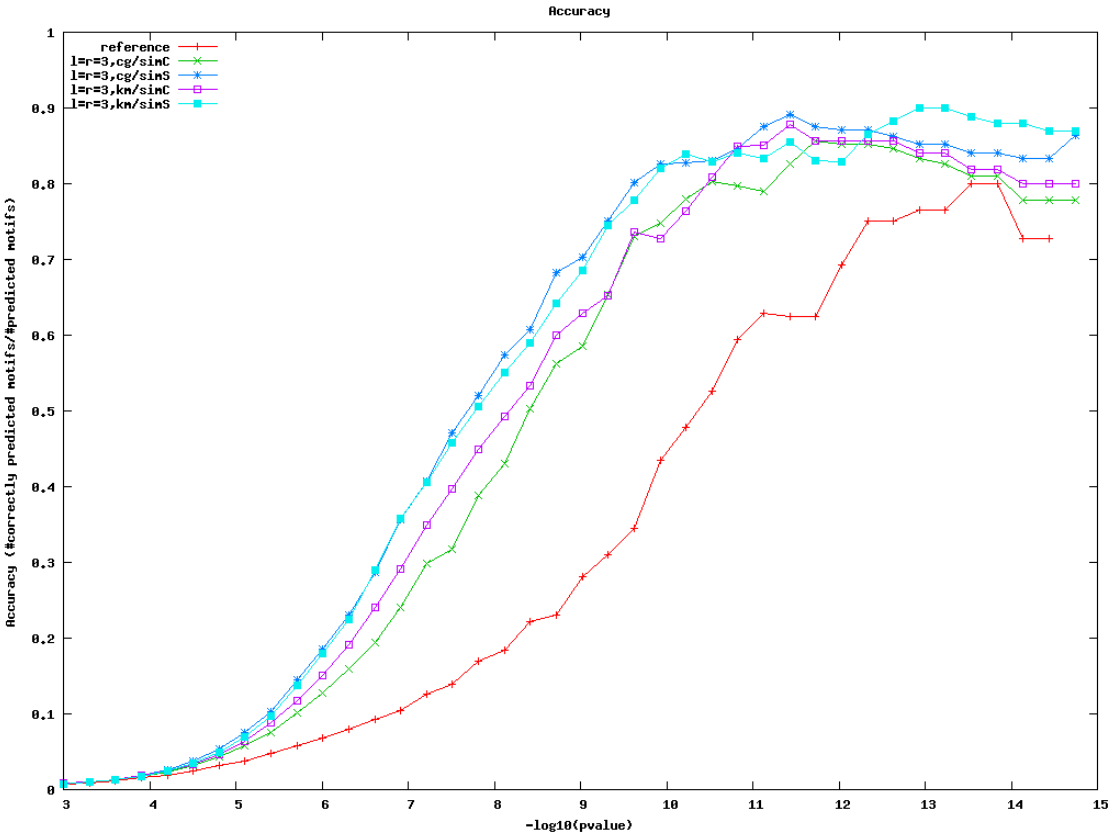


Figure 3.13: Percentage of accurately predicted motifs for different p-value thresholds, for $l = r = 3$. For other values of l and r refer to Figure B.1 in Appendix Section B, page 97. For the reference curve we used original PFMs learned from original database TFBMs.

Figure 3.13 shows the ACC for all PWMs adjusted with MoRAine for $l = r = 3$, for the four combinations of search algorithms and similarity functions, in comparison to the ACC obtained with the original PWMs built from the original database TFBMs (the reference curve). The measured relative ACC is plotted at different p-value thresholds. The ACC obtained with adjusted PWMs is always higher than with original PWMs. The ACC

for adjusted PWMs versus original PWMs for $0 \leq l = r \leq 7$ is plotted in Figure B.2 (page 98) in Appendix Section B. The prediction performance using adjusted PWMs always outperforms the reference. Generally the combination $(cg/simS)$ performs best.

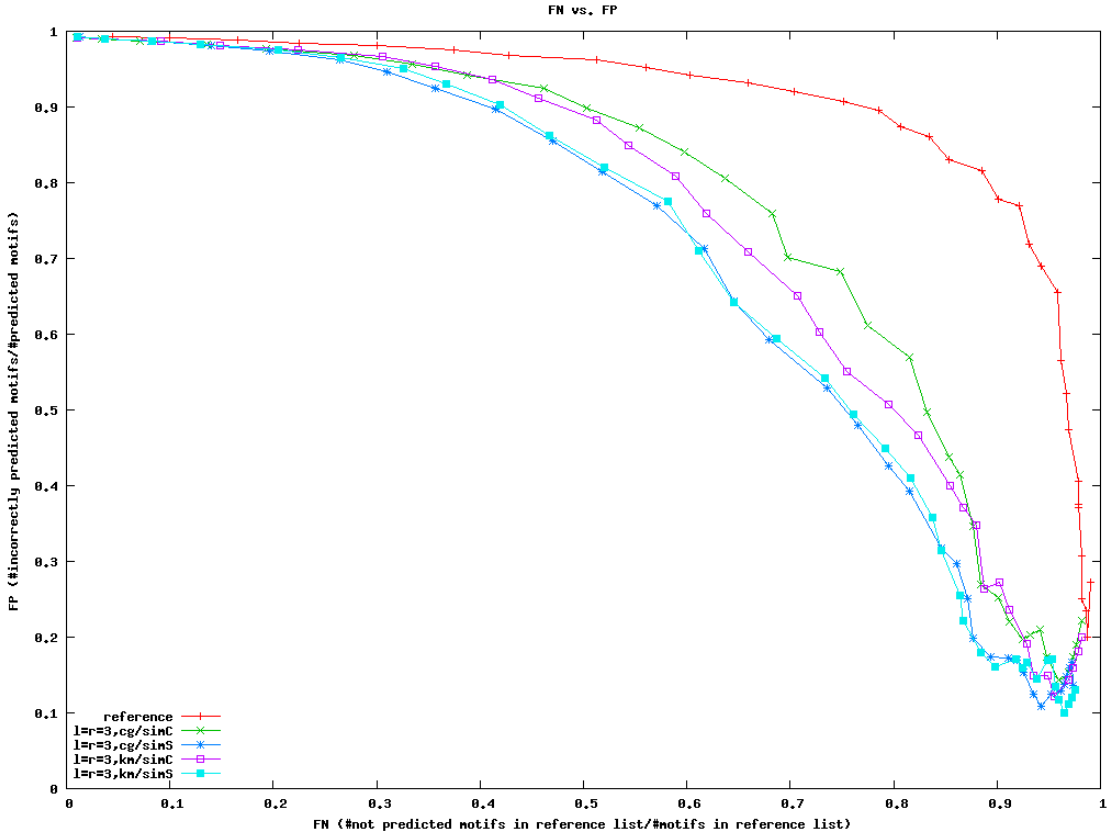


Figure 3.14: False negative (FN) vs. false positive (FP) rates for different p-value thresholds for $l = r = 3$. For other values of l and r refer to Figure B.3 in Appendix Section B, page 98. For the reference curve we used original PFMs learned from original database TFBMs.

Figure 3.14 plots the FN versus the FP rate ($l = r = 3$). The plots show that predictions based on adjusted PWMs outperform those based on original PWMs. The point where the FN rate equals the FP rate is at ≈ 0.65 (adjusted PWMs) and ≈ 0.85 (original PWMs). Again, one can see that the combination $(cg/simS)$ performs best. FN versus FP rates for $0 \leq l = r \leq 7$ are plotted in Figure B.3 (page 98) in Appendix Section B. There is a visible gap in performance between $l = r = 0$ and $l = r = 1$. Increasing l and r further has smaller effects.

3.4.3 Conclusions

Gene regulatory protein-DNA interactions are stored in databases, such as RegulonDB, CoryneRegNet, PRODORIC, or TRANSFAC, along with annotated transcription factor binding sites. These binding sites are manually curated and extracted from scientific literature. Usually, the corresponding binding sequences are stored $5' \rightarrow 3'$ relative to the target gene. Since the exact determination of the TFBM positions down to one basepair is difficult and the annotation of the TFBM strands is sometimes neglected, some of these TFBMs are manually reannotated (e.g. in PRODORIC for the regulators NarL and MalT in *E. coli*). This is both time-consuming and error-prone. Note that e.g. for $l = r = 0$

in ≈35% of all cases MoRAine suggests to switch the strand annotation from forward to reverse.

It should be mentioned that the presented algorithms are heuristics selected for their good running time performance and scalability and do not guarantee an optimal solution in all cases. The observed increase in information content, however, suggests that we generally get useful reannotations, and the speed of the algorithm allows it to be run on a non-dedicated web server.

Summarizing, MoRAine is a software that supports the automatic reannotation of TFBMs to increase the mean information content of a corresponding PFM. We provide a web server to facilitate using MoRAine and to compute sequence logos from transcription factor binding sites. We have demonstrated that a reliable strand annotation is necessary and helps to improve the PWM-based prediction performance. MoRAine-adjusted PWMs provide significantly more accurate classifications. Hence, MoRAine is used for the reannotation of TFBMs that are subsequently included in CoryneRegNet.

3.5 COMA - Contradictions in microarrays

The COMA feature is a novel option in the CoryneRegNet front-end to facilitate consistency checks in microarrays with known regulatory networks.

3.5.1 Method

Table 3.2: Artificial corynebacterial stimulon. This table shows a small, artificial stimulon, which can be applied to the consistency check feature of CoryneRegNet. Expression values are given as M-values. In the last column, we list those transcription factors, which control the gene in the first column. We denote (R) as repression and (A) as activation respectively. Refer to Figure 3.15 for a visualization.

Gene	GeneID	Operon	M-value	Regulated by
ramB	*cg0444*	-	1.9	(R) *ramB*, (A) *ramA*
sdhCD	*cg0445*	*OP_cg*0445	-1.8	(R) *ramB*, (R) *ripA*, (A) *dtxR* , (A) *ramA*
sdhA	*cg0446*	*OP_cg*0445	1.8	(R) *ramB*, (R) *ripA*, (A) *dtxR* , (A) *ramA*
sdhB	*cg0447*	*OP_cg*0445	-2.5	(R) *ramB*, (R) *ripA*, (A) *dtxR* , (A) *ramA*
-	*cg0448*	*OP_cg*0445	-1.7	(R) *ramB*, (R) *ripA*, (A) *dtxR* , (A) *ramA*
ramA	*cg2831*	-	-1.6	(R) *ramA*

In order to analyze a microarray experiment in the context of stored gene regulatory networks, the user has three possibilities to enter gene expression data:

- Copy+paste into a text field.
- Upload a TAB-delimited flat file.
- Usage of stimulon data from the CoryneRegNet database.

For further analyses we discretize the expression levels to upstimulations and downstimulations respectively. We denote an upstimulation with $'+'$ and a downstimulation with $'-'$ respectively. The same can be done to an activation or repression of a target gene by

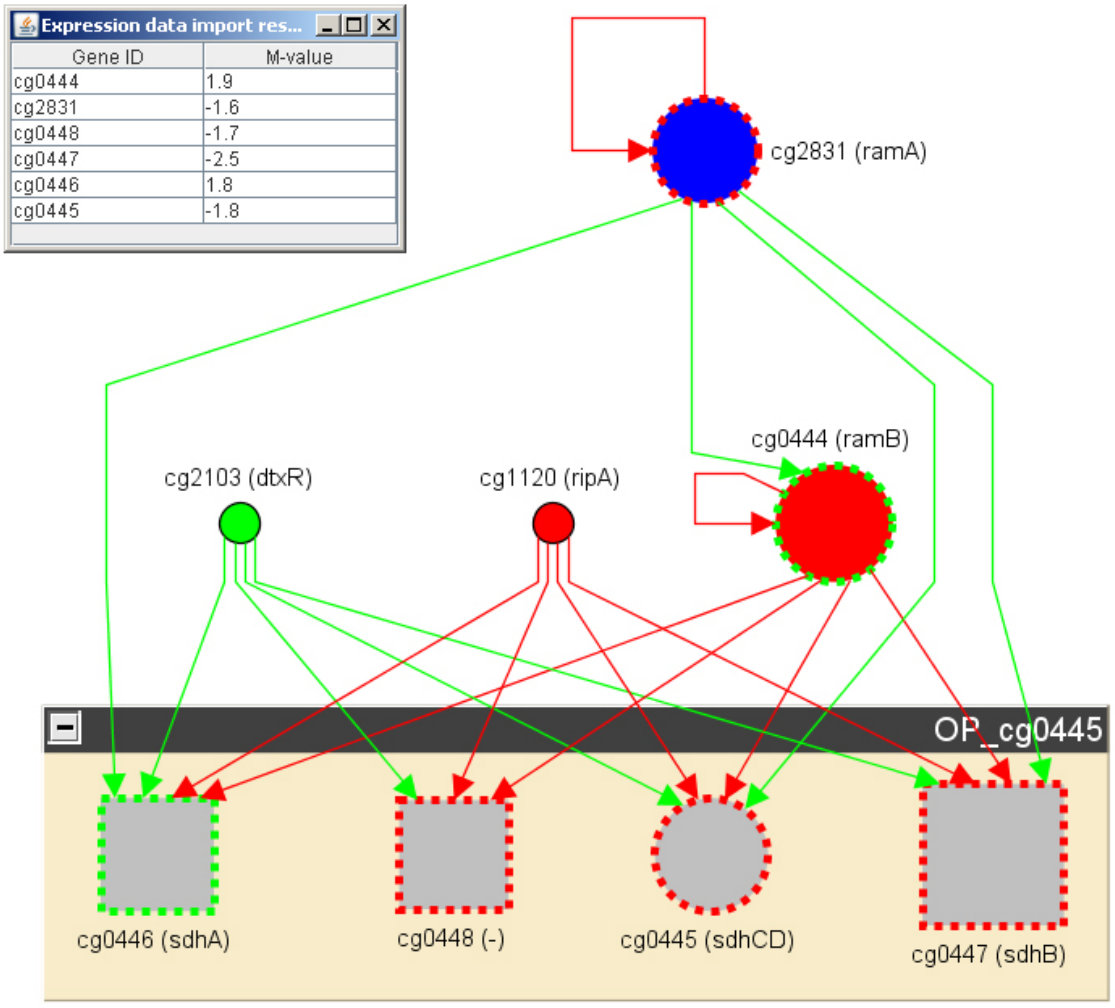


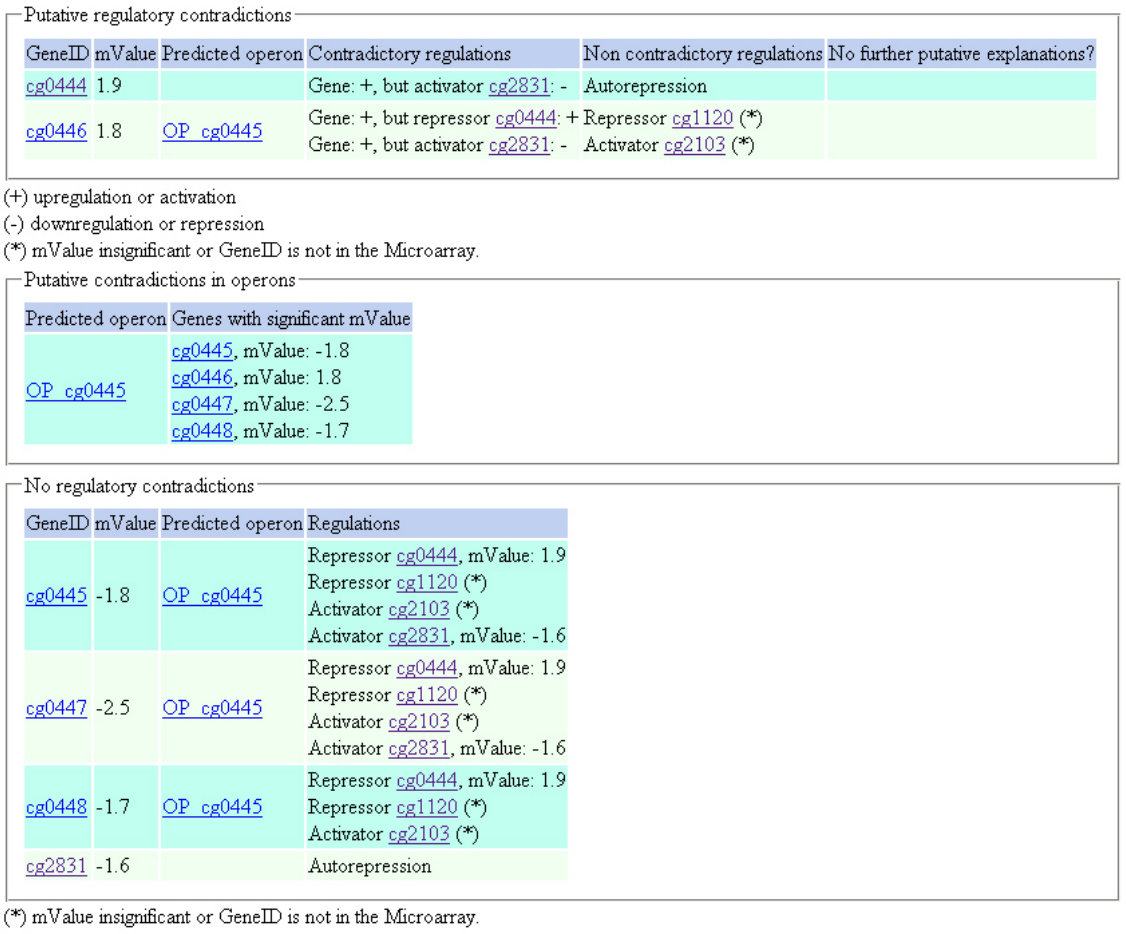
Figure 3.15: An artificial stimulon. This screenshot shows the improved network analysis and visualization feature GraphVis. Presented is the artificial stimulon of Table 3.2 projected onto the underlying gene regulatory network. The nodes represent genes and the edges gene regulations. Red nodes are repressors, green nodes activators, and blue nodes dual regulators. Gray nodes are target genes. A red edge represents a repression and a green edge an activation. The nodes sizes are relative to the expression value (M-value): the bigger the node, the more the differential expression of the respective gene. Genes can be upstimulated (green dotted node border) or downstimulated (red dotted border). The big multi-node represents an operon. The circular node inside the operon is that gene, which is preceded by a transcription factor binding site.

a regulator. Let $g \in \{+, -\}$ be the stimulation state of a gene G. Let $t \in \{+, -\}$ be the stimulation state of the transcription factor T, which regulates G. Let $r \in \{+, -\}$ be the type of the known regulation of G by T. Now consider the algebraic signs in the following equation: $t \cdot g = r$. If the equation is incorrect (e.g. $'+' \cdot '-' = '+'$) we define this as an inconsistency. Following this, for every gene G of a given microarray experiment with expression state g, CoryneRegNet queries the database and retrieves all transcription factors T, which regulate G. Subsequently, we check for all transcription factors the expression state t and the regulation relationship r and apply the above explained inconsistency test. For every inconsistent measurement, we also report, if other transcription factors regulate the gene G and hence possibly could explain the inconsistent expression

level. Furthermore, we test, if all genes within all predicted operons in CoryneRegNet are regulated identical (all $'+'$ or all $'-'$) and report them otherwise.

For simplification, we explain this by means of an artificial example. Consider the small stimulon experiment in Table 3.2 and its visualization in Figure 3.15. One can see 3 putative contradictions: (i) The gene *sdhA* is upregulated, while all the other genes in the same predicted operon are downregulated. (ii) The gene *ramB* is upregulated, but the activator *ramA* is downregulated. (iii) The gene *sdhA* is upregulated, while the activator *ramA* is downregulated and the repressor *ramB* is upregulated.

3.5.2 Results



Putative regulatory contradictions

GeneID	mValue	Predicted operon	Contradictory regulations	Non contradictory regulations	No further putative explanations?
cg0444	1.9		Gene: +, but activator cg2831: -	Autorepression	
cg0446	1.8	OP_cg0445	Gene: +, but repressor cg0444: + Gene: +, but activator cg2831: -	Repressor cg1120 (*) Activator cg2103 (*)	

(+) upregulation or activation
(-) downregulation or repression
(*) mValue insignificant or GeneID is not in the Microarray.

Putative contradictions in operons

Predicted operon	Genes with significant mValue
OP_cg0445	cg0445, mValue: -1.8 cg0446, mValue: 1.8 cg0447, mValue: -2.5 cg0448, mValue: -1.7

No regulatory contradictions

GeneID	mValue	Predicted operon	Regulations
cg0445	-1.8	OP_cg0445	Repressor cg0444, mValue: 1.9 Repressor cg1120 (*) Activator cg2103 (*) Activator cg2831, mValue: -1.6
cg0447	-2.5	OP_cg0445	Repressor cg0444, mValue: 1.9 Repressor cg1120 (*) Activator cg2103 (*) Activator cg2831, mValue: -1.6
cg0448	-1.7	OP_cg0445	Repressor cg0444, mValue: 1.9 Repressor cg1120 (*) Activator cg2103 (*)
cg2831	-1.6		Autorepression

(*) mValue insignificant or GeneID is not in the Microarray.

Figure 3.16: Result of the COMA feature applied to an artificial stimulon. This screenshot shows the result page of the COMA feature if applied to the artificial stimulon of Table 3.2. There are three putative contradictions: two for *sdhA* (*cg0446*) and one for *ramB* (*cg0444*). For both genes, there are further transcriptional regulators listed that possibly could resolve the contradictions.

The integrated COMA feature facilitates with consistency checks in microarray results. The method provides hints for incorrectly predicted operons, missing gene regulatory interactions, and putative mistakes in the experimental setup. Microarray results can be uploaded or copy+pasted easily and subsequently are checked for consistency with the known gene regulations of the integrated bacteria.

If the experimental gene expression results given in Figure 3.15 and Table 3.2 are copy+pasted into the COMA feature input textfield, we retrieve the same results au-

tomatically (refer to the screenshot shown in Figure 3.16). For such an expression profile, one would argue if something went wrong with the experiment, or if there are further yet unknown regulatory interactions. It is obvious that the method also helps with the improvement of the predicted operons.

3.6 FORCE - Protein sequence clustering

To extend the comparative features of CoryneRegNet, we need adequate data on gene and protein clusters. The integration of this information widens the scope of CoryneRegNet and assists the user with the reconstruction of unknown regulatory interactions.

Here we describe the clustering problem and the techniques that we used to attack it, along with appropriate scoring schemes for the given purpose. An evaluation of the clustering model and the final implementation is given afterwards.

3.6.1 Method

The clustering problem

High-throughput genome sequencing projects have generated massive amounts of DNA and protein sequence data, and will do so more rapidly in the near future. One major challenge continues to be determining protein functions based solely on amino acid sequences. Large-scale pairwise sequence comparison directly results in pairwise similarity measures between protein sequences and is an efficient method to transfer biological knowledge from known proteins to newly sequenced ones. The most widely used method to search for sequence similarities is BLAST [2]. Three challenges arise:

1. Deriving a quantitative similarity measure from the sequence comparison that models homology as well as possible; frequently this is based on the negative logarithm of the BLAST E-value.

2. Inventing a clustering strategy that is sufficiently error-tolerant, since experience shows that sequence similarity alone does not lead to perfect clusterings. A common approach is to use a graph-based model, where proteins are represented as nodes and the similarities as weighted edges.

3. Implementing the chosen clustering strategy efficiently.

We note that many approaches do treat the three challenges separately. Here,

1. we use a family of different similarity functions, based on negative logarithms of BLAST E-values and sequence coverage.

2. we show that *weighted graph cluster editing* is an adequate model to identify protein clusters. But weighted graph cluster editing is known to be NP-hard [108].

3. we present a heuristic called FORCE to solve the problem. We show that it provides excellent quality results in practice when compared with an exponential-time exact algorithm, but has a running time that makes it applicable to massive datasets.

The weighted graph cluster editing problem

To specify the clustering model, we need the following definition: An undirected simple graph $G = (V, E)$ is called **transitive** if

$$\text{for all triples } uvw \in \binom{V}{3}, \quad uv \in E \text{ and } vw \in E \text{ implies } uw \in E.$$

A transitive graph is a union of disjoint cliques, i.e., of complete subgraphs. Each clique represents, in our case, a protein cluster. Since the initial graph, derived from protein similarity values and a similarity threshold, may not be transitive, we need to modify it. This leads to the following computational problems.

Graph cluster editing problem (GCEP) Given an undirected graph $G = (V, E)$, find a transitive graph $G^* = (V, E^*)$, with minimal edge modification distance to G, i.e., where $|E \setminus E^*| + |E^* \setminus E|$ is minimal.

Weighted graph cluster editing problem (WGCEP) To respect the similarity between two proteins, we modify the penalty for deleting and adding edges. First we construct a similarity graph $G = (V, E)$ consisting of a set of objects V and a set of edges $E := \{uv \in \binom{V}{2} : s(uv) > t\}$. Here $s\colon \binom{V}{2} \to \mathbb{R}$ denotes a similarity function and t a user-defined threshold. The resulting cost to add or delete an edge uv is set to $cost(uv) := |s(uv) - t|$. The cost to transform a graph $G = (V, E)$ into a graph $G' = (V, E')$ is consequently defined as $cost(G \to G') := cost(E \setminus E') + cost(E' \setminus E)$. As in the GCEP, the goal is to find a transitive graph $G^* = (V, E^*)$, with $cost(G \to G^*) = \min\{cost(G \to G') : G' = (V, E') \text{ transitive}\}$.

It can be easily seen that the WGCEP is NP-hard, since it is a straightforward generalization of the GCEP, where $s\colon \binom{V}{2} \to \{-1, 1\}$ and $t = 0$. The GCEP has been proved to be NP hard several times, e.g., in [35, 120].

The FORCE heuristic

We present an algorithm called FORCE that heuristically solves the WGCEP for a connected component and thus for a whole graph. FORCE is motivated by a physically inspired force-based graph layout algorithm developed by Fruchterman and Reingold [46]. The main idea of this approach is to find an arrangement of the vertices in a two-dimensional plane that reflects the edge density distribution of the graph, i.e., vertices from subgraphs with high intra-connecting edge weights should be arranged close to each other and far away from other nodes. This layout is then used to define the clusters by Euclidean single-linkage clustering of the vertices' positions in the plane. To improve the solution, we implemented an additional postprocessing phase. All in all the algorithm proceeds in three main steps: (i) layouting the graph, (ii) partitioning, and (iii) postprocessing.

Layout phase The goal in this phase is to arrange the vertices in a two-dimensional plane, such that the similarity values are respected. Subsets of nodes with high edge-density should be arranged next to each other, and far away from other nodes. To find a

layout that satisfies this criterion, we use a model inspired by physical forces, i.e., nodes can attract and repulse each other. Starting with an initial layout (a circular layout with user defined radius ρ and random order), the nodes affect each other depending on their similarity and current position, which leads to a displacement vector for each node and a new arrangement. Since this model is only inspired by physical forces without friction, it does not include acceleration.

For a user-defined number of iterations R, the interaction between every pair of nodes and thus the displacement for every node is calculated; then all nodes are simultaneously moved to their new position.

We compute the displacements as follows: As described in Algorithm 3, the strength $f_{u \leftarrow v}$ of the effect of one node v to another node u (i.e., the magnitude of the displacement of u caused by v) depends on the Euclidean distance $d(u, v)$, on the cost to add or delete the edge and a user defined attraction or repulsion factor f_{att}, f_{rep}. More formally,

$$f_{u \leftarrow v} = \begin{cases} \dfrac{cost(uv) \cdot f_{\text{att}} \cdot \log(d(u,v)+1)}{|V|} & \text{for attraction,} \\ \dfrac{cost(uv) \cdot f_{\text{rep}}}{|V| \cdot \log(d(u,v)+1)} & \text{for repulsion.} \end{cases}$$

Two nodes attract each other if $s(uv) > t$ and repulse each other otherwise. One can see that with increasing distance, attraction strength increases while repulsion strength decreases.

To improve convergence to a stable position with minimal interactions, we added a cooling parameter, also inspired by the algorithm of Fruchterman and Reingold. In our implementation, this means that if the displacement distance exceeds a maximal magnitude M_i in iteration i, which starts at an initial value M_0 and decreases with every iteration i, the movement is limited to it.

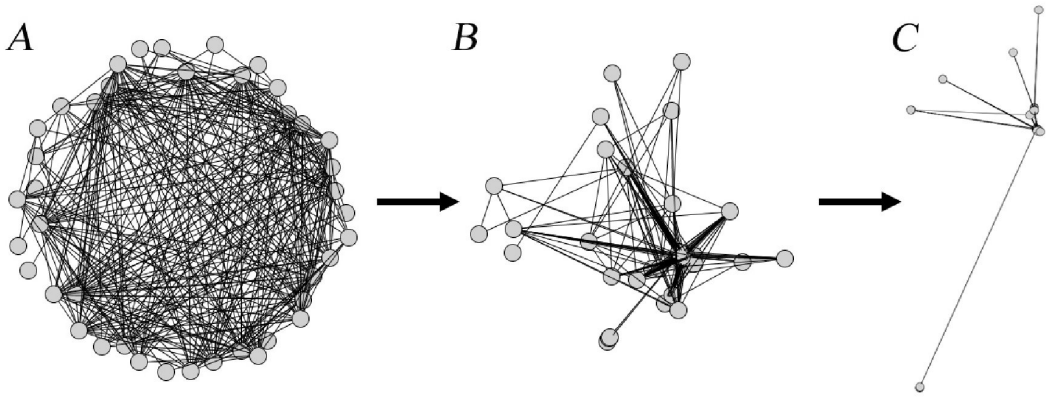


Figure 3.17: The layout process of a graph with 41 nodes after (A) 3, (B) 10, and (C) 90 iterations.

The output of this phase is a two-dimensional array *pos* containing the x-y-position of each node. The Figures 3.17 and 3.18 illustrate the layout process and its convergence for two components with 41 and 10 nodes, respectively.

Partitioning phase Using the positions of the vertices from the layout phase, we define clusters by geometric single-linkage clustering, parameterized by a maximal node distance

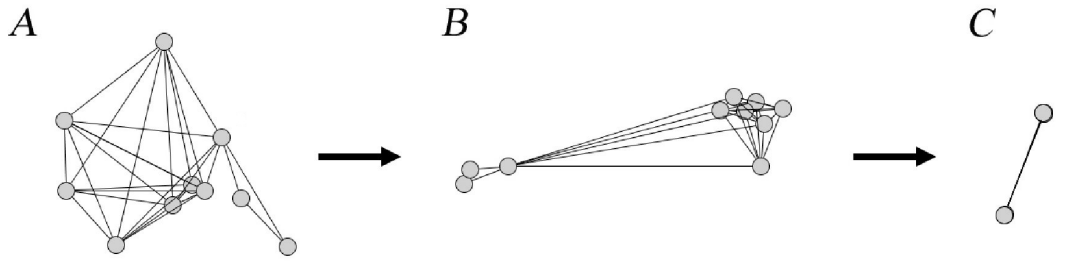


Figure 3.18: The layout process of a graph with 10 nodes after (A) 3, (B) 10, and (C) 40 iterations.

Algorithm 3 Graph layouting

Input: similarity matrix $(S_{ij})_{1 \leq i < j \leq n}$ with $S_{ij} := s(ij) - t$; circular layout radius ρ, attraction factor f_{att}, repulsion factor f_{rep}, number of iterations R

Output: node positions $pos = (pos[1], \ldots, pos[n])$; each $pos[i] \in \mathbb{R}^2$.

$pos = arrangeAllNodesCircular(\rho)$ ▷ *initial layout*
for $r = 1$ to R **do**
    ▷ *Compute displacements* Δ *for iteration* r
    initialize array $\Delta = (\Delta[1], \ldots, \Delta[n])$ of displacement vectors to $\Delta[i] = (0, 0)$ for all i
    for $i = 1$ to n **do**
        for $j = 1$ to $i - 1$ **do**
            if $S_{i,j} > 0$ **then**
                $f_{i \leftarrow j} = \log(d(i,j) + 1) \cdot S_{i,j} \cdot f_{\text{att}}$ ▷ *attraction strength*
            else
                $f_{i \leftarrow j} = (1/\log(d(i,j) + 1)) \cdot S_{i,j} \cdot f_{\text{rep}}$ ▷ *repulsion strength*
            $\Delta[i] \mathrel{+}= f_{i \leftarrow j} \cdot (pos[j] - pos[i])/d(i,j)$
            $\Delta[j] \mathrel{-}= f_{i \leftarrow j} \cdot (pos[j] - pos[i])/d(i,j)$
    ▷ *Move nodes by capped displacement vectors*
    for $i = 1$ to n **do**
        $\Delta[i] = (\Delta[i]/\|\Delta[i]\|) \cdot \min\{\|\Delta[i]\|, M(r)\}$
        $pos[i] \mathrel{+}= \Delta[i]$
return *pos*

δ. As described in Algorithm 4, we start with an arbitrary node $v_1 \in V$ and define a new cluster c_{v_1}. A node i belongs to c_{v_1} if there exist nodes $v_1 = i_0, \ldots, i_N = i \in V$ with $d(i_j, i_{j+1}) \leq \delta$ for all $j = 0, \ldots, N-1$. Nodes are assigned to c_{v_1} until no further nodes satisfy the distance cutoff. Then the next, not yet assigned, node $v_2 \in V$ is chosen to start a new cluster until every node is assigned to some cluster. We denote with $G_\delta := \bigcup_{j=1}^{m} c_{v_j}$ the resulting graph obtained by adding all edges between two nodes of the same cluster and deleting all edges between two nodes of different clusters. To find a good clustering we calculate $cost(G \to G_\delta)$ for different δ. Starting with $\delta \leftarrow \delta_{\text{init}} := 0$ we increase δ by a step size σ up to a limit $\delta_{\max} := 300$. Experimentation shows that it is beneficial to also increase the step size, i.e. to start with $\sigma \leftarrow \sigma_{\text{init}} := 0.01$ and increase it by multiplying with a user-defined factor $f_\sigma := 1.1$. The solution with lowest cost is returned as the resulting clustering. Algorithm 4 returns the clustering in terms of an $n \times n$ adjacency matrix $E^* \in \{0,1\}^{n \times n}$ and the transformation cost c^*.

Postprocessing phase Although the best clustering is not guaranteed to be the optimal one, we often obtain a close to optimal solution in practice. To further improve the results

Algorithm 4 Partitioning the layouted graph

Input: layout positions *pos*, initial and maximal clustering distances δ_{init}, $\delta_{\max}$, initial step size σ_{init}, step size factor f_σ, similarity matrix $(S_{ij})_{1 \le i < j \le n}$ to compute costs

Output: best found $n \times n$ adjacency matrix E^* describing a clustering, associated cost c^*

```
δ = δ_init, σ = σ_init, c* = ∞, E* = (0)^{n×n}
while δ ≤ δ_max do
    construct auxiliary graph G_δ = (V, E_δ) with E_δ := {uv : d(u, v) ≤ δ}
    detect connected components of G_δ
    compute transitively closed adjacency matrix E' from E_δ
    if cost(E') < c* then
        E* = E'; c* = cost(E')
    σ = σ · f_σ; δ = δ + σ
return (E*, c*)
```

we use a two-step postprocessing heuristic. We denote with $cost(C)$ the cost to obtain the clustering C.

1. To reduce the number of clusters and especially the number of singletons, the first step is to join two clusters if this reduces the overall cost:
   Let $C := (c_1, \ldots, c_n)$ be the clustering obtained from the partitioning phase, ordered by size. For all cluster pairs $1 \le i < j \le n$ we calculate $cost\,(c_1, \ldots, c_i \cup c_j, \ldots, c_n)$ until we find a clustering $C' := \big(c_1, \ldots, c_{i'} \cup c_{j'}, \ldots, c_n\big)$ with $cost(C') < cost(C)$. Let $\big(c'_1, \ldots, c'_{n-1}\big)$ be the sorted vector C'. Repeat to attempt joining more clusters until no more join is beneficial.

2. Similar to the Restricted Neighborhood Search Clustering [74], we move a vertex from one cluster to another if this move reduces the overall cost:
   As above, let $C := (c_1, \ldots, c_n)$ be the clustering obtained from step 1, ordered by size. For $i, j \in \{1, \ldots, n\}$, $i \ne j$, and every $k \in c_i$, we tentatively move k from c_i to c_j and calculate $cost\,(c_1, \ldots, c_i \setminus \{k\}, \ldots, c_j \cup \{k\}, \ldots, c_n)$, until we find the first such modified clustering with lower cost than $cost(C)$. We sort the resulting clusters again by size and use them as a new start configuration for the next iteration until no more reassignments are beneficial.

Analysis The worst-case running time of FORCE is given by the addition of those of the three main phases.

Layouting runs in $\Theta(R \cdot n^2)$, where R denotes the number of iterations and n is the number of nodes in the graph. Since R is determined by evolutionary training (see below), it might grow with n, but we set an upper bound for R to $R_{max} = 500$ that in practice suffices even for very large datasets.

Partitioning runs in $O(D \cdot n^2)$, where D is the number of different δ-values used. This is seen as follows: Each δ-value requires the construction of an auxiliary graph in $O(n^2)$ time, the discovery of its connected components in $O(|V| + |E_\delta|) = O(n^2)$ time, setting E' to the transitive closure of E_δ and computing its cost, which is also possible in $O(n^2)$ after detecting connected components.

During postprocessing, each iteration takes $O(n^2)$ time, since the number of clusters is bounded by n. The total time is thus $O(P \cdot n^2)$, where P is the number of postprocessing

iterations. While theoretically P can grow with n, in practice we observe only a small number of iterations until no more improvement occurs.

Thus for all practical purposes, the overall runtime of FORCE is quadratic in the number of nodes.

Evolutionary parameter training There are several user-defined parameters to assign, such as the number of iterations R, the attraction and repulsion scaling factors $f_{\text{att}}, f_{\text{rep}}$, the magnitude M_0, and the initial circular layout radius ρ. A practical method to find good values is evolutionary training. FORCE implements such a strategy in two different ways.

First, a good parameter combination is determined that can be applied to most of the graphs. This is done during a pre-computation on a training data set. Since, however, the optimal parameter constellation depends on the specific graph, we additionally apply such a training algorithm to each graph. FORCE allows to specify the number of generations to train and thus to adjust runtime and the quality of the result.

Training works as follows: First we start with a set of 25 randomly generated parameter sets and the initial parameters mentioned above. The parameter sets are sorted by the cost to solve the WGCEP on the given graph. For each generation, we use the best 10 parameter constellations as parents, to generate 15 new combinations. In order to obtain fast convergence to a good constellation, as well as a wide spectrum of different solutions without running into local minima, FORCE splits these 15 new combinations into 3 groups, with 5 members each. The first group consists of parameters obtained only by random combinations of the 10 best already known parameter constellations. The next group is generated with random parameters, while the third group is obtained by a combination of the previous methods. To reduce the runtime for small or very easy to compute solutions, we added a second terminating condition: If at most two different cost appear while calculating the 25 start parameters, the best one is chosen. No more generations are computed.

Similarity functions for amino acid sequences and parameter choices

Similarity functions Any attempt to (optimally) solve the WGCEP would be in vain if the target function did not model our goal appropriately. As mentioned earlier, the main challenge is to identify appropriate similarity functions and thresholds. We have used a variety of similarity functions that we describe below.

Assume we are given a set of proteins V and a BLAST output file containing multiple high-scoring pairs (HSPs) in both directions. For two proteins u and v we denote with $(u \leftarrow v)_i$ and $(u \rightarrow v)_j$, where $i = 1, \ldots, k$ and $j = 1, \ldots, l$, the corresponding k HSPs in one and l HSPs in the other direction, respectively.

We consider the following three similarity functions.

Best hit (BeH) This widely used method concentrates on the E-value of a single HSP: For both directions, one looks for the best hit, i.e., the HSP with lowest E-value. To obtain a symmetric similarity function $s\colon \binom{V}{2} \rightarrow \mathbb{R}$, the negative logarithm of the worst (largest) of the two E-values is taken as similarity measure between u and v.

The resulting symmetric similarity function is then defined as

$$s\left(uv\right) := -\log_{10}\left(\max\left\{\min_{i=1,\ldots,k}\text{E-value}\left(\left(u \leftarrow v\right)_i\right),\ \min_{j=1,\ldots,l}\text{E-value}\left(\left(u \rightarrow v\right)_j\right)\right\}\right).$$

Sum of hits (SoH) This approach is similar to BeH, but additionally includes every HSP with an E-value smaller than a threshold $m = 10^{-2}$. We use this threshold as penalty for every additional HSP. This leads to the similarity function

$$s\left(uv\right) := -\log_{10}\left(\max\left\{m^{-(k-1)}\cdot\prod_{i=1}^{k}\text{E-value}\left(\left(u \leftarrow v\right)_i\right),\ m^{-(l-1)}\cdot\prod_{j=1}^{l}\text{E-value}\left(\left(u \rightarrow v\right)_j\right)\right\}\right).$$

Coverage (Cov) The third approach integrates the lengths of a HSP into the similarity function. To determine the coverage, we need the following indicator function:

$$\mathbb{I}_{uv}(i) := \begin{cases} 1 & \text{if in } u \text{ the position } i \text{ is covered by any HSP } \left(u \leftarrow v\right)_{n=1,\ldots,k} \text{ or } \left(u \rightarrow v\right)_{m=1,\ldots,l}, \\ 0 & \text{otherwise.} \end{cases}$$

The coverage can now be defined as

$$\text{coverage}\left(uv\right) := \min\left(\frac{1}{|u|}\sum_{i=1}^{|u|}\mathbb{I}_{uv}(i), \frac{1}{|v|}\sum_{i=1}^{|v|}\mathbb{I}_{vu}(i)\right).$$

In order to obtain a good similarity function, we control the influence of the coverage on the overall similarity function by a user-defined factor f, and set

$$s(uv) := s'(uv) + f \cdot \text{coverage}(uv).$$

Here $s'\colon \binom{V}{2} \rightarrow \mathbb{R}$ denotes one of the previously presented similarity functions, BeH or SoH.

Parameter choices The initial parameters obtained from the pre-processing training are $R = 186$, $f_{\text{att}} = 1.245$, $f_{\text{rep}} = 1.687$, $M_0 = 633$, and $\rho = 200$ for the protein clustering problem. Furthermore, we apply evolutionary training to each problem instance, as described earlier.

Integration into CoryneRegNet Using the FORCE heuristic, we calculated protein clusters for all organisms integrated in CoryneRegNet (altogether 22,797 proteins). Based on cluster size distribution, we empirically determined a comparatively high threshold of 30 (which can be explained by the relatively close evolutionary relationship of most organisms in CoryneRegNet) and similarity function SoH to create the FORCE input files based on the all-vs-all BLAST results that are generated during CoryneRegNet's data warehousing process.

The results computed by FORCE are parsed into the object oriented back-end and further on translated into the ontology-based data structure of CoryneRegNet. We added a new concept class $FORCECluster$ and a relation type b_fc (belongs to FORCECluster), which links the proteins to their clusters. Finally, we adapted the CoryneRegNet back-end to import the new data into the database and the web-front-end to present the clusters.

3.6.2 Results

There are several approaches to cluster protein families. One of the earliest approaches that took the transitivity concept formally into account was ProClust [105]; however, the concept of editing the graph was not present in this work. The SYSTERS database [78], now at release 4, is based on a set-theoretic SYSTEmatic ReSearching approach and has existed for some time, but seems to have received little updates since early 2005. One of its main features is that it uses family-specific similarity thresholds to define clusters. It does not, however, employ a transitivity concept. In 2006, Paccanaro et al. [101] presented a comparison of the most popular cluster detection methods, like MCL [39], hierarchical clustering [43], GeneRAGE [40], and their own spectral clustering approach, which performs best when evaluated on a subset of the SCOP database. To evaluate our clustering model, we use the same datasets and performance figure. We furthermore include the recently published Affinity Propagation method in our comparison [45]. Additionally, we evaluate our approach against the COG database.

Here we first describe the datasets used for the subsequent evaluation. First the ASTRAL dataset from SCOP, as used in [101], is introduced. We also describe a considerably larger dataset obtained from the COG database. BLAST is used for all-against-all similarity searches in all datasets that can also be downloaded from the FORCE website.

Evaluation datasets

SCOP and Astral95 SCOP is an expert, manually curated database that groups proteins based on their 3D structures. It has a hierarchical structure with four main levels (class, fold, superfamily, family). Proteins in the same class have the same type(s) of secondary structures. Proteins share a common fold if they have the same secondary structures in the same arrangement. Proteins in the same superfamily are believed to be evolutionarily related, whereas proteins in the same family exhibit a clear evolutionary relationship [3]. We take the SCOP superfamily classification as ground truth against which we evaluate the quality of a clustering generated by a given algorithm, using reasonable quality measures, such as the F-measure (see below). Since the complete SCOP dataset contains many redundant domains that share a very high degree of similarity, most researchers choose to work with the ASTRAL compendium for sequence and structure analysis in order to generate non-redundant data [26]. ASTRAL allows to select SCOP entries that share no more sequence similarity than a given cutoff, removing redundant sequences.

We extracted two subsets of the ASTRAL dataset of SCOP v1.61 with a cutoff of 95 percent, which means that no two protein sequences share more than 95% of sequence identity. We consider ASTRAL95 as the best possible available reference for remote homology detection on a *structural* basis.

The two subsets are exactly those used in Paccanaro et al.'s work [101]. The first comprises 507 proteins from six different SCOP superfamilies, namely *Globin-like*, *EF-hand*, *Cupredoxins*, *(Trans)glycosidases*, *Thioredoxin-like*, and *Membrane all-alpha*. We refer to this dataset as ASTRAL95_1_161.

Due to the fact that SCOP is continuously updated, we decided to evaluate both the original data from [101] (SCOP v1.61) and more recent data from the current SCOP version (SCOP v1.71). The novel version is slightly different. For example, the superfamily

Membrane all-alpha has been removed in the meantime and most of its proteins are assigned to different superfamilies. Also, several other proteins have been reassigned to one of the five other superfamilies. This provides another dataset of 589 sequences from the remaining 5 superfamilies, which we refer to as ASTRAL95_1_171.

The second subset consists of 511 sequences from 7 superfamilies, namely *Globin-like*, *Cupredoxins*, *Viral coat and capsid proteins*, *Trypsin-like serine proteases*, *FAD/NAD(P)-binding domain*, *MHC antigen-recognition domain*, and *Scorpion toxin-like*. We refer to this as ASTRAL95_2_161 and ASTRAL95_2_171 respectively. SCOP can be found at `http://scop.mrc-lmb.cam.ac.uk/scop/`, while the protein sequences are available at `http://astral.berkeley.edu/`.

COG The Cluster of Orthologous Groups (COG) of proteins database is a repository whose main goal is a phylogenetic classification of proteins encoded by complete genomes. It currently consists of 192,187 procaryotic protein sequences from 66 complete genomes distributed across the three domains of life.

COG contains clusters in which at least three individual proteins (or groups of paralogs), originating from three different species, are each other's best BLAST hit in both directions. This strategy is believed to generate clusters of groups of orthologous proteins.

We consider COG as the best possible representation of orthology detection, based on *sequence* data alone. We refer to this dataset as the COG dataset. COG can be found at `http://www.ncbi.nlm.nih.gov/COG/`, while the protein sequences are available at `ftp://ftp.ncbi.nih.gov/pub/COG/COG/myva`.

Evaluation of the WGCEP model

To show that the WGCEP model is adequate for protein homology clustering, we evaluate our algorithm in the same way as Paccanaro et al. did in their article [101], using the so-called F-measure to quantify the agreement of FORCE's result with the reference clustering provided by the ASTRAL dataset.

We first explain the F-measure, which equally combines precision and recall. Let $K = (K_1, \ldots, K_m)$ be the clustering obtained from the algorithm and $C = (C_1, \ldots, C_l)$ the reference clustering. Furthermore, we denote with n the total number of proteins and with n_i, n^j the number of proteins in the cluster K_i and C_j, respectively. Following this, n_i^j is the number of proteins in the intersection $K_i \cap C_j$. The F-measure is defined as

$$F(K, C) := \frac{1}{n} \sum_{j=1}^{l} n^j \cdot \max_{1 \le i \le m} \left(\frac{2n_i^j}{n_i + n^j} \right).$$

As mentioned earlier, Paccanaro et al. previously compared the most popular protein clustering tools against their own spectral clustering: GeneRAGE, TribeMCL, and Hierarchical clustering. Since there is no need to replicate existing results, we use the same data (ASTRAL95_1_161 and ASTRAL95_2_161).

Table 3.3 summarizes the results: Using FORCE, we obtain slightly better agreements than with spectral clustering. The best combination of similarity function parameters and score threshold for the ASTRAL95_1_161 dataset were Cov-scoring using $f = 20$ and

Table 3.3: Evaluation of protein clustering tools. The F-measure (between 0 and 1) measures the agreement between a clustering resulting from a given algorithm and a reference clustering provided with the dataset. An F-measure of 1 indicates perfect agreement. ASTRAL95_1_161 and ASTRAL95_2_161 refer to the two datasets of SCOP v1.61 used by Paccanaro et al. for spectral clustering [101]. All reported values, except for our algorithm FORCE and for Affinity Propagation, are from the same reference.

Dataset	Method	F-measure
ASTRAL95_1_161	FORCE	0.85
ASTRAL95_1_161	Spectral clustering	0.81
ASTRAL95_1_161	Affinity Propagation	0.65
ASTRAL95_1_161	GeneRAGE	0.47
ASTRAL95_1_161	TribeMCL	0.32
ASTRAL95_1_161	Hierarchical clustering	0.26
ASTRAL95_2_161	FORCE	0.89
ASTRAL95_2_161	Spectral clustering	0.82
ASTRAL95_2_161	Affinity Propagation	0.69
ASTRAL95_2_161	GeneRAGE	0.54
ASTRAL95_2_161	TribeMCL	0.52
ASTRAL95_2_161	Hierarchical clustering	0.42

BeH as a secondary scoring function, and $t = -2.2$. For the ASTRAL95_2_161 dataset, this was Cov-scoring with $f = 19$ and SoH as secondary scoring function with $t = -1.6$.

Note that in the present context, we do not consider it as cheating to optimize the similarity function and threshold: We want to check how far the WGCEP model can retrieve the biologically correct clustering under ideal conditions. The same kind of optimization was applied by Paccanaro et al. in [101]. Table 3.3 also shows the F-measures for the Affinity Propagation (AP) approach, which was recently published in [45]. We used the same data and also varied necessary input parameters to evaluate against the best possible performance of AP. For ASTRAL95_1_161, this was Cov-scoring with $f = 20$ and SoH as secondary scoring function with fixed preference $pre = 600$, and damping factor $df = 0.8$. For ASTRAL95_2_161, this was Cov-scoring with $f = 14$ and SoH as secondary scoring function with $pre = 600$, and $df = 0.75$. For both datasets, AP performs worse than Spectral clustering.

Figure 3.19 exemplarily illustrates the obtained clustering results for two similarity functions and dataset ASTRAL95_1_161. A similar picture was presented by Paccanaro et. al in figure 3 in [101] (here it is given in Figure 3.20). One can see that the classification is very good for the superfamilies *Globin-like*, *EF-hand*, *Cupredoxins*, *(Trans)glycosidases*. *Thioredoxin-like* and *Membrane all-alpha* are split into several clusters. Note that for *Globin-like* (left column) using similarity function SoH (B), the superfamily is split into two clusters, where the second (the lower one) represents a family. Further note that in the actual version of SCOP (v1.71), the superfamily *Membrane all-alpha* has been removed and the proteins have been assigned to other superfamilies. Thus, our heuristic correctly partitions this superfamily.

We generated images in the same style for all datasets and provide them in the Appendix in Section C (page 99).

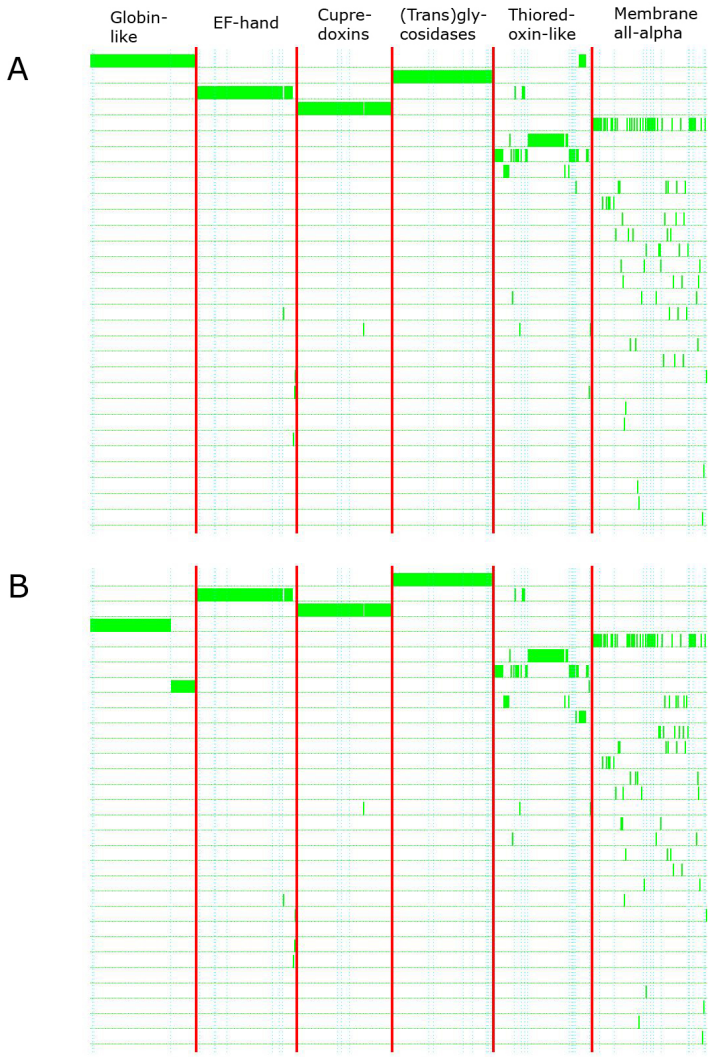


Figure 3.19: Graphical summary of the obtained clustering results of FORCE for the two similarity functions (A) BeH and (B) SoH, and dataset ASTRAL95_1_161. We used MATLAB scripts provided by Paccanaro et al. to create images similar to those of Figure 3.20. Each row corresponds to a cluster. Green bars represent a protein assignment to a cluster; each protein is present in only one of the clusters. Boundaries between superfamilies are shown by vertical red lines and boundaries between families within each superfamily are shown by dotted blue lines.

We additionally evaluate the FORCE heuristic with the newest ASTRAL95 datasets (ASTRAL95_1_171 and ASTRAL95_2_171). Table 3.4 shows the resulting F-measures for a variety of similarity functions and parameter choices. All of these achieve higher F-measures than Spectral clustering, or Affinity Propagation.

In the Appendix in Section D (page 104), we provide F-measures of FORCE for a wide range of thresholds and coverage factors, for all used datasets and similarity functions. Good clustering quality is also reached by using other thresholds and similarity measures for all test datasets. In the Appendix in Section D (page 104), we give F-measures for a range of thresholds, but with fixed coverage factor $f = 20$, for dataset ASTRAL95_1_161, and similarity function BeH. In Appendix Section E (page 107), we provide F-measures for Affinity Propagation for a wide range of parameters and coverage factors, for all used datasets and similarity functions.

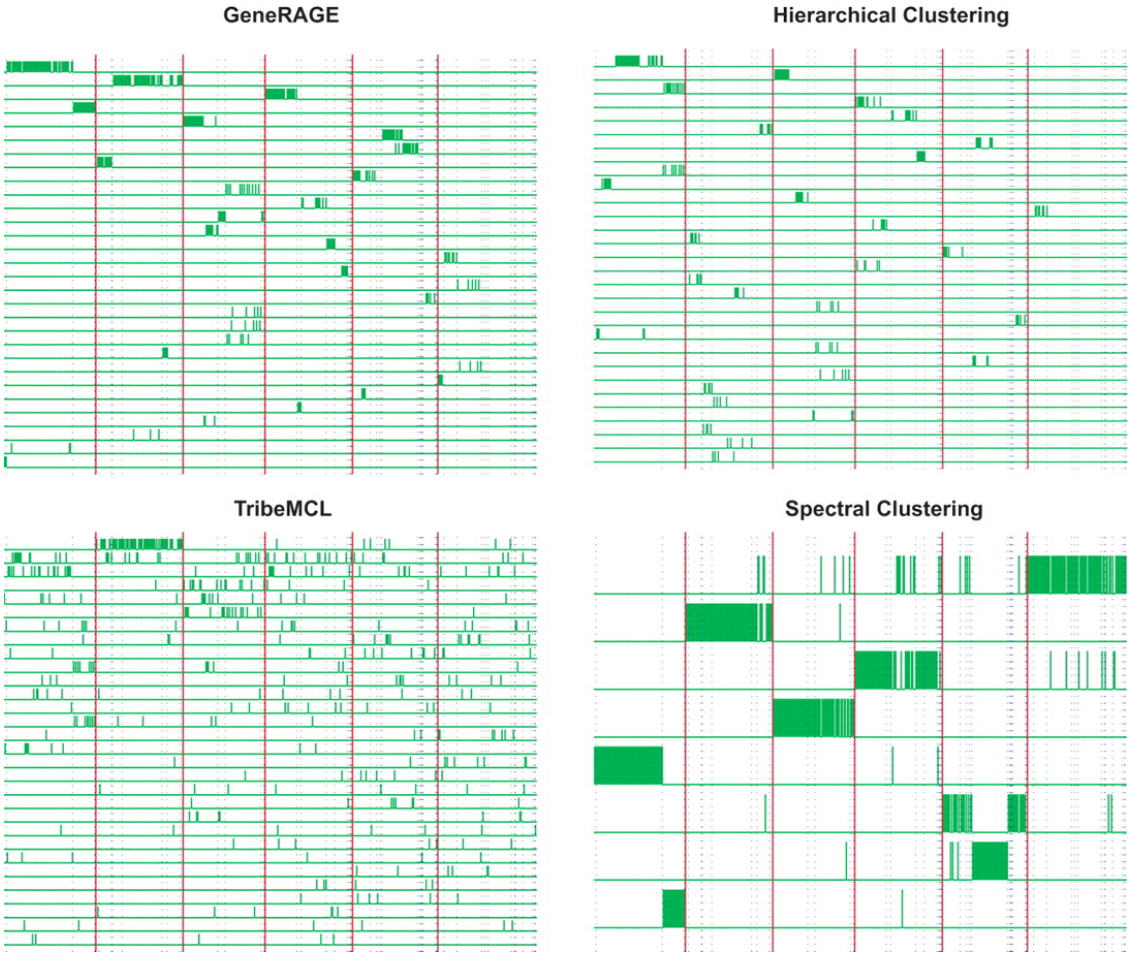


Figure 3.20: Graphical summary of the obtained clustering results of GeneRage, Hierarchical Clustering, TribeMCL, and Spectral Clustering for dataset ASTRAL95_1_161. Taken from [101].

Table 3.4: Evaluation of the WGCEP model. The best F-measures for each dataset and each similarity function. ASTRAL95_1_161 and ASTRAL95_2_161 are as in Table 1. ASTRAL95_1_171 and ASTRAL95_2_171 refer to the updated ASTRAL95 data of SCOP v1.71. BeH or SoH denote the similarity function, while the coverage factor f represents the influence of the coverage to the similarity.

Dataset	Similarity	Factor f	Threshold	F-measure
ASTRAL95_1_171	SoH	18	-3.0	0.91
ASTRAL95_1_171	BeH	15	-3.4	0.90
ASTRAL95_2_161	SoH	19	-1.6	0.89
ASTRAL95_2_171	SoH	15	-3.2	0.88
ASTRAL95_2_161	BeH	14	-2.4	0.87
ASTRAL95_2_171	BeH	13	-2.6	0.85
ASTRAL95_1_161	BeH	20	-2.2	0.85
ASTRAL95_1_161	SoH	20	-1.8	0.83

Evaluation of the heuristic

After evaluating the WGCEP as a reasonable clustering paradigm, we address the performance of the FORCE heuristic: We compare the running time and solution quality against a slow but exact algorithm on the large COG dataset. A recently developed fixed-

parameter (FP) algorithm for the WGCEP [108] extends ideas of previously developed FP algorithms for the (unweighted) GCEP by Gramm et al. [54, 55] and Dehne et al. [34], and has a running time of $O(3^k + |V|^3 \log |V|)$, if there exists a transitive projection of cost at most k. This allows us to find the optimal solution for a WGCEP, given a graph $G = (V, E)$ up to size $|V| \approx 50$ in appropriate time. To our knowledge, the implementation of this algorithm is the fastest available exact WGCEP solving program.

In order to compare the two approaches we use the COG dataset, split into connected subgraphs using similarity function SoH and a threshold of 10. We extracted 1 244 connected components (with $|V| \leq 3\,387$). For the evaluation, we restricted the maximal run time to 48 hours. The FP algorithm thus could only be applied to 825 components with $|V| \leq 56$. For the remaining components, the FP algorithm was terminated unsuccessfully after 48 hours. Due to the large number of graphs, we abstained from applying FP to graphs with $|V| \geq 100$, because it is very likely that the runtime would exceed 48 hours.

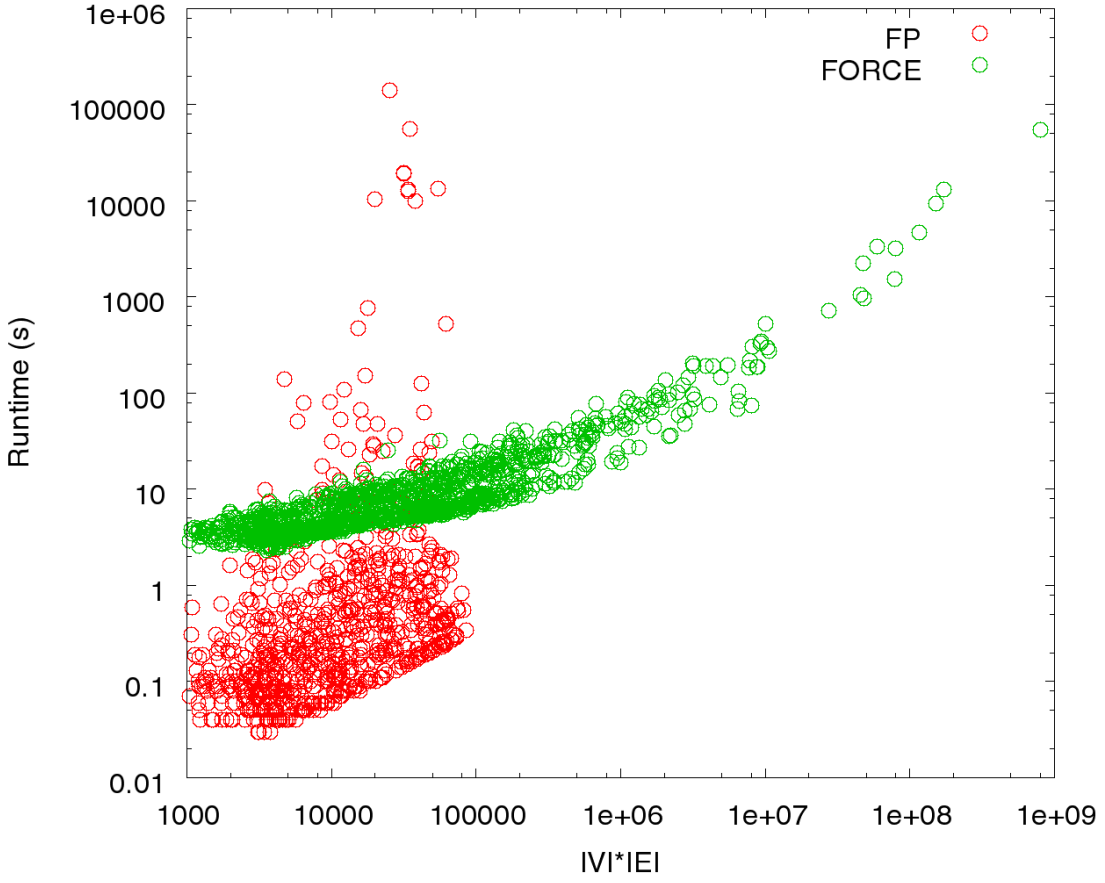


Figure 3.21: Comparison of the running times of FORCE against the exact fixed-parameter algorithm described in [108]. Plotted is the running time (y-axis in seconds) for different graph sizes (x-axis). Solely for visualization purposes, we describe the size of a graph on the x-axis as $|V| \cdot |E|$. All graphs have been constructed from procaryotic COG protein sequence comparisons using BeH as scoring function. Note that both axes are scaled logarithmically. The green points correspond to FORCE running times and the red points to the FP algorithm running times, respectively.

Figure 3.21 illustrates a running time comparison of the FP (red) and the heuristic algorithm (green). FORCE has been configured to use one generation of evolutionary parameter training for each graph. All time measurements were taken on a SunFire 880

with 900 MHz UltraSPARC III+ processors and 32 GB of RAM. One can see that for large graphs ($|V| \cdot |E| \geq 100\,000$), FORCE is much faster than the exact FP algorithm. Note that the axes are logarithmically scaled.

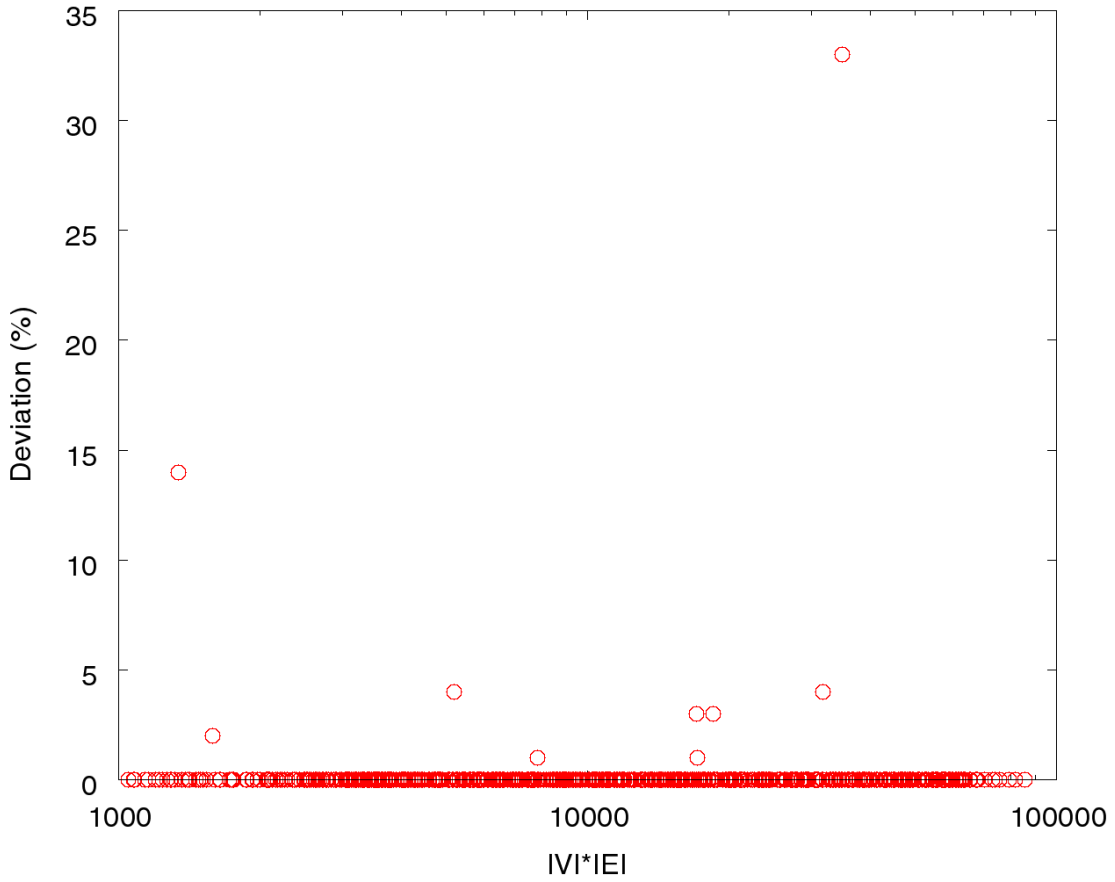


Figure 3.22: Relative cost deviations (y-axis in %) of the FORCE solutions from the optimal solutions found by the exact fixed-parameter algorithm described in [108]. The x-axis is as in Figure 3.21 (logarithmically scaled).

We evaluate the quality of the FORCE heuristic by comparing the relative cost increase of the reported solution, with respect to the provably optimal solution. For 814 out of the 825 comparable components, the heuristic determines the optimal solution. The optimal cost over all 825 components is 171,986.8, while FORCE finds a solution with a total cost of 172,244.6, which is a difference of 0.15%. Figure 3.22 illustrates these numbers. Note that most of the data points lie on the x-axis and hence indicate that the optimal solution was found.

In addition to the direct running time and quality comparison, we make all connected components and clustering results of the COG dataset available on the FORCE website, using the following similarity functions and thresholds: BeH/10, BeH/20, SoH/10, SoH/20. These choices do not reproduce the original COG clustering; we obtain the following F-measures: 0.64 (BeH/10), 0.56 (BeH/20), 0.61 (SoH/10), and 0.53 (SoH/20). It should be noted that (i) the COG clustering problem has very different properties than the SCOP clustering problem, and (ii) here we have not optimized in any way the scoring function and threshold.

3.6.3 Conclusions

We have shown that the WGCEP is an adequate model for remote protein homology clustering from sequence-based similarity measures and can outperform existing clustering approaches. Part of this effect is certainly attributable to the class of similarity functions that we consider. Nevertheless, in this particular application, the WGCEP paradigm (or rather our implementation) even outperforms the Affinity Propagation approach, for which we use the same class of similarity functions and a similar parameter optimization as for our approach.

We described FORCE, a heuristic algorithm for the NP-hard weighted graph cluster editing problem. Compared to the currently most efficient exact (exponential-time) fixed-parameter algorithm for this problem, we have empirically demonstrated that FORCE regularly provides solutions that are optimal, although no guarantee is given by the algorithm. In contrast to the exact algorithm, FORCE can solve the problem for graphs with several thousands of nodes in reasonable time.

We emphasize that FORCE can cluster any set of objects connected by any kind of similarity function using the concept of editing a graph into a transitive graph with minimum cost changes. The integrated evolutionary parameter training method ensures good performance on any kind of data.

Several issues remain to be resolved with the cluster editing or transitive projection approach. One disadvantage of the method is that it uses the same threshold for all clusters to determine the cost of adding or removing edges. The authors of SYSTERS [78] report an interesting approach to choose thresholds in a dynamical way. Finding a way of incorporating dynamic thresholds into cluster editing would certainly enhance its applicability.

The other issue we need to discuss is more global and applies to any clustering algorithm and concerns the choice of parameters. For evaluating the WGCEP model with the SCOP datasets, we have optimized similarity function and threshold by using the known truth as a reference and thus determined that there exists a (reasonably simple) similarity function that models the truth rather well. In practice, given an unknown dataset, we do not know which parameters lead to the unknown truth. Therefore we need to find properties of the resulting clustering (beyond the target function) that tell us something about the quality of the clustering. For CoryneRegNet, we were able to use the cluster size distribution, as we had expert biological support. In other cases, it is an open challenge to find properties of the clustering that can be easily verified by knowledgeable experts in the field.

3.7 Database content and development

CoryneRegNet was and still is subject to continuous improvement. Table 3.5 summarizes the development of the database content from the first release 1.0 to the current version 4.0. One can see that beside novel visualization and analysis features, also the amount of available data increased continuously.

At the Center for Biotechnology at Bielefeld University, we still perform experiments with corynebacteria and now also with *Mycobacterium tuberculosis*. CoryneRegNet is used to predict gene regulatory networks for mycobacteria mainly based on the knowledge from *C. glutamicum*. Therefor, the genomes of *Mycobacterium tuberculosis* CDC1551 and *My-*

Table 3.5: Database content development and growth of CoryneRegNet from the first release 1.0 to the current version 4.0. Abbreviations: Ver, CoryneRegNet version; Org, organisms; Genes, genes; TFs, transcription factors; Reg. genes, regulated genes; Regs, regulations; BM, binding motifs; PWM, position weight matrices; Stim, stimulons; Clust, protein clusters.

Ver.	Org	Genes	TFs	Reg. genes	Regs	BM	PWM	Stim	Clust
1.0	1	3,058	53	331	430	192	23	-	-
2.0	4	10,432	64	499	607	274	29	-	-
3.0	5	14,737	213	1,632	2,912	1,522	130	-	-
4.0	7	22,920	213	1,632	2,912	1,522	130	8	4,548

Table 3.6: This table briefly summarizes the stimulons that are integrated in CoryneRegNet.

Organism	Short description	Nr. of genes	Publication
C. glutamicum	ΔDtxR (*cg2103*) vs. wildtype	255	[22]
C. glutamicum	ΔLtbR (*cg1486*) vs. wildtype	50	[21]
C. glutamicum	ΔMcbR (*cg3253*) vs. wildtype	134	[111]
C. glutamicum	ΔSigM (*cg3420*) vs. wildtype	37	[95]
C. glutamicum	ΔSsuR (*cg0012*) vs. wildtype	29	[75]
C. glutamicum	Grown on acetate/propionate vs. acetate	160	[61]
C. glutamicum	res167 transition vs. res167 exponential	111	[80]
C. jeikeium	Wildtype vs. wildtype + vanillylalcohol	93	[19]

cobacterium tuberculosis H37Rv have been included into CoryneRegNet 4.0. Previously, with release 3.0, we integrated the complete genome annotation of the procaryotic model organism *E. coli* K-12 deposited in GenBank [127] and substantial data on transcriptional gene regulation provided by RegulonDB (refer to Section 2.1.1 on page 13). The database content is updated as soon as novel and experimentally verified data is available. Since we already did this continuously in the past, the number of regulations, regulators, binding motifs, etc. for the releases 3.0 and 4.0 do not differ in the last two rows of Table 3.5.

If a microarray experiment has been performed in wet lab, EMMA can be used for storing and analyzing the results. The Web Service client of CoryneRegNet can be used for the projection of gene expression levels to a visualized gene regulatory network to check for consistency with known regulatory pathways and to gain new insights. Beside the possibility to use unpublished, short-dated, and often transient expression data from EMMA, we additionally imported, verified, and published corynebacterial stimulon data directly into the CoryneRegNet database. A stimulon is a set of genes and we integrated that genes where the M-value $|m| > 1$. Table 3.6 summarizes the available experiments. Further microarray results can be included easily upon request.

As mentioned earlier in Section 3.1.2 (page 23), the back-end of CoryneRegNet is implemented by using an ontology-based data structure. It mainly consists of typed concepts and relations which are attached to values stored in a generalized data structure. In the following we describe the database content in terms of these vocabularies.

Table 3.7 summarizes the concept classes of CoryneRegNet. Given is the ID, a short description and the number of concepts that are of the corresponding type (concept class).

Table 3.7: Concept classes in CoryneRegNet.

ID	Short description	Nr. of concepts
CoryneRegNetModule	Functional module	12
ForceCluster	Protein cluster	4,548
Gene	Gene	22,920
Operon	Operon	2,945
Organism	Organism	7
Protein	Protein	22,189
SF	Sigma factor	2
Stimulon	Stimulon	8
TF	Transcription factor	606

Table 3.8: Relation types in CoryneRegNet.

ID	Short description	Nr. of relations
1*goop*	First gene of an operon	2,945
b_fc	Belongs to protein cluster	16,282
b_mod	Belongs to a functional module	557
b_op	Belongs to an operon	8,675
b_org	Belongs to an organism	48,682
en_by	Encoded by (gene)	22,796
ex_by	Expressed by (transcription/sigma factor)	1,480
ortho	Ortholog to (gene/protein)	231,763
para	Paralog to (gene/protein)	60,143
re_by	Repressed by (transcription factor)	1,432
stim_down	Downstimulated by (stimulon)	183
stim_up	Upstimulated by (stimulon)	686

For example, the back-end stores information on 4,548 concepts of type *ForceCluster* (protein cluster). Note that e.g. the class TF is a specialization of the class *Protein* (also see Section 3.1.2, page 23).

An outline of the ontological relations is presented in Table 3.8. Listed is the ID, the description, and the number of relations that are of the specific relation type. For example 8,675 concepts of the type *Gene* are connected to concepts of type *Operon* by using relations of type *b_op*. More informal: 8,675 genes are organized in 2,945 operons (refer to row four of Table 3.8).

Every concept is unambiguously defined by its ID and concept class. To simplify the attachment of attributes we use a generalized data structure and link every concept to an attribute of a certain type (*attribute_name*; also refer to Section 3.1.2). A summary of all attribute types is given in Table 3.9. Since, e.g. every gene has a start/stop position, and is located either at forward or backward strand we have stored 22,920 gene start/stop/strand values. The nucleotide content NCM is stored for every gene and every organism what leads to 22,798 attributes of type NCM. Note that we just calculated the nucleotide content for 'real', coding genes (as annotated in the NCBI database). CoryneRegNet furthermore computes the nucleotide content of coding/noncoding regions ($CNCM$, and $NCNCM$ respectively) for all of the seven organisms as background model for the binding

Table 3.9: Attribute types for ontological concepts.

ID	Short description	Nr. of concepts
CNCM	Nucleotide content of coding regions	7
COST	Codon start position	22,920
GECPL	Gene located at complementary strand?	22,920
GEEN	Gene end/stop position	22,920
GEST	Gene start position	22,920
NCM	Nucleotide content	22,798
NCNCM	Nucleotide content of noncoding regions	7
OPCPL	Operon located at complementary strand?	2,945
PWM	Position Weight Matrix	130
RT	Regulator Type	577

Table 3.10: Attribute types for ontological relations.

ID	Short description	Nr. of relations
BEV	Evidence for binding (experimental, predicted?)	2,912
BLEV	BLAST E-Value	291,906
BM	Binding motif	1,522
MK	Binding motif known?	2,912
MVAL	M-value (for a stimulated gene)	873
PMID	PubmedID (literature evidence for a regulation)	3,667

site prediction feature (see Section 3.3 on page 35). Also for that purpose the 130 available PWMs are stored in attributes of type *PWM*.

All attributes for relations are also typed (mainly gene regulations, stimulations, and homologies). An overview is presented in Table 3.10. For example 291,906 BLAST E-values are available for pairwise all-vs.-all gene/protein comparisons. In this case, a concept of class *Gene* is linked to another concept of class *Gene* by using a relation of type *ortho* (or *para*). To that relation, an attribute of type (*attribute_name*) *BLEV* is linked and the corresponding E-value is stored in the table *GDS_CONCEPT* (refer to the ER-diagram in Figure 3.2 on page 24). Another example would be one of the 873 M-values that are stored for relations of type *stim_up* (or *stim_down*) which in turn connect pairs of concepts of the concept classes *Gene* and *Stimulon*.

Here, one can see the power of the used ontology-based data structure. Such a generic back-end organization helps enormously, if a database project is started but if the nature of future data is unclear. When no triangular relations are necessary, the integration of any kind of data is possible. Even relations that link more than two concepts together are indirectly supported by creating an interjacent concept (of an intermediate concept class) and by linking all the concepts to the interjacent one. In the case of CoryneRegNet, from the very first beginning with release 1.0 (just one organism, no stimulons, no protein clusters, no PWMs etc.) up to release 4.0 (seven integrated species), we never modified the data structure of the back-end again.

4 Results and discussion

CoryneRegNet has been developed to facilitate the integrated analysis of corynebacterial gene regulatory networks. In Section 2.2 (page 16), we summarized the database content and the analysis features of related platforms and concluded with a requirement analysis. Now, we first briefly compare the database content of CoryneRegNet with that of the related systems. Subsequently, we describe how CoryneRegNet contributes to the required data analysis features.

Table 4.1: Comparison of the database content of related platforms for procaryotic gene regulatory networks (refer to Section 2.2) at organism level. Note that we exclude TRANSFAC and those organisms from consideration where only a few gene regulations are available.

Organism	RegulonDB	MtbRegList	PRODORIC	DBTBS	CoryneRegNet
Bacillus subtilis			+	+	
C. diphtheriae					+
C. efficiens					+
C. glutamicum					+
C. jeikeium					+
E. coli	+		+		+
M. tuberculosis CDC1551					+
M. tuberculosis H37Rv		+			+
Pseudomonas aeruginosa			+		

Table 4.1 summarizes the database content of CoryneRegNet and that of the related systems at organism level. Since TRANSFAC focuses on eucaryotes with no exception, we do not consider it here. Also, we exclude those organisms from consideration, where just the NCBI genome annotation and only a few gene regulatory interactions are available (this solely effects PRODORIC). RegulonDB focuses on *E. coli*. PRODORIC and CoryneRegNet also include corresponding data. The integrated *E. coli* data in both PRODORIC and CoryneRegNet was provided by RegulonDB and hence is congruent in all systems, if seen solely from the data content perspective. PRODORIC is the only data repository that covers data on *Pseudomonas aeruginosa*. The data on *Bacillus subtilis* in PRODORIC has mainly been extracted from DBTBS, which focuses on that organism. MtbRegList specializes on *Mycobacterium tuberculosis* H37Rv data, but does not include data on the strain CDC1551. CoryneRegNet is the only repository for corynebacterial species and therefore it is a reference database.

In the following, we summarize our contributions to each point mentioned in the requirement analysis in Section 2.2 (page 16).

- Genome browser: In the detailed view of a gene, the genomic context is visualized along with known sequence features: TFBMs, gene start/stop positions, and operon

organization (refer to Figure 3.4 on page 29). Most of the related platforms provide similar features (also refer to Figure 2.3 on page 18).

- Network visualization: CoryneRegNet incorporates a considerable network visualization by means of the Java applet GraphVis. It supports various graph layout styles and graph-based analysis features. GraphVis has been discussed in Section 3.2.2 on page 32. From the related systems only RegulonDB provides a network visualization (refer to Figure 2.4 on page 19) by using a fixed circular graph layout style.

- Raw data access: Principally, raw data access could be supported. Since CoryneRegNet provides Web Service based data access (see below), it is not necessary to support manual data download on the web site.

- BM prediction: The user has the possibility to start PWM-based TFBM predictions by using the integrated tool PoSSuMsearch. The advantages over other tools are (i) the on-the-fly calculation of p-values that indicate statistical significance for a putative BM and (ii) the very short response times. For a given TF, one can scan for BMs in the upstream sequences of other genes. For a given gene, the user can scan for BMs of all TFs in the database of a certain organism. The user interface is depicted in Figure 3.6 on page 31, while PoSSuMsearch and its integration is described in Section 3.3 on page 35. To further improve the PWM-based classification performance of putative TFBMs, we developed MoRAine (refer to Section 3.4 on page 37). In comparison to others, our approach of the joint usage of MoRAine and PoSSuMsearch outperforms the tools that are integrated in the related platforms in speed and accuracy. In the application case in Section 4.1.2 on page 72 we demonstrate the power of our approach. How MoRAine improves the TFBM prediction performance is evaluated in detail in Section 3.4.2 on page 40.

- Data exchange methods: By using SOAP-based Web Services, data on further gene annotations of a gene of interest is queried from GenDB automatically. The user does not even recognize that the data is downloaded from another service. Additionally, CoryneRegNet provides a client to the EMMA system, which enables the user to retrieve gene expression data for further analyses. Aside from this, CoryneRegNet offers a Web Service server to share the database content with others in a well-structured way. The corresponding Web Service description is provided as a WSDL-file from the CoryneRegNet or the CoryneCenter web site. A detailed description of CoryneRegNet's data exchange methods is given in Section 3.1.3 on page 25. Not one of the related platforms provides such an important functionality for integrated systems biology analyses.

- Network analysis: As mentioned earlier, bacterial gene regulatory networks normally show a hierarchical structure (clearly seen e.g. in Figure 4.2 on page 71) that is reasonably conserved between closely related species. The GraphVis feature of CoryneRegNet provides network comparison capabilities for both known and predicted networks. Aside from the BM prediction functionality, this assists scientists with cross-species knowledge transfer and hence with the identification of novel promising targets for wet lab analyses. Furthermore, it is possible to project gene

expression levels onto graphs, which helps to gain a first overview of experimental data. Beside these very special features, various graph layouting methods are available that e.g. clearly depict the hierarchical network structure. The main network analysis features are described in Section 3.2.2 on page 32 and have been applied in the Sections 4.1.3 and 4.1.4 (pages 75 and 77). Again, there is no equivalent data analysis functionality offered by other related systems.

- Homology detection: With FORCE we developed and integrated a powerful protein sequence clustering tool that helps to identify protein families across several species. It is based on weighted graph cluster editing, provides good results in practice, and can be applied on a large scale. FORCE is discussed in Section 3.6 on page 48. It outperforms the most popular protein homology detection tools, as we demonstrated in Section 3.6.2 on page 55. None of the related platforms has appropriate homology information directly included in the database.

- Contradictions/inconsistencies in gene expression experiments: For a given microarray experiment, one can scan the experimental results for contradictions in the relative gene expression levels concerning (i) operons and (ii) the known regulatory network by using the COMA feature. It is listed (i) whether all genes within an operon are regulated in a rectified way (all up or all down) and (ii) if an experiment hints for unknown transcriptional interactions due to contradictions to the stored networks (refer to Figure 3.15 on page 46). The COMA feature is described in detail in Section 3.5 on page 45. Again, such an integrated analysis of novel experimental data in the context of proven knowledge is not supported by related platforms.

With CoryneRegNet, we presented a comprehensive data analysis platform for corynebacterial gene regulatory networks that fulfills all the developed requirements (refer to Section 2.2). Aside from these special aims, its web-based user interface supports the standard tasks that are also offered by all other related systems: (i) browsing by navigating through the database entries and (ii) searching by identifying entries based on restrictions on the values of data fields within the database.

In a future release of CoryneRegNet, it could be beneficial to integrate a motif discovery tool. In this case, the user could upload a microarray result and it would be possible for CoryneRegNet to extract the upstream regions of differentially expressed genes. With the stored gene regulatory networks at hand, the user has a powerful toolkit for the prediction of novel TFBMs. Tompa et al. analyzed a variety of software that is publicly available for this purpose [81, 126].

The data stored in the back-end of CoryneRegNet is manually curated and mainly extracted from scientific literature. This work is very time-consuming and error-prone. An intelligent text mining component would ease this work, which is the most important prerequisite for a structured data management.

The next logical step would be to extend CoryneRegNet to a reference database for all procaryotic organisms by integrating more data into the back-end. This data would be available to the whole scientific community via the Web Service server and could be used for further external analyses by other software developers. To handle the huge amount of data, one would need (i) more compute power or (ii) a method to scatter the data over

several databases and to combine them on demand. The probably most beneficial strategy for such a federated database system would be to distribute all data separated by genus (one server for acidobacteria, one for actinobacteria, etc.).

With CoryneRegNet, we developed a platform, which is specialized for gene regulatory interactions of procaryotes. Although any kind of biological data can be stored in the ontology-based back-end in general, the front-end (especially the graph visualization tool GraphVis) is designed for transcriptional regulatory interactions. The integration of e.g. protein-protein interactions into CoryneRegNet's database would be simple, but the web interface as well as GraphVis would need to be extended in order to visualize the corresponding data. Generally, the more levels of abstraction used for such a system, the more data can be integrated and analyzed. However, one has to pay for every novel level with (i) runtime and (ii) usability. Hence, when respecting the aims and objectives of CoryneRegNet it is more beneficial to provide a special purpose tool rather than an all-rounder that is user-unfriendly in practice. With the SOAP-based Web Service server, CoryneRegNet offers a well-suited method to query the data for further integrated analyses, as we demonstrated with the CoryneCenter platform (refer to the application example in Section 4.1.4 on page 77).

A disadvantage of the integrated Web Service access to external data sources is the dependency on the online reachability of the connected components. For CoryneRegNet, this means to have no additional microarray results or no genome annotation data available when one of the systems, EMMA or GenDB, is not accessible. To avoid this risk, one would have to store external data in the local repository in regular time intervalls using import programs. A compromise would be to use both techniques in parallel: periodic downloads from the external data but usage of the Web Services when possible. More memory-efficient but less extensive would be to query the external data by using the Web Service on demand and to store just the queried data locally afterwards. Here the problem is to recognize modifications in the external repository.

The back-end's design as a data warehouse has mainly two consequences: (i) The user has no possibility to enter own data. (ii) The import process is time-consuming and has to be performed again whenever new data is available. The first point is desired since only curated (published) data is to be stored in the database persistently. But we address this point by providing the possibility to upload own gene expression data, binding motifs, or upstream sequences temporarily for an integrated analysis with the data stored in CoryneRegNet. We contribute to the second point by using incremental procedures for time-consuming import computations where possible. An import operation is only started if necessary, in order to speed-up the data warehousing process. For example BLAST as well as FORCE calculations are performed only when a novel organism has to be integrated.

Aside these running time considerations, the back-end design as data warehouse has practical disadvantages for the database curators at the moment. Although we simplified the import procedure to the execution of three programs, which parse the raw data flat files, this process is complex and has to be supported by technical staff. In a future release, we will provide a special web interface that allows the curators to modify the database content and to execute the import scripts directly from the user interface. Ideally, the above mentioned text mining component would be directly integrated into such a front-end. RegulonDB provides such a feature for database curators.

The usage of a Java Applet for network visualizations (GraphVis) allows the implementation of various graph layout and graph analysis functionalities. At the same time this strategy causes disadvantages. Due to security restrictions for Java Applets, GraphVis has no access to external data sources. Instead of querying the database server (MySQL) directly, we implemented a workaround via the Apache server. GraphVis sends all requests to a PHP script that in turn queries the database and provides the results (refer to Section 3.2.2 on page 32). The same problem occurs for the Applet's Web Service connections to GenDB and EMMA. We use the same workaround via the Apache. This is both intricate and time-consuming. Until now we found no practical way to solve this problem. The only possibility would be to disable the security restrictions at end-user side but this is not reasonable.

To speed-up convergence and improve the results, the embedded protein sequence clustering software FORCE should be extended to layout the cluster graphs not just in two but n dimensions. Furthermore, FORCE arranges all nodes in a circle as the first step of the layouting process (refer to Section 3.6 on page 48). One can attack this problem, e.g. by using Ant Colony Clustering (ACC) for an initial node placement. ACC could be used for a rough pre-clustering of the nodes and the force-based layout can be used for fine-tuning afterwards. The other way round (first FORCE and ACC for fine-tuning) is also possible. Since FORCE can cluster any kind of objects as long as an appropriate similarity function is available, one could also think about other areas of application; e.g. metabolic profiling of human breath (we recently discussed a corresponding platform in [13]).

4.1 Application cases

In the following sections, we exemplarily show the applicability of CoryneRegNet. We illustrate how it helps to address typical questions that biotechnological researchers ask for by means of four use cases, which can not be addressed directly using other existing platforms.

4.1.1 Reconstruction of the SOS and stress response module of *C. glutamicum*

We used CoryneRegNet to reconstruct and visualize the transcriptional regulatory network of the SOS and stress response module of *C. glutamicum* (see Figures 4.1 and 4.2). The module currently includes six DNA-binding transcription factors and 42 regulated genes. Since sigma factors play a key role in regulating gene expression when the cell is exposed to stress conditions and switches in part to the program "maintenance and survival" [98, 99], the regulatory network is apparently linked to components of the sigma factor competition module. Thus, the reconstructed network reveals a hierarchical scheme also including the top level regulator ppGpp, synthesized by the Rel protein and influencing expression of the sigma factors SigH and SigB [63, 99]. The reconstructed network allowed us to characterize the transcription factor module "SOS and stress response" in more detail: Several genes are under dual control by a DNA-binding transcription factor and by the alternative sigma factor SigH, whereas the *groEL2* gene is co-regulated by two transcription factors. The network is additionally characterized by a number of autoregulatory loops (Figure 4.1) in

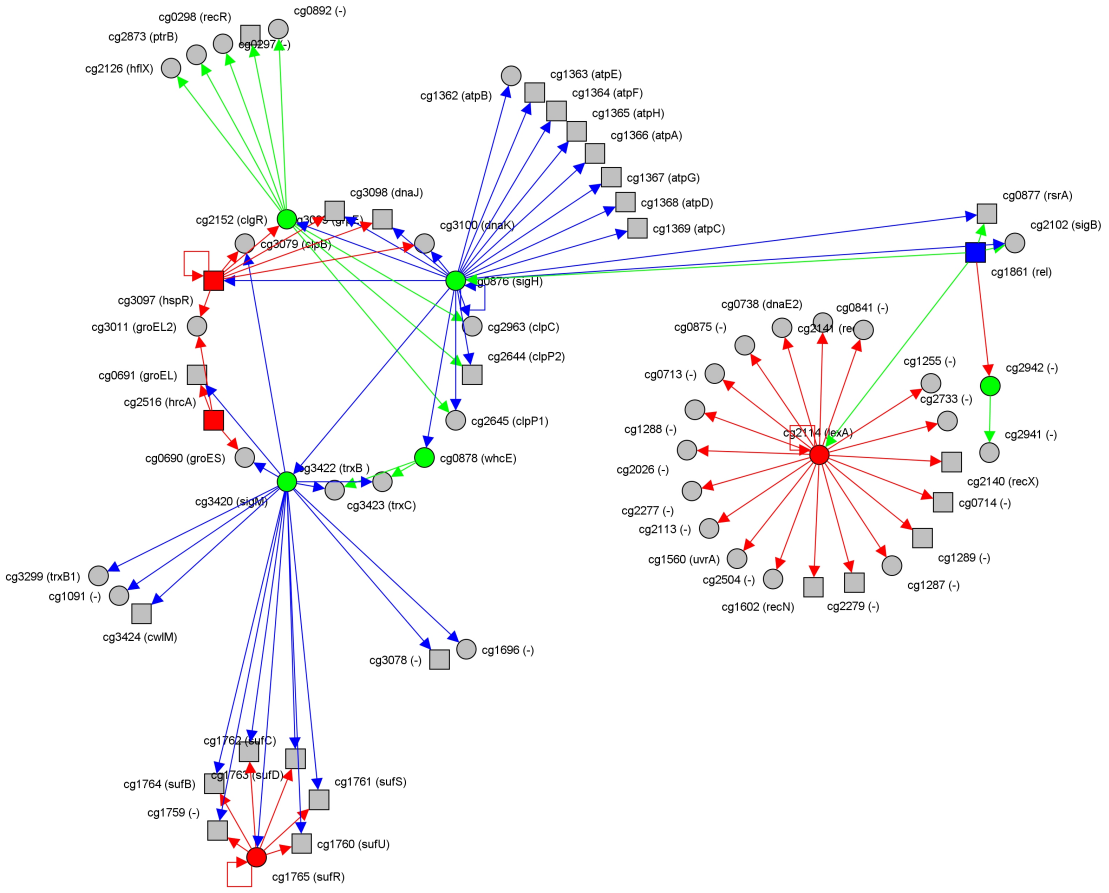


Figure 4.1: This screenshot of GraphVis shows the reconstruction of the SOS and stress response module of *C. glutamicum*. The graph was generated by using the compact circular layout mode. Nodes represent genes included in this functional module. Color code: red node and line, repressor and repressing regulatory interaction; green node and line, activator and activating regulatory interaction; green node and blue line, sigma factor and sigma factor interaction; blue node, dual regulator; gray node, regulated target gene preceded by a transcription factor binding site; gray box, regulated target gene that is part of an operon and not preceded by a transcription factor binding site.

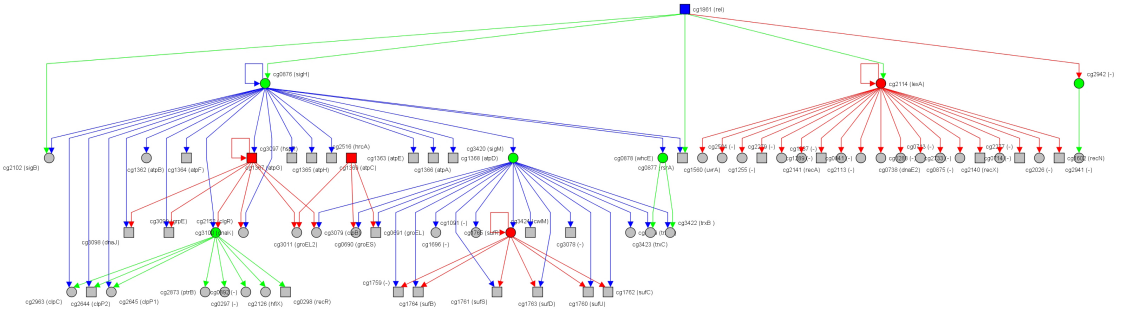


Figure 4.2: This screenshot of GraphVis shows the reconstruction of the SOS and stress response module of *C. glutamicum* using the hierarchical layout mode. The top level regulation of gene expression is indicated by the blue node of the gene *rel* that is responsible for the cellular amount of ppGpp. Color code: refer to the legend of Figure 4.1.

which the transcription factor regulates its own expression. Regarding regulatory network motifs, the presence of feed-forward loops is apparent when considering the regulatory action on gene expression of both a transcriptional regulator (HspR or ClgR) and an alternative sigma factor (SigH). This is consistent with observations in *E. coli* that feed-forward loop motifs tend to be implemented within modules, whereas bi-fan motifs seem to be responsible for the connection between different physiological modules [110]. Two types of feed-forward loops are present in the reconstructed network of the SOS and stress response module, namely the coherent type 1 and the incoherent type 1 motif [86]. In a coherent type 1 feed-forward loop all the regulatory connections are activating (SigH, ClgR, ClpP1-ClpP2), while in the incoherent type 1 motif one of the regulatory links represses the activity of the target node (SigH, HspR, DnaK). It is also apparent that the reconstructed regulatory network is composed of two distinct submodules reflecting different responses of the cell upon exposure to environmental stresses (Figure 4.2). The SOS response is induced by DNA damage and under control of the LexA protein, while the heat-shock and oxidative stress response is induced by denaturation and/or inactivation of proteins and is under SigH control [87]. Accordingly, the reconstruction and visualization of the SOS and stress response module of *C. glutamicum* by CoryneRegNet reflects the hierarchical and modular scheme of the cell's transcriptional regulatory system.

4.1.2 Transfer of the global regulatory network of DtxR from *C. glutamicum* to *C. diphtheriae*

First comparative studies revealed a high-level conservation of orthologous genes for corynebacteria [94] that may also be reflected in the structure of their transcriptional regulatory networks. The bioinformatics identification of the total sets of DNA-binding transcription factors was an initial step in defining the regulatory machinery of these bacteria and revealed different quantities of transcription factors depending on the habitat of the organism and its genome size [20]. One transcription factor conserved in all four species is DtxR, the diphtheria toxin repressor of *C. diphtheriae*, which has been subject to several genetic studies over the last years. Recently, the orthologous protein of *C. glutamicum* has been characterized on transcriptional level using DNA microarray technology [22]. Supported by bioinformatics analysis of the genome sequence of *C. glutamicum* a 19-bp palindromic sequence was identified in the upstream region of differentially expressed genes and was verified by DNA band shift assays *in vitro*. By this means the DtxR protein of *C. glutamicum* is directly activating or repressing the transcription of at least 64 genes with function in iron transport and utilization as well as in central carbohydrate metabolism and in transcriptional regulation [22]. In this study we report on the bioinformatics prediction of the regulatory network of DtxR of *C. diphtheriae* with CoryneRegNet. We use experimental data on DtxR binding sites of *C. glutamicum* to calculate a PWM. This matrix is applied to search for possible DtxR target sites in the non-coding regions of the genome sequence of *C. diphtheriae* with the $TFBScan$ option. Here we do not consider reverse or complementary hits although this would be possible. Since we already know part of the *C. diphtheriae* network, we can evaluate the promise of such an approach.

We vary the p-value cut-off between 10^{-7} (extremely strict) to 10^{-4} (relatively loose); Figure 4.3 shows the search results for 10^{-5}. Using the methods from [107] we predict

Table 4.2: For different p-value cut-offs, the table shows the expected number of hits due to chance (E-value) based on the effective search space size of 1.3 MB, the expected coverage of real binding sites (E(Coverage), an optimistic estimate based on the assumption that the PWM is the correct model), the number of detected hits at this threshold (Hits), the coverage of known true binding sites (Coverage), and the remaining number of hits to investigate (To study). That number should be compared to the E-value.

p-value	E-value	E(Coverage)	Hits	Coverage	To Study
10^{-7}	0.13	26%	3	$3/32 = 9.4\%$	0
10^{-6}	1.3	48%	8	$7/32 = 21.9\%$	1
10^{-5}	13	73%	22	$9/32 = 28.1\%$	13
10^{-4}	130	90%	83	$24/32 = 75\%$	59

Results

Target gene ID	Target gene name	Predicted operon	Rev./Compl.	pValue	eValue	Score	Sequence	Position	Candidates for homologous proteins to proteins regulated by cg2103 in validated original list
DIP0222	tox			1.1E-08	2.7E-02	1.447000e+03	TTAGGATAGCTTTACCTAA	-49..-31	
DIP1520	chtA	OP_dip1520		4.8E-08	1.2E-01	1.352000e+03	ATAGGTTAGGTTAACCTTG	-85..-67	
DIP1520	chtA	OP_dip1520		6.2E-08	1.5E-01	1.334000e+03	TTAGGTTAACCTTGCTTAA	-80..-62	
DIP0586	ciuE	OP_dip0586		9.6E-08	2.4E-01	1.304000e+03	TTAGGGTAGCTTCGCCTAA	-18..0	
DIP0370	-	OP_dip0370		1.6E-07	4E-01	1.267000e+03	TTAGGTCAGGGTACCCTAA	-127..-109	YP_224670.1, cg0445 (sdhCD), eValue: 2.5E-95
DIP0625	htaA	OP_dip0625		2.5E-07	6.1E-01	1.236000e+03	TTAGGTAAGTGTAGCCTAT	-169..-151	YP_224689.1, cg0466 (-), eValue: 4.1E-53 YP_224693.1, cg0470 (-), eValue: 8.7E-11 YP_224694.1, cg0471 (-), eValue: 7.2E-11
DIP2329	-			6.1E-07	1.5E+00	1.166000e+03	TGAGGTTAGGCATGCCTAT	-312..-294	
DIP2161	sidA	OP_dip2161		6.8E-07	1.7E+00	1.158000e+03	TTAGGGTAGGCTAATCCAA	-238..-220	
DIP0417	-			1.2E-06	2.9E+00	1.115000e+03	ATAGGCAAGGTTAAGCTAA	-213..-195	
DIP0369	-			1.5E-06	3.8E+00	1.092000e+03	TTAGGGTACCCTGACCTAA	-339..-321	
DIP0699	secA			1.9E-06	4.7E+00	1.074000e+03	TTGTGTTAGCCTAGGCTAA	-127..-109	
DIP1669	hmuO			2.5E-06	6.3E+00	1.049000e+03	TGAGGGGAACCTAACCTAA	-79..-61	YP_226469.1, cg2445 (hmuO), eValue: 9.1E-50
DIP2162	-			3.1E-06	7.7E+00	1.032000e+03	TTGGATTAGCCTACCCTAA	-165..-147	
DIP0415	-			3.2E-06	8E+00	1.028000e+03	TTAGCTTAACCTTGCCTAT	-29..-11	YP_224747.1, cg0527 (-), eValue: 2.4E-14
DIP0658	pccB1			3.9E-06	9.6E+00	1.012000e+03	TTTGGTAAACCTACCCTTT	-393..-375	
DIP1296	-	OP_dip1296		4.4E-06	1.1E+01	1.001000e+03	TTAGGGTGGGCTAACCTGC	-172..-154	
DIP0755	-			4.5E-06	1.1E+01	9.990000e+02	TTATGATTGCGTAGCCTAT	-505..-487	YP_225105.1, cg0928 (-), eValue: 1.1E-15
DIP1866	-			5.1E-06	1.3E+01	9.870000e+02	TTATGCTGGGCTATCTTAA	-54..-36	YP_226767.1, cg2782 (ftn), eValue: 6.3E-62
DIP1366	-			6E-06	1.5E+01	9.720000e+02	TTCGGTTGGGATAGCCTTG	-103..-85	
DIP0539	-			7.1E-06	1.8E+01	9.570000e+02	TTAGGCACCCCTAACCTAG	-240..-222	YP_224692.1, cg0469 (-), eValue: 1.9E-13 YP_224798.1, cg0589 (-), eValue: 1.6E-17 YP_224957.1, cg0768 (-), eValue: 3.1E-19 YP_225105.1, cg0928 (-), eValue: 1.3E-21
DIP1865	nrdF1			8.6E-06	2.1E+01	9.390000e+02	TTAAGATAGCCCAGCATAA	-274..-256	
DIP2330	-			1E-05	2.5E+01	9.240000e+02	ATAGGCATGCCTAACCTCA	-75..-57	
DIP0624	htaC			1E-05	2.5E+01	9.240000e+02	ATAGGCTACACTTACCTAA	-59..-41	YP_224688.1, cg0465 (-), eValue: 3.9E-12

Figure 4.3: Results of a search of transcription factor binding sites with the integrated *TFBScan* tool.

the number of hits due to chance and the fraction of identified true binding sites at this threshold, assuming that the PWM is an accurate model. The results are shown in Table 4.2. Choosing a p-value cut-off of 10^{-6} (E-value of 1.3), we find almost a quarter of the known regulated genes and are left with one hit for further study, which the E-value predicts to be due to chance. The statistics suggest that we should find almost half of the true regulated genes, but this prediction is based on the assumption that the true binding sites are independent samples from the frequency matrix, which is not true for two reasons: First and foremost, we are using the *C. glutamicum* PWM for predictions of binding sites in *C. diphtheriae*, where the binding motif is different. This is the price we pay for moving from one organism to a different one. Second, true binding sites do not behave according to a simple probabilistic model, and therefore the expected coverage prediction can only be true up to an order of magnitude. Taking these caveats into account, the integrated *TFBScan* of CoryneRegNet is a valuable tool to predict regulatory networks in taxonomically related microorganisms by using PWMs of experimentally defined regulons.

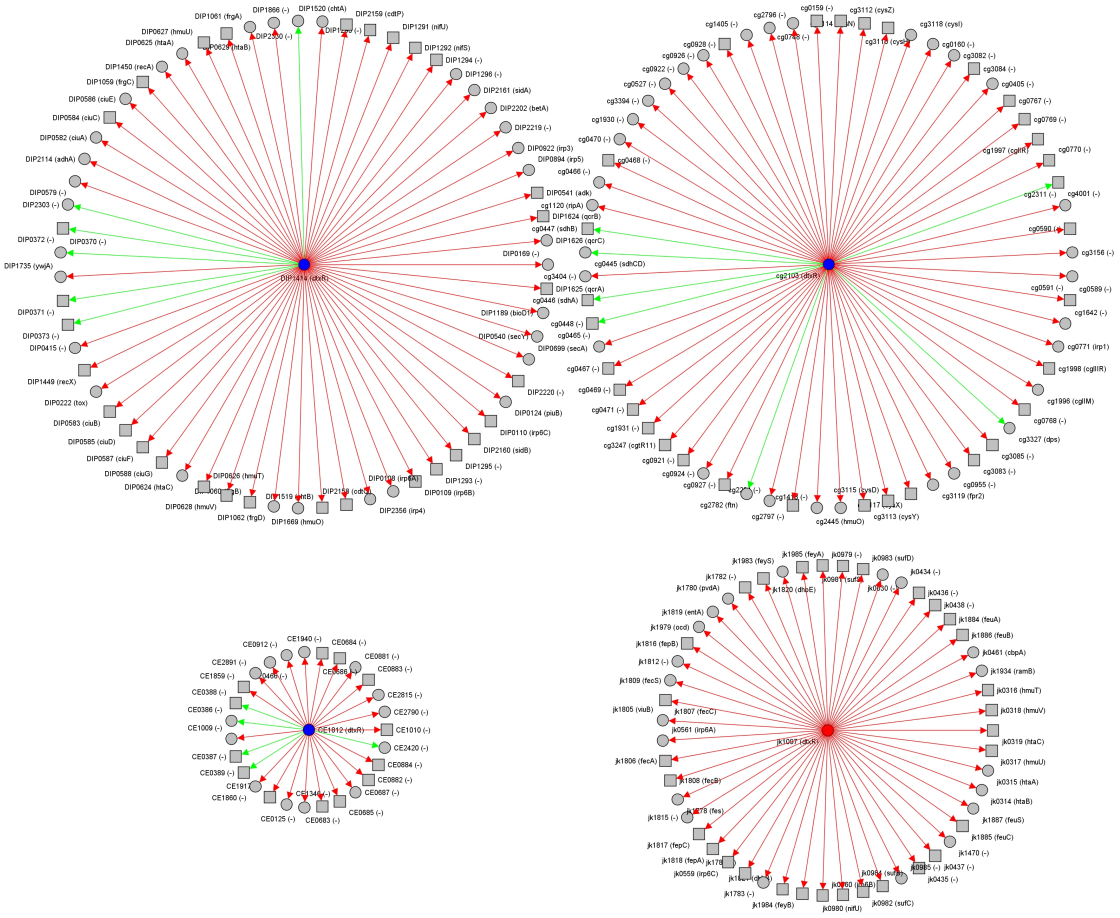


Figure 4.4: Comparative visualization of the DtxR regulons of *C. diphtheriae* (top left), *C. glutamicum* (top right), *C. jeikeium* (bottom right), and *C. efficiens* (bottom left). The graph was generated by using the circular layout style with a depth cut-off of 1. Color code: refer to legend of Figure 4.1

As a second feature of CoryneRegNet, we use the graph visualization tool GraphVis to display the DtxR regulons of all four corynebacterial genomes with a depth cut-off of 1. These regulatory networks are displayed in Figure 4.4. They are based on experimental data and bioinformatics predictions [22] stored in the database. DtxR is located in the center of each graph connected to the target genes by green or red arrows indicating activation or repression, respectively. Furthermore, one can see the differentiation between genes preceded by transcription factor binding sites (circles) and genes located in operons (squares). In order to extend the information content of the graph the depth cut-off can be varied for single nodes by using the 'extend graph' option in the GraphVis applet. In Figure 4.5 we extended the graph of the DtxR regulon of *C. glutamicum* for *cg1120*, coding for the transcription factor RipA, disclosing a regulatory sub-network of the DtxR regulon in *C. glutamicum*. The direct extension of the graph within the GraphVis Java applet enables the user to dynamically reconstruct and visualize the hierarchical structure of regulatory networks. These networks can further be compared based on homologies between the proteins. Figure 4.6 shows such a comparative layout. It is also possible to visualize a predicted network (by using the *TFBScan* feature; refer to Section 3.3, page 35) and to compare it with another one, stored in the database. This is an easy way to reveal potential targets for further wet lab experiments.

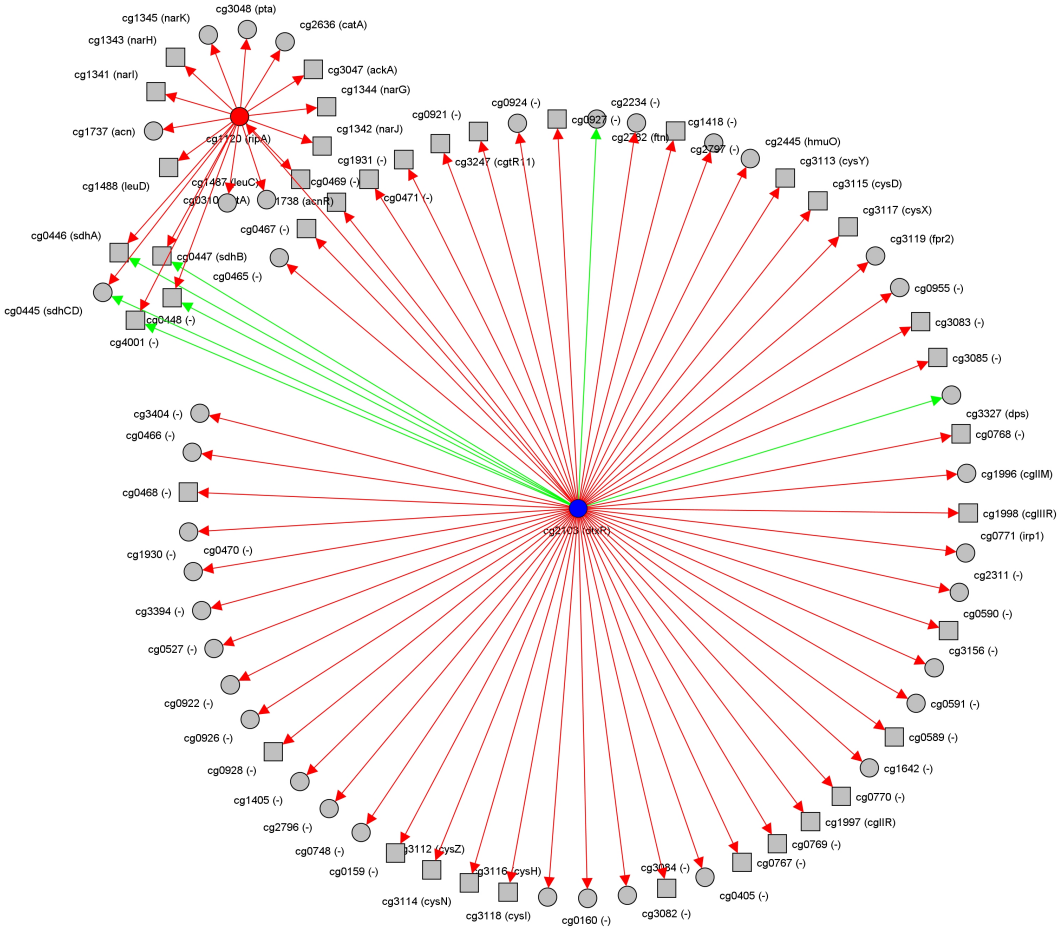


Figure 4.5: Dynamic extension of the regulatory network of DtxR of *C. glutamicum*. The graph is extended at the transcription factor RipA, which is encoded by *cg1120*. The sub-regulatory network of this repressor (red nodes and red lines) is visualized including genes co-regulated by DtxR (blue node) and *cg1120*. Color code: refer to the legend of Figure 4.1

4.1.3 Reconstruction and comparison of the LexA regulons in *C. glutamicum* and *E. coli*

Figure 4.8 illustrates the reconstruction and comparison of the LexA regulons from *C. glutamicum* and *E. coli* by means of the homology layouter of CoryneRegNet. Generally, bacteria respond to DNA damage by increasing the expression of a number of genes, resulting in DNA repair and an enhanced rate of survival. In many species, this transcriptional response is negatively regulated by the LexA protein that binds to a regulatory DNA sequence termed SOS box. The nucleotide sequence of the SOS box is strongly conserved among taxonomic closely related bacterial species, but the LexA recognition motifs are different in distantly related microorganisms [89]. This observation is apparent when comparing the sequence logos calculated for the SOS boxes of *C. glutamicum* and *E. coli* (Figure 4.7). In addition to differences in the DNA binding motif of LexA, also the gene content of the LexA regulon varies among bacterial species. For instance, *in silico* analysis and experimental studies in proteobacteria revealed that a LexA core regulon structure comprises, among others, the *lexA*, *recA*, *recN* and *uvrA* genes [41,42]. Comparison of the LexA regulons of *C. glutamicum* and *E. coli* with the homology layouter of CoryneRegNet also identifies *lexA*, *recA*, *recN* and *uvrA* as the small common set of LexA-regulated genes

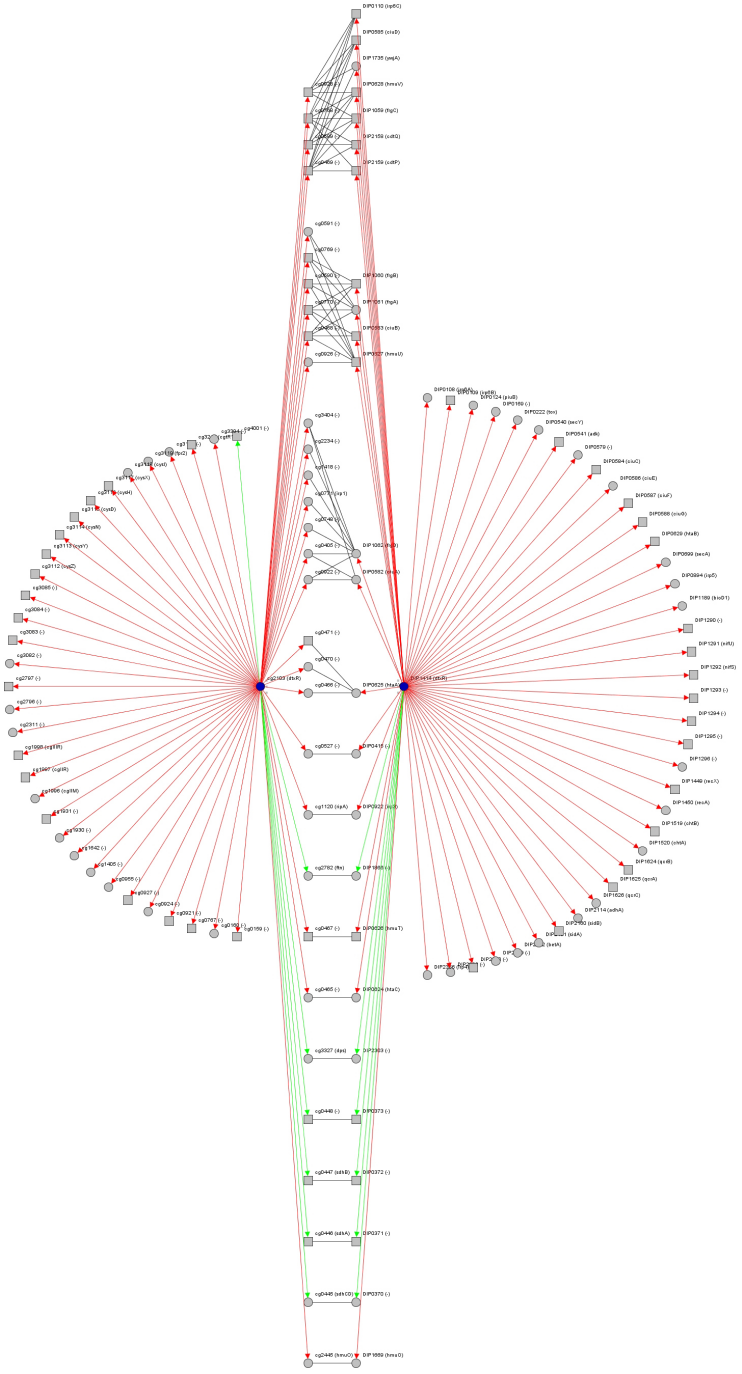


Figure 4.6: Visualization of the regulatory networks of DtxR of *C. glutamicum* (left side) and *C. diphtheriae* (right side) by using a comparative graph layout. The genes in the middle part are connected due to a positive sequence-based homology detection. Color code: refer to the legend of Figure 4.1

in these distantly related bacteria (refer to Figure 4.8). The *recX* gene located downstream of *recA* in both species was not classified into the common set of LexA-regulated genes due to the low level of amino acid sequence similarity that was below the E-value threshold of 10^{-10}. Consequently, homology-based network reconstruction and comparison may provide valuable insights into the gene composition, the genetic core and the evolution of transcriptional regulatory networks.

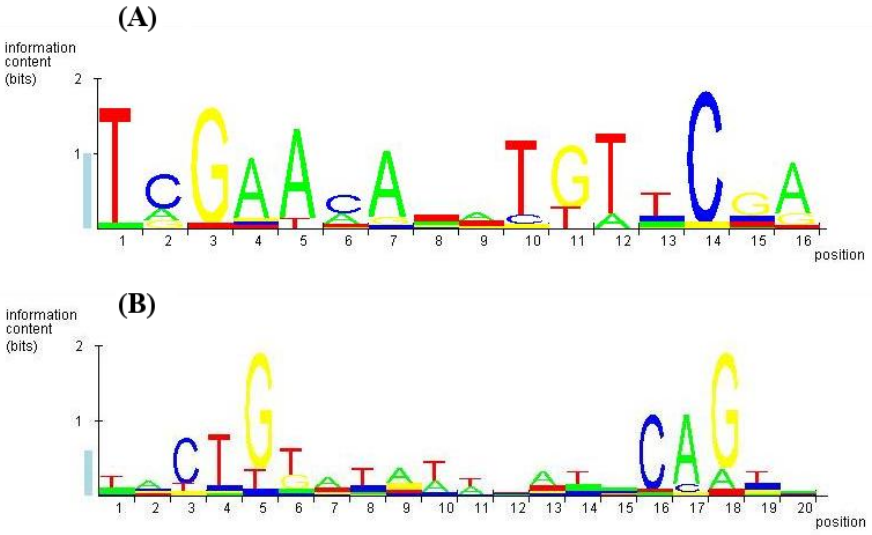


Figure 4.7: Sequence logos generated by CoryneRegNet. The graphics display sequence logos of the DNA binding site (SOS box) of the LexA protein from *C. glutamicum* (A) and *E. coli* (B). The height of the stacks indicates the information content at each position of the motif in bits. The blue bar to the left of the y-axis indicates the mean information content of the motif. According to the evolutionary history of the LexA-binding sequence, the DNA binding sites in *C. glutamicum* and *E. coli* are apparently different with respect to the motif size and sequence [89].

4.1.4 Dissection of the global transcriptional response in microarray data by comparing *C. glutamicum* grown on two different carbon sources

We use CoryneRegNet to display transcriptional differences in gene regulatory networks which were detected by microarray analysis of *C. glutamicum* grown with either glucose or acetate as sole source of carbon and energy. The metabolic utilization of glucose and acetate and the cellular adaptation to these different carbon sources are principally different and were subject of intensive investigations in the past (reviewed in [50]).

In *C. glutamicum*, sugars such as glucose are simultaneously taken up and activated by the phosphoenolpyruvate:phosphotransferase system (PTS). The resulting sugar phosphates are metabolized by the glycolysis pathway forming acetyl-Coenzyme A (acetyl-CoA), which then enters the tricarboxylic acid (TCA) cycle.

A different situation emerges when *C. glutamicum* is cultivated on acetate as sole carbon and energy source. Acetate has to be activated by acetate kinase and phosphate acyltransferase. The formed acetyl-CoA enters the TCA cycle. Besides the uptake and activation of acetate, the reactions of the glyoxylate shunt are necessary to replenish the tricarboxylic acid cycle. Theses reactions, catalyzed by isocitrate lyase (ICL) and malate synthase (MS), bypass the TCA cycle to avoid the oxidative decarboxylation steps and finally lead to the formation of the acceptor molecule oxaloacetate from two molecules acetyl-CoA. Oxaloacetate is needed for gluconeogenesis and to keep the TCA cycle running under these conditions [77]. In addition, the reverse operation of glycolysis (gluconeogenesis) is necessary to synthesize essential sugar phosphates.

The metabolic and regulatory switch between glucose and acetate consumption in *C. glu-*

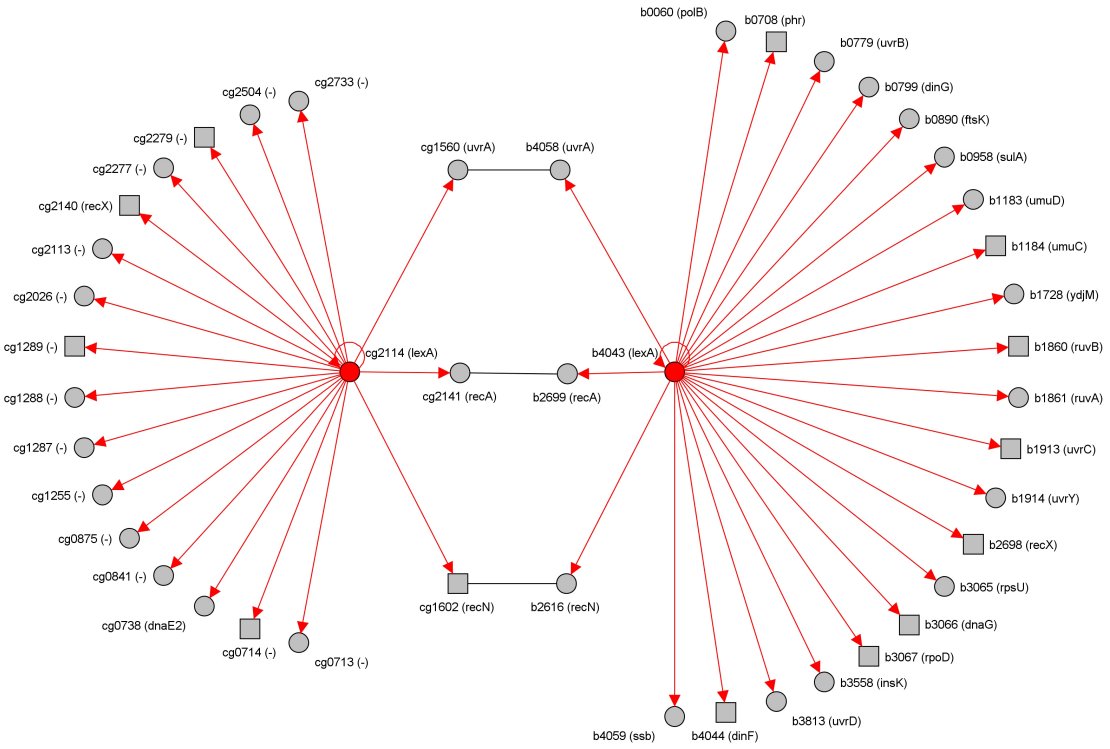


Figure 4.8: Comparative visualization of gene regulatory networks by using GraphVis. The LexA regulatory networks involved in the SOS response of *C. glutamicum* (left) and *E. coli* (right) were reconstructed, visualized and compared by including homology data on the proteins that are part of each regulon. Black lines represent the respective sequence similarities in the homology-based layout mode of GraphVis, using the homology layouter. Color code: refer to the legend of Figure 4.1

tamicum is well-studied. It has been shown that the glyoxylate shunt in *C. glutamicum* is mainly controlled by transcriptional regulation of the genes *aceA* and *aceB* coding for ICL and MS, respectively. Three different regulatory proteins that influence transcription by interacting with the upstream regions of *aceA* and *aceB* have so far been identified: The RamB protein acts as a negative transcriptional regulator on the two genes in presence of glucose [29], RamA as a positive transcriptional regulator in the presence of acetate [30], and GlxR as a negative regulator in the presence of cyclic AMP [73]. All three regulators do not only address *aceA* and *aceB* but act as global regulators with regulatory networks including several target genes. All of their known interactions, either experimentally determined from *in vitro* experiments or predicted, are stored in CoryneRegNet and can be used to dissect the complex data from microarray experiments.

In the this study, we analyze the transcriptional stimulon of *C. glutamicum* grown on acetate as sole carbon and energy source. The different influences of each of the three known regulators on their networks will be checked for consistency with the known or predicted regulatory interactions in CoryneRegNet under *in vivo* conditions. For this purpose we compare the transcriptome of acetate-grown cells to the transcriptome of glucose-grown cells, using microarray hybridization results stored in EMMA.

In CoryneRegNet, the data of the microarray experiment can easily be mapped onto regulatory networks. Figure 4.9 shows the gene regulatory networks of RamA, RamB and

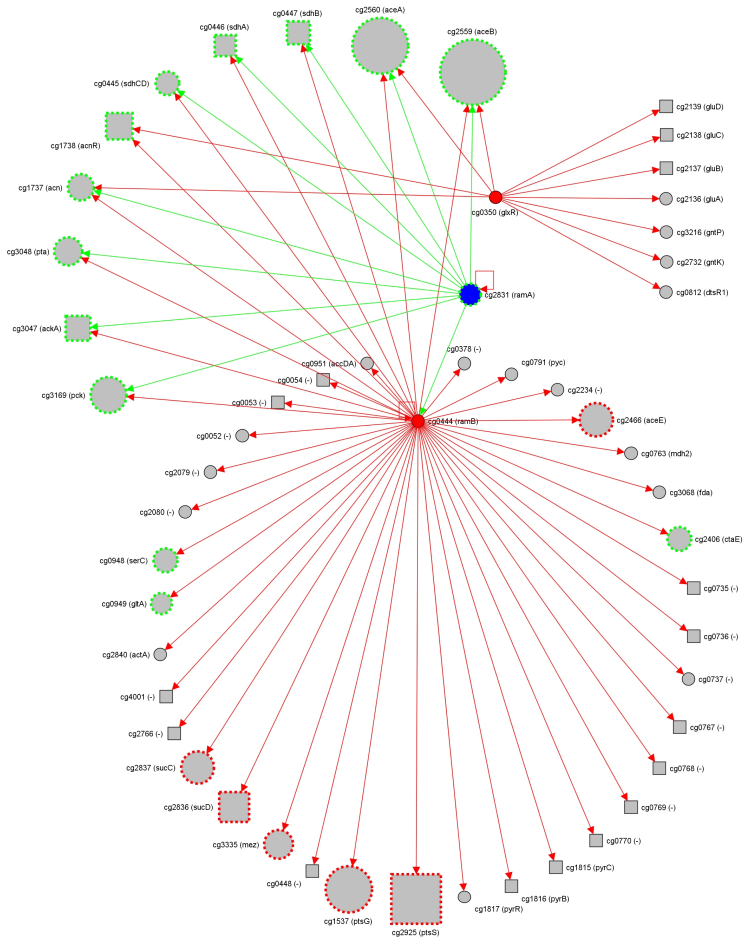


Figure 4.9: The screenshot shows the reconstruction of the gene regulatory networks of RamA, RamB and GlxR and a simultaneous visualization of relative transcript abundances obtained from comparative microarray analysis of *C. glutamicum* grown on either acetate or glucose as sole carbon source. The blue node represents the gene of the selected regulator (RamA). Red dashed node borders indicate a significantly reduced amount of transcript in the acetate grown culture compared to the glucose grown culture, while green dashed node borders mean a significantly enhanced amount of transcript in the acetate grown culture compared to the glucose grown culture. Black bordered nodes show insignificantly altered transcript levels. The size of the nodes is proportional to the relative differential gene expression measured in the microarray experiment (m-value). Color code: refer to the legend of Figure 4.1

GlxR indicating the relative transcript abundances of the genes encoding the regulatory proteins and of their target genes in acetate-grown *C. glutamicum* cells relative to those from glucose-grown cells. Nodes with green dashed borders indicate enhanced transcript levels, while nodes with red borders describe decreased levels during growth on acetate. The size of the nodes is proportional to the relative differential gene expression (m-value).

In this visualization, the RamA network shows a consistent answer to the stimulus. All target genes except *ramB* showed elevated transcript levels, which correlates to the enhanced transcription of the *ramA* gene. These observations confirm the results of Cramer and coworkers [29], who showed that RamA activates its target genes in the presence of acetate and that the negative auto-regulation of RamA has no influence under this condition. Interestingly, the transcription level of the RamA target gene *ramB* was unaffected

(or not detectable) in this experiment. This finding is in contrast to data from the RamB protein quantification by immunoblotting during growth of *C. glutamicum* on different carbon sources, where less RamB protein was found in acetate-grown cells than in glucose-grown cells [28]. In addition, inspection of the RamB target genes showed that most genes are not significantly detected as altered in their transcript levels, which is in accordance to the unchanged *ramB* transcript level. It seems that the regulatory activity of RamB is subdominant in this experiment.

However, the strongly decreased transcript levels of the RamB target genes *ptsS* and *ptsG* encoding sucrose and glucose transport proteins of the PTS system, respectively, could not be explained in this way and point to an additional regulatory network active under acetate or glucose feeding conditions. Recently, the regulator SugR was identified that represses transcription of PTS genes in the absence of sugar-phosphates in *C. glutamicum*. [38] Therefore the detected changes in the transcript level of *ptsG* and *ptsS* are most probably due to a repression by SugR (which is slightly overexpressed) when the cells are grown on acetate.

A regulatory effect of GlxR seems not to be dominant, because a consistent de-repression of its regulon was not detected. For the *glxR* gene itself, unchanged transcript levels were expected because the regulatory activity of the protein is thought to be due to an interaction with the second messenger cAMP. It is known that intracellular cAMP levels during growth on acetate are significantly lower than on glucose [73]. This implies that the genes of the GlxR regulon should show enhanced transcript levels. This effect was not consistently detected in the microarray analysis and might mean that the cAMP levels are not different enough to provoke a detectable response in the GlxR regulatory network.

5 Conclusion

Novel ultra-fast sequencing and large-scale post-genomic analysis techniques of complete genome sequences recently generated a vast amount of experimental data. With the impressive advances in global data generation by high-throughput technologies, systems biology has emerged to use genome-wide data and cell-wide measurements in elucidating the optimal design of new production strains by genome-scale modeling and simulation [1]. Accordingly, a major challenge in molecular biology is the development of suitable bioinformatics platforms for storage and evaluation of high-throughput data to make it feasible to perform large-scale modeling and simulation studies. An apparent requirement in this challenge is the ability to identify and reconstruct the global connectivity of transcriptional regulatory interactions in a bacterial cell [109]. This asks for user-oriented software platforms supporting (i) the integration of existing knowledge, (ii) visualization capabilities, (iii) the generation of novel hypotheses, and (iv) the possibility to share this post-processed data with others..

To address these tasks, several approaches have been implemented and established. Until now, there is no platform available that provides data on gene regulations for *C. glutamicum*, *C. diphthaeriae*, *C. efficiens*, and *C. jeikeium*. We analyzed five related systems, which are specialized for other organisms regarding their advantages and disadvantages. Not one of these platforms provide all the necessary methods to satisfactorily support the above mentioned data processing tasks.

To provide a comprehensive system for the integrated analysis of procaryotic gene regulatory networks we developed the online platform CoryneRegNet. The database contains information on DNA-binding transcription factors and on transcriptional regulatory interactions of corynebacteria, mycobacteria, and *E. coli*. Also the results of global DNA microarray hybridization experiments have been integrated as stimulons into the CoryneRegNet data repository. A web-based user interface provides access to the database content, allows various queries, and supports the reconstruction, visualization, validation, and prediction of regulatory networks at different hierarchical levels. CoryneRegNet is moreover linked to several databases (EMMA, GenDB, COG, GO, NCBI, etc.). Although CoryneRegNet initially was developed as a data warehouse of transcriptional regulatory networks of *C. glutamicum*, its ontology-based design along with its programs and scripts has been designed for a general applicability to other species. Hence, it has been extended with genomic and transcriptional data on six more organisms, experimental results (stimulons), and computer predictions (protein clusters, PWM-based binding motif predictions, etc.). CoryneRegNet is connected to other data sources using SOAP-based Web Services and it provides its own Web Service server.

CoryneRegNet is not just another system that is focused on corynebacteria. Generally, its database content can be extended with data on any other organism. It provides features that are not offered by other platforms: (i) graph visualization with different layout styles,

(ii) comparative network visualizations, (iii) statistically sound and fast TFBM prediction, (iv) reliable homology detection, (v) well-structured data exchange by using Web Services, and (vi) network analysis in the context of experimental results.

In contrast to related platforms, CoryneRegNet provides all capabilities that are necessary in modern systems biology: (i) data integration of existing knowledge, (ii) visualization, (iii) the generation of novel hypotheses, and (iv) data exchange methods.

Consequently, CoryneRegNet is a versatile systems biology platform to support the efficient and large-scale analysis of transcriptional regulation of gene expression in microorganisms.

Bibliography

[1] Adrio JL and Demain AL. Genetic improvement of processes yielding microbial products. *FEMS Microbiol Rev*, 30(2):187–214, Mar 2006.

[2] Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, and Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, 1997.

[3] Andreeva A, Howorth D, Brenner SE, Hubbard TJ, Chothia C, and Murzin AG. SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Research*, 32:D226–D229, 2004.

[4] Babu MM, Luscombe NM, Aravind L, Gerstein M, and Teichmann SA. Structure and evolution of transcriptional regulatory networks. *Curr Opin Struct Biol*, 14(3):283–291, Jun 2004.

[5] Babu MM and Teichmann SA. Evolution of transcription factors and the gene regulatory network in Escherichia coli. *Nucleic Acids Res*, 31(4):1234–1244, Feb 2003.

[6] Babu MM, Teichmann SA, and Aravind L. Evolutionary dynamics of prokaryotic transcriptional regulatory networks. *J Mol Biol*, 358(2):614–633, Apr 2006.

[7] Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, and Yeh LSL. The Universal Protein Resource (UniProt). *Nucleic Acids Res*, 33(Database issue):D154–D159, Jan 2005.

[8] Balaji S, Babu MM, and Aravind L. Interplay between network structures, regulatory modes and sensing mechanisms of transcription factors in the transcriptional regulatory network of E. coli. *J Mol Biol*, 372(4):1108–1122, Sep 2007.

[9] Barthelmes J, Ebeling C, Chang A, Schomburg I, and Schomburg D. BRENDA, AMENDA and FRENDA: the enzyme information system in 2007. *Nucleic Acids Res*, 35(Database issue):D511–D514, Jan 2007.

[10] Baumbach J. CoryneRegNet 4.0 - A reference database for corynebacterial gene regulatory networks. *BMC Bioinformatics*, 8(1):429, Nov 2007.

[11] Baumbach J, Brinkrolf K, Czaja L, Rahmann S, and Tauch A. CoryneRegNet: An ontology-based data warehouse of corynebacterial transcription factors and regulatory networks. *BMC Genomics*, 7(1):24, Feb 2006.

[12] Baumbach J, Brinkrolf K, Wittkop T, Tauch A, and Rahmann S. CoryneRegNet 2: An Integrative Bioinformatics Approach for Reconstruction and Comparison of

Transcriptional Regulatory Networks in Prokaryotes. *Journal of Integrative Bioinformatics*, 3(2):24, 2006.

[13] Baumbach J, Bunkowski A, Lange S, Oberwahrenbrock T, Kleinboelting N, Rahmann S, and Baumbach J. IMS2 - An integrated medical software system for early lung cancer detection using ion mobility spectrometry data of human breath. *Journal of Integrative Bioinformatics*, 4(3):75, 2007.

[14] Baumbach J, Wittkop T, Rademacher K, Rahmann S, Brinkrolf K, and Tauch A. CoryneRegNet 3.0-An interactive systems biology platform for the analysis of gene regulatory networks in corynebacteria and Escherichia coli. *J Biotechnol*, 129(2):279–289, Apr 2007.

[15] Baumbach J, Wittkop T, Weile J, Kohl T, and Rahmann S. MoRAine - A web server for fast computational transcription factor binding motif reannotation. (submitted), 2008.

[16] Beckstette M, Homann R, Giegerich R, and Kurtz S. Fast index based algorithms and software for matching position specific scoring matrices. *BMC Bioinformatics*, 7:389, 2006.

[17] Beckstette M, Strothmann D, Homann R, Giegerich R, and Kurtz S. PoSSuMsearch: Fast and sensitive matching of position specific scoring matrices using enhanced suffix arrays. *GI Lecture Notes in Informatics*, P-53:53–64, 2004.

[18] Bergman CM, Carlson JW, and Celniker SE. Drosophila DNase I footprint database: a systematic genome annotation of transcription factor binding sites in the fruitfly, Drosophila melanogaster. *Bioinformatics*, 21(8):1747–1749, Apr 2005.

[19] Brune I, Becker A, Paarmann D, Albersmeier A, Kalinowski J, Puehler A, and Tauch A. Under the influence of the active deodorant ingredient 4-hydroxy-3-methoxybenzyl alcohol, the skin bacterium Corynebacterium jeikeium moderately responds with differential gene expression. *J Biotechnol*, 127(1):21–33, Dec 2006.

[20] Brune I, Brinkrolf K, Kalinowski J, Pühler A, and Tauch A. The individual and common repertoire of DNA-binding transcriptional regulators of Corynebacterium glutamicum, Corynebacterium efficiens, Corynebacterium diphtheriae and Corynebacterium jeikeium deduced from the complete genome sequences. *BMC Genomics*, 6(1):86, 2005.

[21] Brune I, Jochmann N, Brinkrolf K, Hueser AT, Gerstmeir R, Eikmanns BJ, Kalinowski J, Puehler A, and Tauch A. The IclR-type transcriptional repressor LtbR regulates the expression of leucine and tryptophan biosynthesis genes in the amino acid producer Corynebacterium glutamicum. *J Bacteriol*, 189(7):2720–2733, Apr 2007.

[22] Brune I, Werner H, Huser A, Kalinowski J, Puhler A, and Tauch A. The DtxR protein acting as dual transcriptional regulator directs a global regulatory network involved in iron metabolism of Corynebacterium glutamicum. *BMC Genomics*, 7(1):21, Feb 2006.

[23] Camus JC, Pryor MJ, Médigue C, and Cole ST. Re-annotation of the genome sequence of Mycobacterium tuberculosis H37Rv. *Microbiology*, 148(Pt 10):2967–2973, Oct 2002.

[24] Cases I, deLorenzo V, and Ouzounis C. Transcriptional regulation and environmental adaptation in bacteria. *Trends Microbiol*, 11(6):248–253, 2003.

[25] Cerdeño-Tárraga AM, Efstratiou A, Dover LG, Holden MTG, Pallen M, Bentley SD, Besra GS, Churcher C, James KD, Zoysa AD, Chillingworth T, Cronin A, Dowd L, Feltwell T, Hamlin N, Holroyd S, Jagels K, Moule S, Quail MA, Rabbinowitsch E, Rutherford KM, Thomson NR, Unwin L, Whitehead S, Barrell BG, and Parkhill J. The complete genome sequence and analysis of Corynebacterium diphtheriae NCTC13129. *Nucleic Acids Res*, 31(22):6516–6523, Nov 2003.

[26] Chandonia JM, Hon G, Walker NS, Conte LL, Koehl P, Levitt M, and Brenner SE. The ASTRAL Compendium in 2004. *Nucleic Acids research*, 32:D189–D192, 2004.

[27] Chekmenev DS, Haid C, and Kel AE. P-Match: transcription factor binding site search by combining patterns and weight matrices. *Nucleic Acids Res*, 33(Web Server issue):W432–W437, Jul 2005.

[28] Cramer A, Auchter M, Frunzke J, Bott M, and Eikmanns BJ. RamB, the transcriptional regulator of acetate metabolism in Corynebacterium glutamicum, is subject to regulation by RamA and RamB. *J Bacteriol*, 189(3):1145–1149, Feb 2007.

[29] Cramer A and Eikmanns BJ. RamA, the transcriptional regulator of acetate metabolism in Corynebacterium glutamicum, is subject to negative autoregulation. *J Mol Microbiol Biotechnol*, 12(1-2):51–59, 2007.

[30] Cramer A, Gerstmeir R, Schaffer S, Bott M, and Eikmanns BJ. Identification of RamA, a novel LuxR-type transcriptional regulator of genes involved in acetate metabolism of Corynebacterium glutamicum. *J Bacteriol*, 188(7):2554–2567, Apr 2006.

[31] Crooks GE, Hon G, Chandonia JM, and Brenner SE. WebLogo: a sequence logo generator. *Genome Res*, 14(6):1188–1190, Jun 2004.

[32] Curbera F, Duftler M, Khalaf R, Nagy W, Mukhi N, and Weerawarana S. Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, 6:86–93, 2002.

[33] Day-Richter J, Harris MA, Haendel M, and Lewis S. OBO-Edit - An Ontology Editor for Biologists. *Bioinformatics*, Jun 2007.

[34] Dehne F, Langston MA, Luo X, Pitre S, Shaw P, and Zhang Y. The Cluster Editing Problem: Implementations and Experiments. In *Proc. of International Workshop on Parameterized and Exact Computation (IWPEC 2006)*, volume 4169 of *LNCS*, pages 13–24. Springer, 2006.

[35] Delvaux S and Horsten L. On best transitive approximations of simple graphs. *Acta informatica*, 40(9):637–655, 2004.

[36] Dondrup M, Goesmann A, Bartels D, Kalinowski J, Krause L, Linke B, Rupp O, Sczyrba A, Pühler A, and Meyer F. EMMA: a platform for consistent storage and efficient analysis of microarray data. *J Biotechnol*, 106(2-3):135–146, Dec 2003.

[37] Drysdale RA, Crosby MA, and Consortium F. FlyBase: genes and gene models. *Nucleic Acids Res*, 33(Database issue):D390–D395, Jan 2005.

[38] Engels V and Wendisch VF. The DeoR-type regulator SugR represses expression of ptsG in Corynebacterium glutamicum. *J Bacteriol*, 189(8):2955–2966, Apr 2007.

[39] Enright AJ, Dongen SV, and Ouzounis CA. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*, 30(7):1575–1584, Apr 2002.

[40] Enright AJ and Ouzounis CA. GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics*, 16(5):451–457, May 2000.

[41] Erill I, Escribano M, Campoy S, and Barbé J. In silico analysis reveals substantial variability in the gene contents of the gamma proteobacteria LexA-regulon. *Bioinformatics*, 19(17):2225–2236, Nov 2003.

[42] Erill I, Jara M, Salvador N, Escribano M, Campoy S, and Barbé J. Differences in LexA regulon structure among Proteobacteria through in vivo assisted comparative genomics. *Nucleic Acids Res*, 32(22):6617–6626, 2004.

[43] Everitt BS. *Cluster Analysis*. Edward Arnold, London, 3rd edition, 1993.

[44] Fawcett P, Eichenberger P, Losick R, and Youngman P. The transcriptional profile of early to middle sporulation in Bacillus subtilis. *Proc Natl Acad Sci U S A*, 97(14):8063–8068, Jul 2000.

[45] Frey BJ and Dueck D. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976, 2007.

[46] Fruchterman TMJ and Reingold EM. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[47] Fudou R, Jojima Y, Seto A, Yamada K, Kimura E, Nakamatsu T, Hiraishi A, and Yamanaka S. Corynebacterium efficiens sp. nov., a glutamic-acid-producing species from soil and vegetables. *Int J Syst Evol Microbiol*, 52(Pt 4):1127–1131, Jul 2002.

[48] Galas DJ and Schmitz A. DNAse footprinting: a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Res*, 5(9):3157–3170, Sep 1978.

[49] Garwood K, McLaughlin T, Garwood C, Joens S, Morrison N, Taylor CF, Carroll K, Evans C, Whetton AD, Hart S, Stead D, Yin Z, Brown AJP, Hesketh A, Chater K, Hansson L, Mewissen M, Ghazal P, Howard J, Lilley KS, Gaskell SJ, Brass A, Hubbard SJ, Oliver SG, and Paton NW. PEDRo: a database for storing, searching and disseminating experimental proteomics data. *BMC Genomics*, 5(1):68, Sep 2004.

[50] Gerstmeir R, Wendisch VF, Schnicke S, Ruan H, Farwick M, Reinscheid D, and Eikmanns BJ. Acetate metabolism and its regulation in Corynebacterium glutamicum. *J Biotechnol*, 104(1-3):99–122, Sep 2003.

[51] Goesmann A, Linke B, Bartels D, Dondrup M, Krause L, Neuweger H, Oehm S, Paczian T, Wilke A, and Meyer F. BRIGEP–the BRIDGE-based genome-transcriptome-proteome browser. *Nucleic Acids Res*, 33(Web Server issue):W710–W716, Jul 2005.

[52] Goesmann A, Linke B, Rupp O, Krause L, Bartels D, Dondrup M, McHardy AC, Wilke A, Pühler A, and Meyer F. Building a BRIDGE for the integration of heterogeneous data from functional genomics into a platform for systems biology. *J Biotechnol*, 106(2-3):157–167, Dec 2003.

[53] Goldovsky L, Cases I, Enright AJ, and Ouzounis CA. BioLayout(Java): versatile network visualisation of structural and functional relationships. *Applied Bioinformatics*, 4(1):71–74, 2005.

[54] Gramm J, Guo J, Hüffner F, and Niedermeier R. Automated generation of search tree algorithms for hard graph modification problems. *Algorithmica*, 39(4):321–347, 2004.

[55] Gramm J, Guo J, Hüffner F, and Niedermeier R. Graph-modeled data clustering: Exact algorithm for clique generation. *Theor. Comput. Syst.*, 38(4):373–392, 2005.

[56] Guo A, He K, Liu D, Bai S, Gu X, Wei L, and Luo J. DATF: a database of Arabidopsis transcription factors. *Bioinformatics*, 21(10):2568–2569, May 2005.

[57] Hartigan JA. *Clustering Algorithms*. Wiley, 1975.

[58] Hellman LM and Fried MG. Electrophoretic mobility shift assay (EMSA) for detecting protein-nucleic acid interactions. *Nat Protoc*, 2(8):1849–1861, 2007.

[59] Hermann T. Industrial production of amino acids by coryneform bacteria. *J Biotechnol*, 104(1-3):155–172, Sep 2003.

[60] Herrgård MJ, Covert MW, and Palsson B. Reconstruction of microbial transcriptional regulatory networks. *Curr Opin Biotechnol*, 15(1):70–77, Feb 2004.

[61] Hüser AT, Becker A, Brune I, Dondrup M, Kalinowski J, Plassmeier J, Pühler A, Wiegräbe I, and Tauch A. Development of a Corynebacterium glutamicum DNA microarray and validation by genome-wide expression profiling during growth with propionate as carbon source. *J Biotechnol*, 106(2-3):269–286, Dec 2003.

[62] Huerta AM, Salgado H, Thieffry D, and Collado-Vides J. RegulonDB: a database on transcriptional regulation in Escherichia coli. *Nucleic Acids Res*, 26(1):55–59, Jan 1998.

[63] Ishihama A. Functional modulation of Escherichia coli RNA polymerase. *Annu Rev Microbiol*, 54:499–518, 2000.

[64] Ishii T, Yoshida K, Terai G, Fujita Y, and Nakai K. DBTBS: a database of Bacillus subtilis promoters and transcription factors. *Nucleic Acids Res*, 29(1):278–280, Jan 2001.

[65] Jacques PE, Gervais AL, Cantin M, Lucier JF, Dallaire G, Drouin G, Gaudreau L, Goulet J, and Brzezinski R. MtbRegList, a database dedicated to the analysis of transcriptional regulation in Mycobacterium tuberculosis. *Bioinformatics*, 21(10):2563–2565, May 2005.

[66] Janga SC, Collado-Vides J, and Moreno-Hagelsieb G. Nebulon: a system for the inference of functional relationships of gene products from the rearrangement of predicted operons. *Nucleic Acids Res*, 33(8):2521–2530, 2005.

[67] Kalinowski J, Bathe B, Bartels D, Bischoff N, Bott M, Burkovski A, Dusch N, Eggeling L, Eikmanns BJ, Gaigalat L, Goesmann A, Hartmann M, Huthmacher K, Krämer R, Linke B, McHardy AC, Meyer F, Möckel B, Pfefferle W, Pühler A, Rey DA, Rückert C, Rupp O, Sahm H, Wendisch VF, Wiegräbe I, and Tauch A. The complete Corynebacterium glutamicum ATCC 13032 genome sequence and its impact on the production of L-aspartate-derived amino acids and vitamins. *J Biotechnol*, 104(1-3):5–25, Sep 2003.

[68] Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, and Hirakawa M. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res*, 34(Database issue):D354–D357, Jan 2006.

[69] Kel AE, Gössling E, Reuter I, Cheremushkin E, Kel-Margoulis OV, and Wingender E. MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res*, 31(13):3576–3579, Jul 2003.

[70] Keseler IM, Collado-Vides J, Gama-Castro S, Ingraham J, Paley S, Paulsen IT, Peralta-Gil M, and Karp PD. EcoCyc: a comprehensive database resource for Escherichia coli. *Nucleic Acids Res*, 33(Database issue):D334–D337, Jan 2005.

[71] Köhler J, Baumbach J, Taubert J, Specht M, Skusa A, Rüegg A, Rawlings C, Verrier P, and Philippi S. Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics*, 22(11):1383–1390, Jun 2006.

[72] Köhler J, Philippi S, and Lange M. SEMEDA: ontology based semantic integration of biological databases. *Bioinformatics*, 19(18):2420–2427, Dec 2003.

[73] Kim HJ, Kim TH, Kim Y, and Lee HS. Identification and characterization of glxR, a gene involved in regulation of glyoxylate bypass in Corynebacterium glutamicum. *J Bacteriol*, 186(11):3453–3460, Jun 2004.

[74] King AD, Przulj N, and Jurisica I. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, 2004.

[75] Koch DJ, Rückert C, Albersmeier A, Hüser AT, Tauch A, Pühler A, and Kalinowski J. The transcriptional regulator SsuR activates expression of the Corynebacterium glutamicum sulphonate utilization genes in the absence of sulphate. *Mol Microbiol*, 58(2):480–494, Oct 2005.

[76] Koehler J, Rawlings C, Verrier P, Mitchell R, Skusa A, Ruegg A, and Philippi S. Linking experimental results, biological networks and sequence analysis methods using Ontologies and Generalised Data Structures. *In Silico Biol*, 5(1):33–44, 2005.

[77] Kornberg HL. The role and control of the glyoxylate cycle in Escherichia coli. *Biochem J*, 99(1):1–11, Apr 1966.

[78] Krause A, Stoye J, and Vingron M. Large scale hierarchical clustering of protein sequences. *BMC Bioinformatics*, 6:15, 2005.

[79] Küster H, Becker A, Firnhaber C, Hohnjec N, Manthey K, Perlick AM, Bekel T, Dondrup M, Henckel K, Goesmann A, Meyer F, Wipf D, Requena N, Hildebrandt U, Hampp R, Nehls U, Krajinski F, Franken P, and Pühler A. Development of bioinformatic tools to support EST-sequencing, in silico- and microarray-based transcriptome profiling in mycorrhizal symbioses. *Phytochemistry*, 68(1):19–32, Jan 2007.

[80] Larisch C, Nakunst D, Hueser AT, Tauch A, and Kalinowski J. The alternative sigma factor SigB of Corynebacterium glutamicum modulates global gene expression during transition from exponential growth to stationary phase. *BMC Genomics*, 8:4, 2007.

[81] Li N and Tompa M. Analysis of computational approaches for motif discovery. *Algorithms Mol Biol*, 1:8, 2006.

[82] Lozada-Chavez I, Janga SC, and Collado-Vides J. Bacterial regulatory networks are extremely flexible in evolution. *Nucleic Acids Res*, 34(12):3434–3445, 2006.

[83] Ma HW, Buer J, and Zeng AP. Hierarchical structure and modules in the Escherichia coli transcriptional regulatory network revealed by a new top-down approach. *BMC Bioinformatics*, 5:199, 2004.

[84] Ma HW, Kumar B, Ditges U, Gunzer F, Buer J, and Zeng AP. An extended transcriptional regulatory network of Escherichia coli and analysis of its hierarchical structure and network motifs. *Nucleic Acids Res*, 32(22):6643–6649, 2004.

[85] Makita Y, Nakao M, Ogasawara N, and Nakai K. DBTBS: database of transcriptional regulation in Bacillus subtilis and its contribution to comparative genomics. *Nucleic Acids Res*, 32(Database issue):D75–D77, Jan 2004.

[86] Mangan S and Alon U. Structure and function of the feed-forward loop network motif. *Proc Natl Acad Sci U S A*, 100(21):11980–11985, Oct 2003.

[87] Matic I, Taddei F, and Radman M. Survival versus maintenance of genetic stability: a conflict of priorities during stress. *Res Microbiol*, 155(5):337–341, Jun 2004.

[88] Matys V, Kel-Margoulis OV, Fricke E, Liebich I, Land S, Barre-Dirrie A, Reuter I, Chekmenev D, Krull M, Hornischer K, Voss N, Stegmaier P, Lewicki-Potapov B, Saxel H, Kel AE, and Wingender E. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res*, 34(Database issue):D108–D110, Jan 2006.

[89] Mazon G, Erill I, Campoy S, Cortes P, Forano E, and Barbe J. Reconstruction of the evolutionary history of the LexA-binding sequence. *Microbiology*, 150(Pt 11):3783–3795, Nov 2004.

[90] Meyer F, Goesmann A, McHardy AC, Bartels D, Bekel T, Clausen J, Kalinowski J, Linke B, Rupp O, Giegerich R, and Pühler A. GenDB–an open source genome annotation system for prokaryote genomes. *Nucleic Acids Res*, 31(8):2187–2195, Apr 2003.

[91] Mlecnik B, Scheideler M, Hackl H, Hartler J, Sanchez-Cabo F, and Trajanoski Z. PathwayExplorer: web service for visualizing high-throughput expression data on biological pathways. *Nucleic Acids Res*, 33(Web Server issue):W633–W637, Jul 2005.

[92] Münch R, Hiller K, Grote A, Scheer M, Klein J, Schobert M, and Jahn D. Virtual Footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics*, 21(22):4187–4189, Nov 2005.

[93] Muench R, Hiller K, Barg H, Heldt D, Linz S, Wingender E, and Jahn D. PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res*, 31(1):266–269, 2003.

[94] Nakamura Y, Nishio Y, Ikeo K, and Gojobori T. The genome stability in Corynebacterium species due to lack of the recombinational repair system. *Gene*, 317(1-2):149–155, Oct 2003.

[95] Nakunst D, Larisch C, Hueser AT, Tauch A, Puehler A, and Kalinowski J. The Extracytoplasmic Function-Type Sigma Factor SigM of Corynebacterium glutamicum ATCC 13032 Is Involved in Transcription of Disulfide Stress-Related Genes. *J Bacteriol*, 189(13):4696–4707, Jul 2007.

[96] Neuweger H, Baumbach J, Albaum S, Bekel T, Dondrup M, Hueser A, Kalinowski J, Oehm S, Puehler A, Rahmann S, Weile J, and Goesmann A. CoryneCenter - An online resource for the integrated analysis of corynebacterial genome and transcriptome data. *BMC Syst Biol*, 1(1):55, Nov 2007.

[97] Nishio Y, Nakamura Y, Kawarabayasi Y, Usuda Y, Kimura E, Sugimoto S, Matsui K, Yamagishi A, Kikuchi H, Ikeo K, and Gojobori T. Comparative complete genome sequence analysis of the amino acid replacements responsible for the thermostability of Corynebacterium efficiens. *Genome Res*, 13(7):1572–1579, Jul 2003.

[98] Nyström T. Conditional senescence in bacteria: death of the immortals. *Mol Microbiol*, 48(1):17–23, Apr 2003.

[99] Nyström T. Growth versus maintenance: a trade-off dictated by RNA polymerase availability and sigma factor competition? *Mol Microbiol*, 54(4):855–862, Nov 2004.

[100] Pabo CO and Sauer RT. Transcription factors: structural families and principles of DNA recognition. *Annu Rev Biochem*, 61:1053–1095, 1992.

[101] Paccanaro A, Casbon JA, and Saqi MA. Spectral clustering of protein sequences. *Nucleic Acids Research*, 34(5):1571–1580, 2006.

[102] Perez-Rueda E and Collado-Vides J. The repertoire of DNA-binding transcriptional regulators in Escherichia coli K-12. *Nucleic Acids Res*, 28(8):1838–1847, Apr 2000.

[103] Philippi S and Köhler J. Addressing the problems with life-science databases for traditional uses and systems biology. *Nat Rev Genet*, 7(6):482–488, Jun 2006.

[104] Pillai S, Silventoinen V, Kallio K, Senger M, Sobhany S, Tate J, Velankar S, Golovin A, Henrick K, Rice P, Stoehr P, and Lopez R. SOAP-based services provided by the European Bioinformatics Institute. *Nucleic Acids Res*, 33(Web Server issue):W25–W28, Jul 2005.

[105] Pipenbacher P, Schliep A, Schneckener S, Schoenhuth A, Schomburg D, and Schrader R. ProClust: improved clustering of protein sequences with an extended graph-based approach. *Bioinformatics*, 18 Suppl 2:S182–S191, 2002.

[106] Price MN, Huang KH, Alm EJ, and Arkin AP. A novel method for accurate operon predictions in all sequenced prokaryotes. *Nucleic Acids Res*, 33(3):880–892, 2005.

[107] Rahmann S, Mueller T, and Vingron M. On the power of profiles for transcription factor binding site detection. *Statistical Applications in Genetics and Molecular Biology*, 2(1), 2003.

[108] Rahmann S, Wittkop T, Baumbach J, Martin M, Truß A, and Böcker S. Exact and Heuristic Algorithms for Weighted Cluster Editing. *Comput Syst Bioinformatics Conf*, 6(1):391–401, Aug 2007.

[109] Reed JL and Palsson B. Thirteen years of building constraint-based in silico models of Escherichia coli. *J Bacteriol*, 185(9):2692–2699, May 2003.

[110] Resendis-Antonio O, Freyre-González JA, Menchaca-Méndez R, Gutiérrez-Ríos RM, Martínez-Antonio A, Avila-Sánchez C, and Collado-Vides J. Modular analysis of the transcriptional regulatory network of E. coli. *Trends Genet*, 21(1):16–20, Jan 2005.

[111] Rey DA, Nentwich SS, Koch DJ, Rückert C, Pühler A, Tauch A, and Kalinowski J. The McbR repressor modulated by the effector substance S-adenosylhomocysteine controls directly the transcription of a regulon involved in sulphur metabolism of Corynebacterium glutamicum ATCC 13032. *Mol Microbiol*, 56(4):871–887, May 2005.

[112] Robison K, McGuire AM, and Church GM. A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete Escherichia coli K-12 genome. *J Mol Biol*, 284(2):241–254, Nov 1998.

[113] Salgado H, Gama-Castro S, Martínez-Antonio A, Díaz-Peredo E, Sánchez-Solano F, Peralta-Gil M, Garcia-Alonso D, Jiménez-Jacinto V, Santos-Zavaleta A, Bonavides-Martínez C, and Collado-Vides J. RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12. *Nucleic Acids Res*, 32(Database issue):D303–D306, Jan 2004.

[114] Salgado H, Gama-Castro S, Peralta-Gil M, Diaz-Peredo E, Sanchez-Solano F, Santos-Zavaleta A, Martinez-Flores I, Jimenez-Jacinto V, Bonavides-Martinez C, Segura-Salazar J, Martinez-Antonio A, and Collado-Vides J. RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res*, 34(Database issue):D394–D397, Jan 2006.

[115] Salgado H, Santos A, Garza-Ramos U, van Helden J, Díaz E, and Collado-Vides J. RegulonDB (version 2.0): a database on transcriptional regulation in Escherichia coli. *Nucleic Acids Res*, 27(1):59–60, Jan 1999.

[116] Salgado H, Santos-Zavaleta A, Gama-Castro S, Millán-Zárate D, Blattner FR, and Collado-Vides J. RegulonDB (version 3.0): transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Res*, 28(1):65–67, Jan 2000.

[117] Salgado H, Santos-Zavaleta A, Gama-Castro S, Millán-Zárate D, Díaz-Peredo E, Sánchez-Solano F, Pérez-Rueda E, Bonavides-Martínez C, and Collado-Vides J. RegulonDB (version 3.2): transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Res*, 29(1):72–74, Jan 2001.

[118] Schneider TD and Stephens RM. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*, 18(20):6097–6100, Oct 1990.

[119] Schuster-Boeckler B, Schultz J, and Rahmann S. HMM Logos for visualization of protein families. *BMC Bioinformatics*, 5:7, Jan 2004.

[120] Shamir R, Sharan R, and Tsur D. Cluster graph modification problems. *Discrete Applied Mathematics*, 144:173–182, 2004.

[121] Stormo GD. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, Jan 2000.

[122] Sun LV, Chen L, Greil F, Negre N, Li TR, Cavalli G, Zhao H, Steensel BV, and White KP. Protein-DNA interaction mapping using genomic tiling path microarrays in Drosophila. *Proc Natl Acad Sci U S A*, 100(16):9428–9433, Aug 2003.

[123] Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, Koonin EV, Krylov DM, Mazumder R, Mekhedov SL, Nikolskaya AN, Rao BS, Smirnov S, Sverdlov AV, Vasudevan S, Wolf YI, Yin JJ, and Natale DA. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4:41, Sep 2003.

[124] Tauch A, Kaiser O, Hain T, Goesmann A, Weisshaar B, Albersmeier A, Bekel T, Bischoff N, Brune I, Chakraborty T, Kalinowski J, Meyer F, Rupp O, Schneiker S, Viehoever P, and Pühler A. Complete genome sequence and analysis of the multiresistant nosocomial pathogen Corynebacterium jeikeium K411, a lipid-requiring bacterium of the human skin flora. *J Bacteriol*, 187(13):4671–4682, Jul 2005.

[125] Teichmann SA and Babu MM. Gene regulatory network growth by duplication. *Nat Genet*, 36(5):492–496, May 2004.

[126] Tompa M, Li N, Bailey TL, Church GM, Moor BD, Eskin E, Favorov AV, Frith MC, Fu Y, Kent WJ, Makeev VJ, Mironov AA, Noble WS, Pavesi G, Pesole G, Régnier

M, Simonis N, Sinha S, Thijs G, van Helden J, Vandenbogaert M, Weng Z, Workman C, Ye C, and Zhu Z. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol*, 23(1):137–144, Jan 2005.

[127] Wheeler DL, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Edgar R, Federhen S, Geer LY, Kapustin Y, Khovayko O, Landsman D, Lipman DJ, Madden TL, Maglott DR, Ostell J, Miller V, Pruitt KD, Schuler GD, Sequeira E, Sherry ST, Sirotkin K, Souvorov A, Starchenko G, Tatusov RL, Tatusova TA, Wagner L, and Yaschenko E. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*, 35(Database issue):D5–12, Jan 2007.

[128] Wilkinson MD and Links M. BioMOBY: an open source biological web services proposal. *Brief Bioinform*, 3(4):331–341, Dec 2002.

[129] Wingender E. TRANSFAC, TRANSPATH and CYTOMER as starting points for an ontology of regulatory networks. *In Silico Biol*, 4(1):55–61, 2004.

[130] Wingender E, Chen X, Fricke E, Geffers R, Hehl R, Liebich I, Krull M, Matys V, Michael H, Ohnhäuser R, Prüss M, Schacherer F, Thiele S, and Urbach S. The TRANSFAC system on gene expression regulation. *Nucleic Acids Res*, 29(1):281–283, Jan 2001.

[131] Wittkop T, Baumbach J, Lobo F, and Rahmann S. Large scale clustering of protein sequences with FORCE – A layout based heuristic for weighted cluster editing. *BMC Bioinformatics*, 8(1):396, Oct 2007.

[132] Wu TD, Nevill-Manning CG, and Brutlag DL. Fast probabilistic analysis of sequence function using scoring matrices. *Bioinformatics*, 16(3):233–244, Mar 2000.

Abbreviations

BeH	Best Hit
BM	Binding Motif
BS	Binding Site
B. subtilis	*Bacillus subtilis*
C. diphthaeriae	*Corynebacterium diphthaeriae*
C. efficiens	*Corynebacterium efficiens*
cg	Cluster growing
C. glutamicum	*Corynebacterium glutamicum*
C. jeikeium	*Corynebacterium jeikeium*
COG	Cluster of Orthologous Groups of proteins
COMA	COntradictions in MicroArrays (feature of CoryneRegNet)
CoryneRegNet	Corynebacterial Regulatory Networks
Cov	Coverage
DBMS	Database Managment System
DNA	DeoxyriboNucleic acid
EC number	Enzyme Commission number
E. coli	*Escherichia coli*
FORCE	FORce based Cluster Editing (feature of CoryneRegNet)
GCEP	Graph Cluster Editing Problem
HMM	Hidden Markov Model
HSP	High-Scoring Pair
km	k-means
MoRAine	Motif Re-Annotation (feature of CoryneRegNet)
M. tuberculosis	*Mycobacterium tuberculosis*
NCBI	National Center for Biotechnology Information
ONDEX	ONtological inDEXing
PFM	Position Frequency Matrix
PSSM	Position Specific Scoring Matrix
PWM	Position Weight Matrix
SCOP	Structural Classification of Proteins
simC	Motif-cluster similarity
simS	Motif-seed similarity
SOAP	Simple Object Access Protocol
SoH	Sum of Hit
TF	Transcription Factor
TFBS	Transcription Factor Binding Site
TFBScan	Transcription Factor Binding site Scan (feature of CoryneRegNet)
TFBM	Transcription Factor Binding Motif

WGCEP	Weighted Graph Cluster Editing Problem
WSDL	Web Service Definition Language
XML	Extensible Markup Language

A Cooperations

Large parts of the biological reconstruction of the database content have been performed by Karina Brinkrolf, who worked on the sister project of CoryneRegNet at the wet lab side. She furthermore helped to test most of the analysis and visualization features of CoryneRegNet.

The interconnection to GenDB and EMMA, which finally has resulted in CoryneCenter was performed in cooperation with Heiko Neuweger, who contributed to the implementation of the GenDB Web Service server. Dr. Michael Dondrup implemented the EMMA Web Service server.

The work on FORCE has been performed together with Tobias Wittkop, conjointly in all parts of development, implementation, evaluation, and publication.

The application cases in the Sections 4.1.1, 4.1.2, and 4.1.3 have been performed with the help of Karina Brinkrolf, while the last application case in Section 4.1.4 was done with the help of Dr. Andrea Hüser.

B MoRAine motif readjustment results

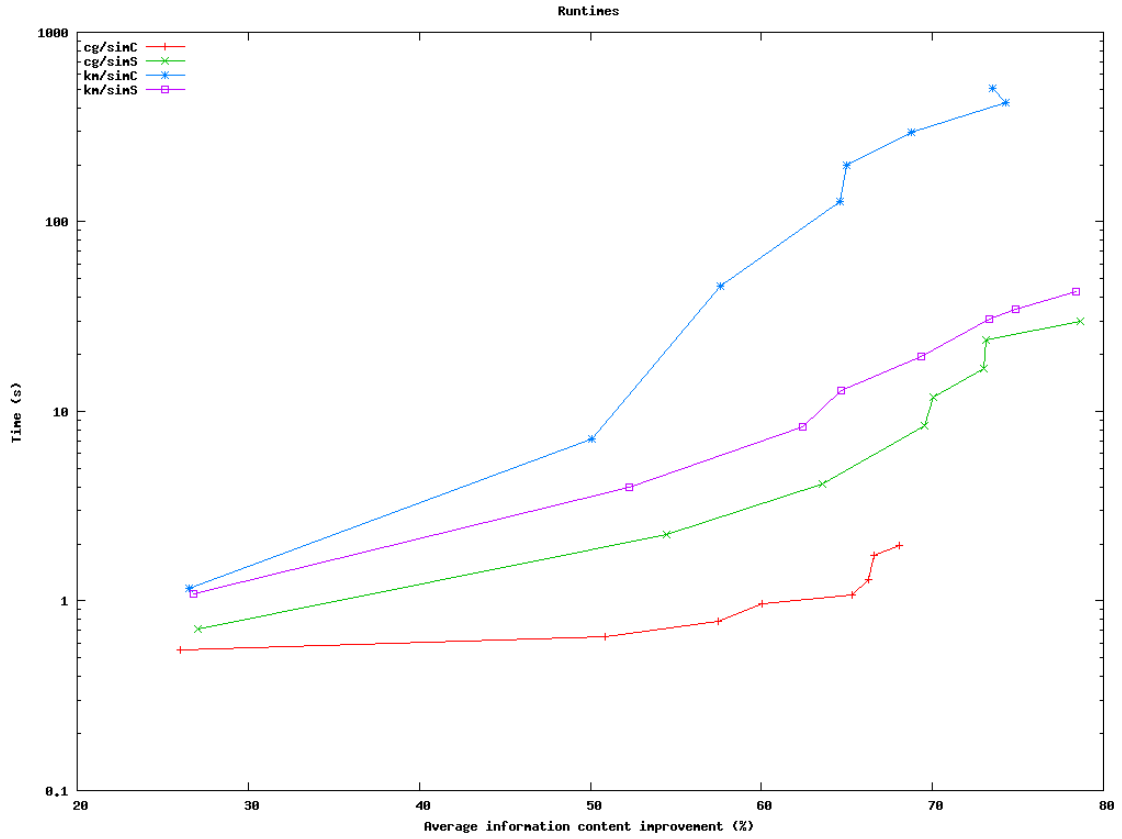


Figure B.1: This image illustrates the average information content improvement plotted against the necessary running time of MoRAine. Note that the y-axis is log-scaled.

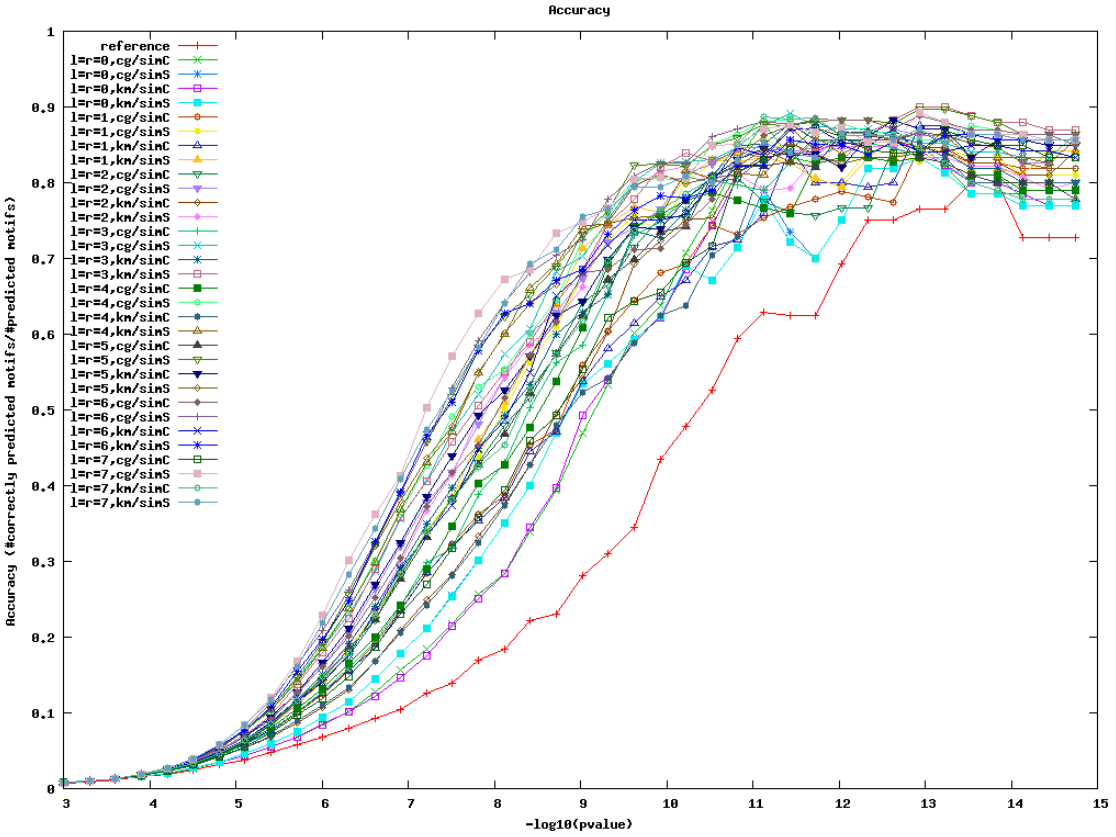


Figure B.2: Accurately predicted motifs for different p-value thresholds, for $0 \leq l = r \leq 7$. For the reference curve we used original PFMs learned from original database TFBMs.

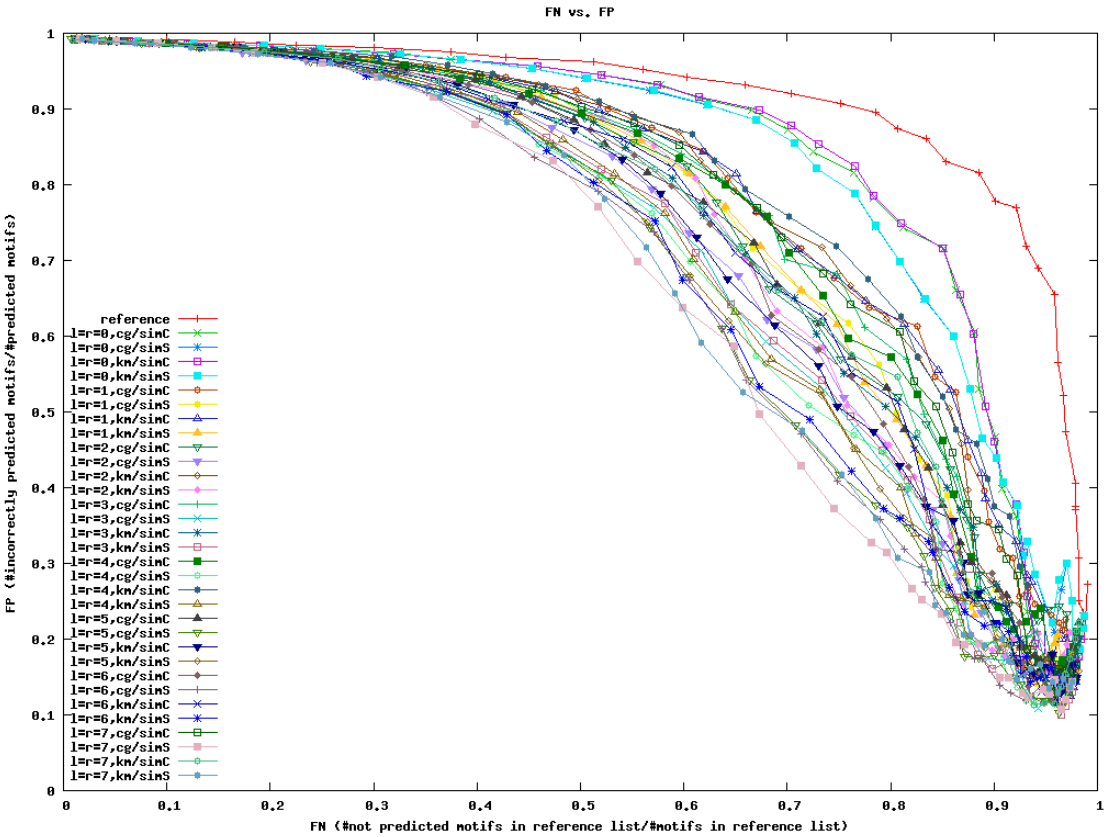


Figure B.3: False negative (FN) vs. false positive (FP) rate for $0 \leq l = r \leq 7$. For the reference curve we used original PFMs learned from original database TFBMs.

C Visualization of FORCE clustering results

Graphical summary of the obtained clustering results of FORCE. We used MATLAB scripts provided by Paccanaro to create images similar to those of Figure 3 in [101]. Each row corresponds to a cluster. Green bars represent a protein assignment to a cluster; each protein is present in only one of the clusters. Boundaries between superfamilies are shown by vertical red lines; boundaries between families within each superfamily are shown by dotted blue lines. To each figure is given the used similarity function and the dataset.

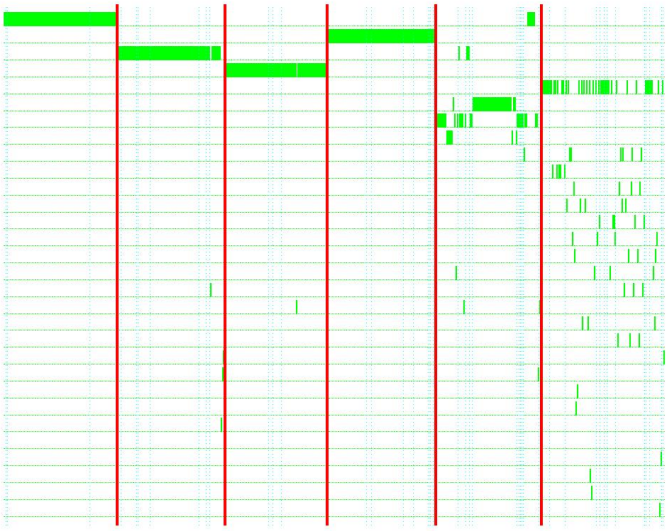


Figure C.1: Similarity function: BeH, dataset ASTRAL95_1_161

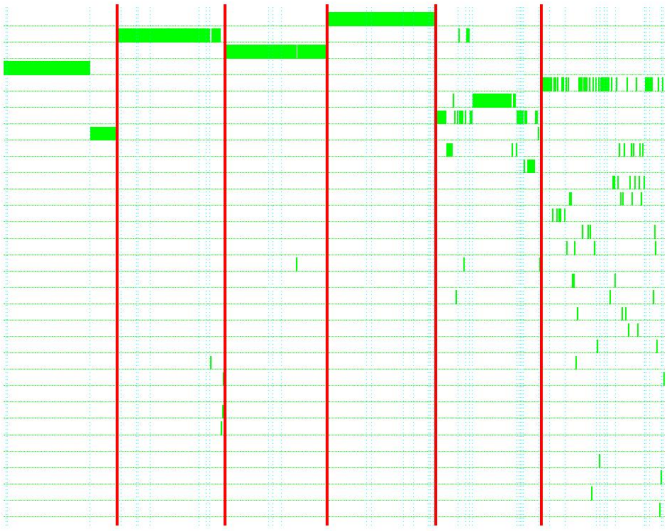


Figure C.2: Similarity function: SoH, dataset ASTRAL95_1_161

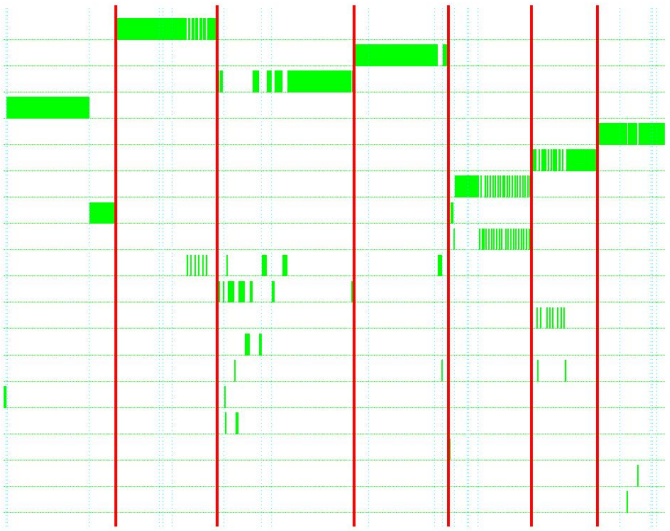


Figure C.3: Similarity function: BeH, dataset ASTRAL95_2_161

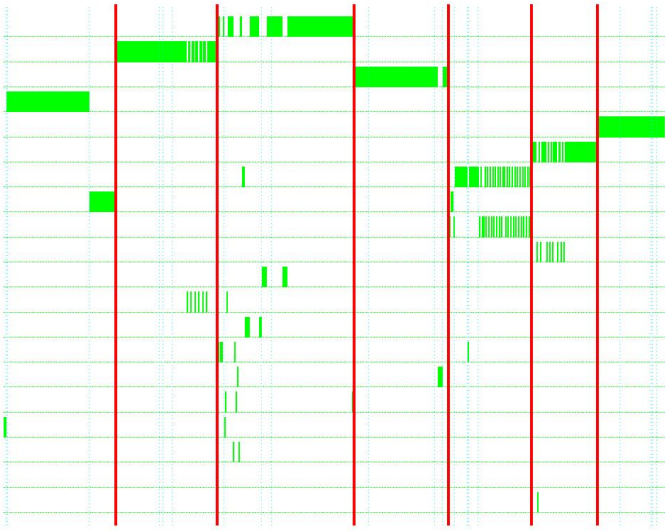


Figure C.4: Similarity function: SoH, dataset ASTRAL95_2_161

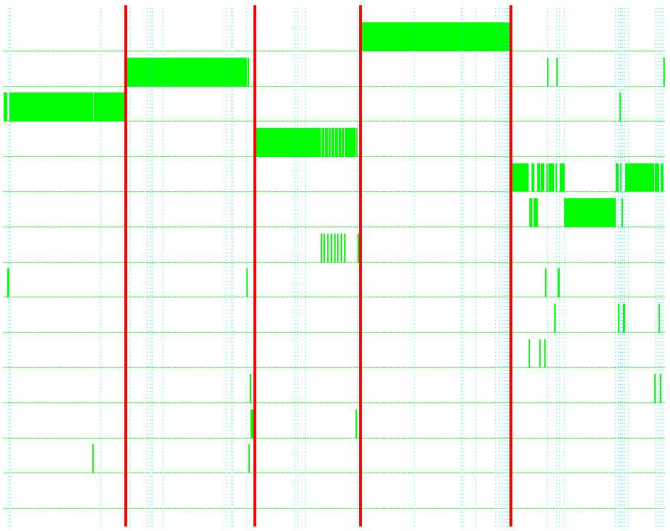


Figure C.5: Similarity function: BeH, dataset ASTRAL95_1_171

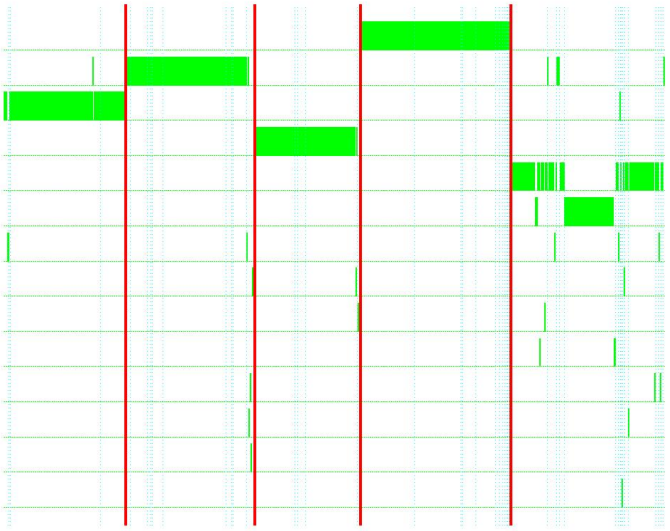


Figure C.6: Similarity function: SoH, dataset ASTRAL95_1_171

Figure C.7: Similarity function: BeH, dataset ASTRAL95_2_171

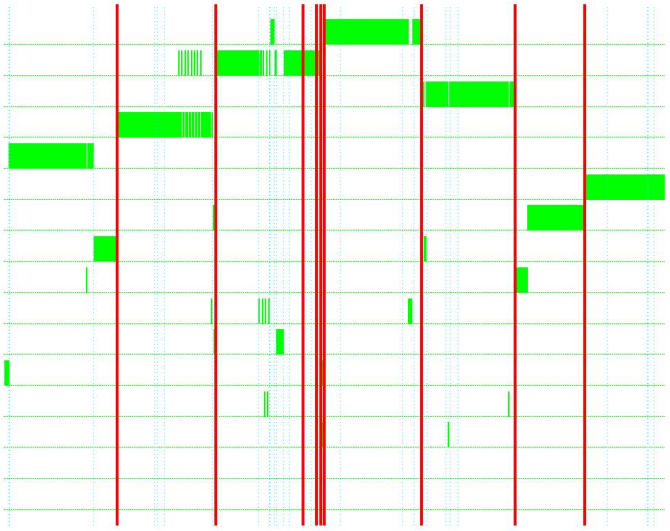


Figure C.8: Similarity function: SoH, dataset ASTRAL95_2_171

D FORCE evaluation for varying parameters

Clustering evaluation Quality evaluation for different scoring themes and datasets. This table shows for the 50 best F-measures the threshold, the coverage factor, the used dataset and the similarity function.

F-measure	Threshold	Coverage factor	Dataset/Scoring sceme
0.911	18	-3	1v171SoH
0.899	15	-3.4	1v171BeH
0.896	17	-3	1v171BeH
0.895	15	-3.2	2v171SoH
0.888	20	-3.4	1v171BeH
0.887	13	-4	1v171BeH
0.885	14	-2.4	2v161BeH
0.885	12	-3.6	1v171BeH
0.882	20	-3.2	1v171BeH
0.881	17	-2.6	1v171SoH
0.880	19	-1.8	2v161BeH
0.880	16	-2.8	1v171SoH
0.879	17	-2.4	2v171SoH
0.878	11	-3	2v171SoH
0.878	16	-3.2	1v171BeH
0.878	19	-1.6	2v161SoH
0.877	13	-3	1v171BeH
0.877	18	-1.8	2v161SoH
0.877	16	-3.2	1v171SoH
0.876	20	-1.8	2v161BeH
0.875	8	-4.2	1v171SoH
0.875	6	-4	1v171BeH
0.874	18	-3.8	1v171BeH
0.874	20	-4	1v171BeH
0.874	14	-2	2v161SoH
0.874	19	-3.8	1v171BeH
0.873	15	-1.6	2v161SoH
0.873	18	-2.2	2v161BeH
0.873	15	-3.2	1v171BeH
0.873	16	-3	1v171SoH
0.873	18	-2.8	2v171SoH
0.872	12	-4	1v171SoH
0.872	7	-4	1v171BeH
0.872	19	-3.6	1v171BeH
0.872	4	-4	1v171SoH
0.871	15	-2.8	1v171SoH
0.870	10	-4	1v171BeH
0.870	13	-3.2	1v171SoH
0.870	7	-3.6	1v171BeH
0.870	10	-3.6	1v171SoH
0.869	16	-2.6	1v171SoH
0.868	8	-3.4	1v171SoH
0.868	14	-4	1v171BeH
0.867	14	-3.4	1v171BeH
0.867	18	-2	2v161BeH
0.866	5	-3.6	1v171SoH
0.866	18	-2.8	1v171SoH
0.866	13	-3.2	2v171SoH
0.866	19	-1.8	1v171SoH
0.865	11	-3	1v171SoH

In Figure D.1 all results are plotted, for every dataset/scoring theme separately as heatmap.

Clustering evaluation for a fixed coverage In Figure D.2, we give F-measures for a range of thresholds, but with fixed coverage factor $f = 20$, for dataset ASTRAL95_1_161, and similarity function BeH.

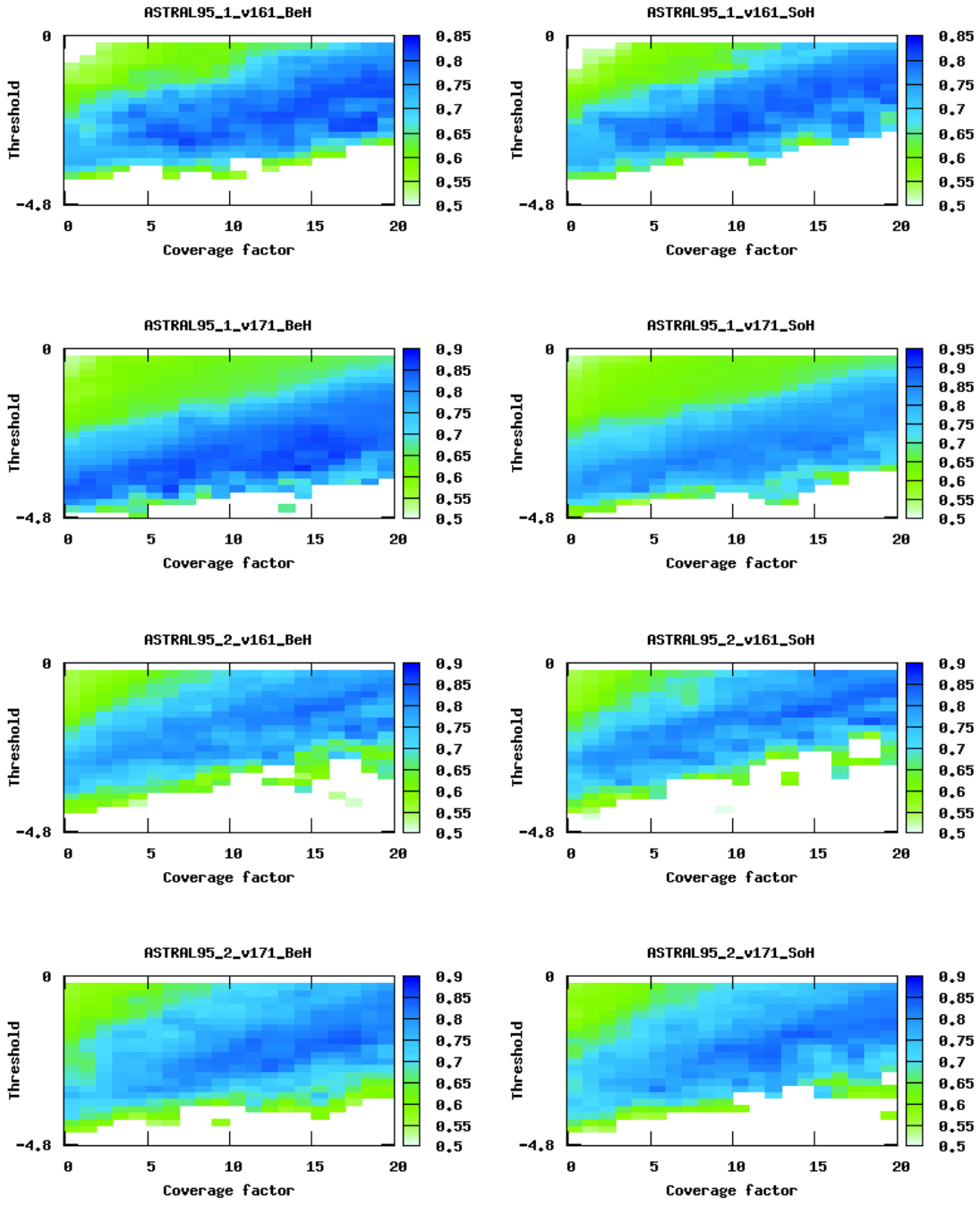


Figure D.1: Quality evaluation for different scoring themes and datasets. Plotted is the coverage factor at the x-axis and the threshold at the y-axis. The achieved F-measure is color-coded if it is > 0.5 (refer to the right bar beside the plots).

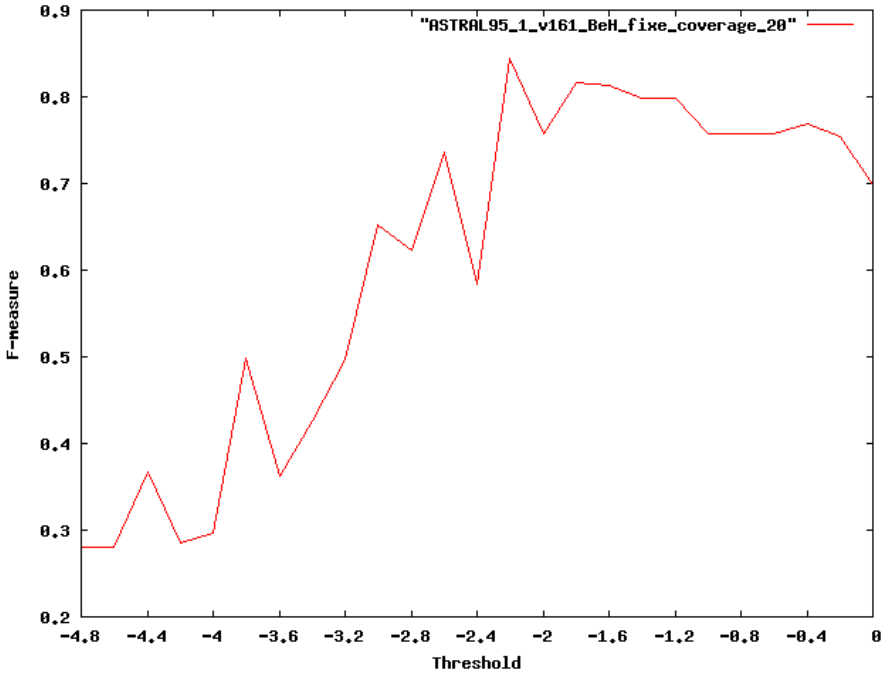


Figure D.2: Quality evaluation for different thresholds, but with fixed coverage factor $f = 20$, for dataset ASTRAL95_1_161, and similarity function BeH.

E Affinity propagation evaluation for varying parameters

Quality evaluation of Affinity Propagation for different scoring themes and datasets. This table shows the 50 best F-measures for a wide range of parameter constellations and coverage factors, for all used datasets and similarity functions.

F-measure	Coverage factor	preference *pre*	damping factor *df*	Dataset/Scoring sceme
0.709	13	700	0.80	2v171SoH
0.703	19	900	0.80	2v171SoH
0.702	16	800	0.80	2v171SoH
0.700	15	800	0.80	2v171BeH
0.699	13	800	0.80	2v171SoH
0.699	10	700	0.80	2v171SoH
0.699	14	800	0.80	2v171SoH
0.697	13	800	0.65	2v171BeH
0.697	17	800	0.80	2v171BeH
0.697	16	800	0.80	2v171BeH
0.697	15	800	0.80	2v171SoH
0.697	15	800	0.65	2v171SoH
0.697	11	700	0.80	2v171SoH
0.697	9	700	0.60	2v171SoH
0.697	15	800	0.60	2v171SoH
0.696	18	800	0.80	2v171BeH
0.695	11	700	0.80	2v171BeH
0.695	13	800	0.65	2v171SoH
0.695	13	800	0.70	2v171SoH
0.695	19	900	0.80	2v171BeH
0.695	19	800	0.60	2v171BeH
0.695	14	800	0.70	2v171SoH
0.694	20	900	0.80	2v171BeH
0.693	12	700	0.65	2v171SoH
0.693	12	700	0.60	2v171SoH
0.693	17	800	0.80	2v171SoH
0.692	9	700	0.80	2v171SoH
0.692	14	800	0.65	2v171SoH
0.691	20	800	0.70	2v171SoH
0.691	20	800	0.85	2v171SoH
0.691	11	700	0.65	2v171SoH
0.691	14	600	0.75	2v161SoH
0.690	9	700	0.65	2v171BeH
0.690	18	900	0.80	2v171SoH
0.690	20	900	0.80	2v171SoH
0.689	11	700	0.60	2v171BeH
0.689	20	800	0.90	2v171SoH
0.689	10	700	0.70	2v171BeH
0.689	15	800	0.70	2v171BeH
0.689	9	700	0.80	2v171BeH
0.688	15	800	0.70	2v171SoH
0.688	20	900	0.95	1v171BeH
0.687	16	800	0.65	2v171SoH
0.687	16	800	0.70	2v171SoH
0.687	18	900	0.80	2v171BeH
0.687	15	800	0.75	2v171SoH
0.687	20	900	0.70	2v171BeH
0.686	19	900	0.65	2v171BeH
0.686	12	700	0.65	2v171BeH
0.686	20	900	0.75	2v171SoH

In Figure E.1 all results are plotted, for every dataset/scoring theme separately as heatmap, but for a fixed damping factor $df = 0.8$.

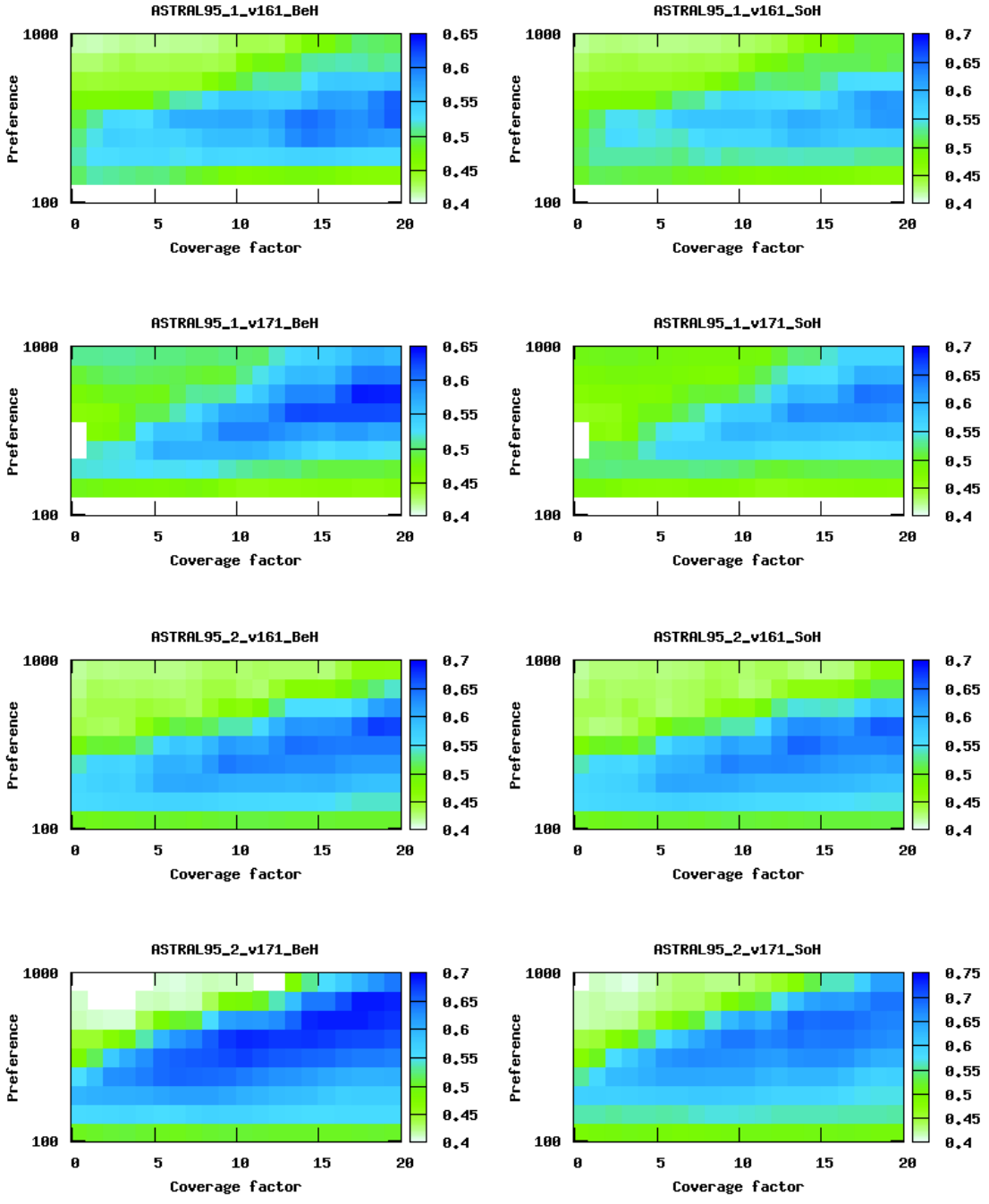


Figure E.1: Quality evaluation of Affinity Propagation for different scoring themes and datasets, for a fixed damping factor $df = 0.8$. Plotted is the coverage factor at the x-axis and the preference *pre* at the y-axis. The achieved F-measure is color-coded if it is > 0.4 (refer to the right bar beside the plots).

Erklärung

Hiermit versichere ich, dass ich diese Dissertation selbständig verfasst, keine anderen als die angegeben Quellen und Hilfsmittel benutzt und alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entlehnt sind, durch die Angabe von Quellen als Entlehnungen kenntlich gemacht habe.

Bielefeld, 21. Januar 2008

Jan Baumbach