INAUGURAL-DISSERTATION

zur Erlangung der Doktorwürde der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Kitiporn Plaimas, M.Sc.
aus Bangkok, Thailand

Tag der mündlichen Prüfung: 01.12.2011

Computational Analysis of the Metabolic Network of Microorganisms to Detect Potential Drug Targets

Gutachter:	Prof. Dr. Roland Eils
	Prof. Dr. Gerhard Reinelt

Abstract

Identifying essential genes in pathogens facilitates the identification of the corresponding proteins as potential drug targets and is the basis for understanding the minimum requirements for a synthetic cell. However, the experimental assessment of gene essentiality is resource-intensive and not feasible for all organisms, especially pathogens. Thus, the computational identification of new drug targets has become an important pursuit in biomedical research. In particular, essential metabolic enzymes have been successfully targeted by specific drugs. For directed drug development, the prediction of essential genes, especially in metabolic networks, is needed. In this thesis, I describe our development of a graph-based investigation tool aimed at finding possible deviations in a mutated network by knocking out particular reactions, and examining its producibility with a breadth-first search algorithm. We showed that this approach performed well at predicting new targets for antimalarial drugs. In addition, we analyzed the metabolic networks of bacteria and developed a machine learning approach based on various graph-based descriptors, including our own developed descriptor, that were potentially associated with the robustness and stabilization of metabolic networks. These descriptors were related to gene essentiality and included flux deviations, centrality and shortest paths. Besides these network topological features, we also used genomic and transcriptomic features, such as sequence characteristics and co-expression properties, as descriptors.

The machine learning technique was developed to identify drug targets in metabolism. The metabolic networks of *Escherichia coli*, *Pseudomonas aeruginosa* and *Salmonella typhimurium* were analyzed. The well-studied metabolic network of *Escherichia coli* was used because it was an ideal model for formulating and validating our method. With publicly available genome-wide knockout screens, it was shown that topological, genomic and transcriptomic features describing the network are sufficient for defining drug targets. Furthermore, we tested our method across bacterial species and strains by using the experimental data from the genome-wide knockout screens of one bacterial organism to infer essential genes for another related bacterial organism. Our method is generic, and it enables the prediction of essential genes from a bacterial reference organism to a related query organism without any knowledge about the essentiality of the genes of the query organism. In general, such a method is beneficial for inferring drug targets when experimental data about genome-wide knockout screens are not available for the investigated organism.

Zusammenfassung

Die Identifizierung von essentiellen Genen in Krankheitserregern unterstützt die Bestimmung von zugehörigen Proteinen als potentielle Zielmoleküle für Medikamente und erweitert unser Verständnis minimaler Bedingungen für eine synthetische Zelle. Der experimentelle Nachweis von essentiellen Genen ist jedoch kostenintensiv und nicht für alle pathogenen Organismen durchführbar. Daher ist die bioinformatische Identifizierung neuer Zielmoleküle ein wichtiger Bestandteil biomedizinischer Forschung geworden. Besonders essentielle metabolische Enzyme dieser Pathoorganismen erwiesen sich als gute Targets für spezifische Medikamente. Für eine gezielte Entwicklung von Medikamenten ist deshalb die Vorhersage von essentiellen Genen, speziell in metabolischen Netzwerken, vielversprechend. In dieser Arbeit habe ich metabolische Netzwerke von Bakterien mit Hilfe verschiedener Graph-basierten Deskriptoren, die mit Robustheit und Stabilität metabolischer Netzwerke verbunden sind, analysiert. Diese beschreiben den Grad des möglichen Einfluss und der Unersetzbarkeit der Knoten, wie z.B. Zentralität und Konnektivität. Dazu wurde von uns ein neuer Deskriptor entwickelt, der auf einem Kürzester-Wege Algorithmus basiert und denkbare „Umleitungen“ zu einem Targetenzym bestimmt. Im Weiteren haben wir eine Maschinenlernmethode zur Identifizierung von Zielmolekülen im Metabolismus entwickelt, die verschiedene topologische Eigenschaften des Netzwerks und genomische und transkriptomische Eigenschaften, wie Sequenzmerkmale und Ko-Expressionseigenschaften, berücksichtigt.

Wir haben metabolische Netzwerke von *Escherichia coli*, *Pseudomonas aeruginosa* und *Salmonella typhimurium* analysiert. Das bereits gut untersuchte metabolische Netzwerk von *Escherichia coli* wurde benutzt, da es ein ideales Modell darstellte, um unsere Methode zu entwerfen und zu validieren. Wir haben mit öffentlich verfügbaren Genom-weiten experimentellen Datensätzen von Knock-out Screens gezeigt, dass topologische, genomische und transkriptomische Eigenschaften des Netzwerkes ausreichend sind, um essenzielle Zielmoleküle zu bestimmen. Außerdem haben wir unsere Methode an weiteren Bakterien getestet, wobei wir experimentelle Daten eines Genom-weiten Knock-out Screens eines Organismus benutzt haben, essentielle Gene eines anderen verwandten Bakteriums abzuleiten. Unsere Methode ist allgemein anwendbar und ermöglicht die Vorhersage essenzieller Gene eines Organismus mit Hilfe eines bakteriellen Referenzorganismus ohne Wissen über die Essentialität der Gene des eigentlichen Organismus. Damit kann unsere Methode auch dann angewendet werden, Zielmoleküle abzuleiten, wenn experimentelle Daten von Genom-weiten Knock-out Screens für den analysierten Organismus nicht vorhanden sind.

Acknowledgments

I would like to express my sincere appreciation and thanks to all of the people who supported me during my PhD thesis.

First, I would like to thank Prof. Dr. Roland Eils for giving me the opportunity to work in his division and for his support, guidance and for having encouraged me. Furthermore, I would like to sincerely thank PD Dr. Rainer König for his invaluable assistance, dedicated time, critical feedback and discussions, as well as encouragement on my thesis work and cooperative projects. From the early stage of my work to my last final report, he always helped me accomplish the work. *Many thanks*!

I also wish to express my special thanks to Prof. Dr. Gerhard Reinelt and Prof. Dr. Hans Georg Bock, for their kindness, valuable advice and discussions. I would like to also thank Prof. Dr. Victor Sourjik (Center for Molecular Biology (ZMBH), University of Heidelberg, Germany) for his support in experimental validations and Prof. Dr. Ezekiel Adebiyi (Department of Computer and Information Sciences, Covenant University, Nigeria) for the malaria project I have been involved with. Furthermore, I would like to thank the Commission on Higher Education (CHE) of Thailand for generous financial support during my stay in Germany.

Many thanks to my colleagues Jan-Phillip Mallm for an excellent explanation of his previous project, Fabian Svara who tested our predictions in wet-lab experiments, Segun Fatumo and Yulin Wang for discussions and work for the target identification of malaria, Thorsten Bonato for our cooperative MILP project, and Marcus Oswald for providing his useful code to investigate the topology of metabolic networks and for reading my thesis. I am also grateful to all of my colleagues and friends at the *i*Bios group for our time together, having fun during our retreats and for every smile and greeting. I would especially like to thank members of the Network modeling group, Anna, Gunnar, Tobias, Richa, Kanna, Moritz, Heiko, Rosario and Robert for their help and care, discussions and the warm atmosphere at work and in our subgroup retreats. For providing an excellent IT infrastructure, I would like to thank Karlheinz Groß and Rolf Kabbe.

My thanks also go to all of my past and present instructors for their valuable lectures and instructions. I would also like to express my thanks to all of my friends for their encouragement and care during my thesis work. Moreover, I would like to express my sincere gratitude to my parents, sisters, brothers and relatives for their love, hearty encouragement, patience and unselfish sacrifices. Finally, a special thanks also goes to Apichat Suratanee for his love, warmest care and especially his patience during demanding times.

List of Publications

Proceedings

- Plaimas, K., Oswald, M., Eils, R. and König, R. (2007). Integrating genomic and transcriptomic data into graph based approaches for defining essential reactions in the metabolic network of *Escherichia coli*, Proceedings of the KDML 2007: Knowledge Discovery, Data Mining, and Machine Learning, Halle, Germany, pages 55-60.

Journals

- Plaimas, K., Mallm, JP., Oswald, M., Svara, F., Sourjik, V., Eils, R. and König, R. (2008). Machine learning based analyses on metabolic networks supports high-throughput knockout screens, BMC Systems Biology 2(67).

- Fatumo, S., Plaimas, K., Mallm, JP., Schramm, G., Adebiyi, E., Oswald, M., Eils, R. and König, R. (2009). Estimating novel potential drug targets of *Plasmodium falciparum* by analysing the metabolic network of knock-out strains *in silico*, Infection Genetics and Evolution 9(3):351-8.

- Plaimas, K., Eils, R. and König, R. (2010). Identifying essential genes in bacterial metabolic networks with machine learning methods, BMC Systems Biology 4(56).

- Fatumo, S., Plaimas, K., Adebiyi, E. and König, R. (2011). Comparing metabolic network models based on genomic and automatically inferred enzyme information from *Plasmodium* and its human host to define drug targets *in silico*, Infection, Genetics and Evolution 11(4):708-15.

Book chapters

- Plaimas, K. and König, R. (2011). Machine learning methods for identifying essential genes and proteins in networks, in Applied Statistics for Network Biology: Methods in Systems Biology (eds Dehmer, M., Emmert-Streib, F., Graber, A. and Salvador, A.), Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany. doi: 10.1002/9783527638079.ch10.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Defining essential genes or their corresponding proteins enables the identification of potential drug targets, and it may also provide an understanding of the minimal requirements for a synthetic cell. However, high-throughput experimental assays of the essentiality of coding genes are error-prone. Additionally, experimental screens are resource-intensive and not feasible for all organisms because, typically, a knock-out strain needs to be constructed for each gene. Furthermore, pathogenic bacterial organisms, such as *Salmonella*, are hazardous when cultivating and therefore require higher laboratory safety efforts.

Besides this, a variety of post-genomic techniques have emerged, and biochemical research is providing an ever extending amount of data about the molecular signaling interactions and metabolism of cells. This leads to novel insights into cellular mechanisms, not only for a single pathway, but also for multiple interacting pathways and players. Furthermore, understanding the behavior and interactions among cellular components contributes to the identification of new drug targets. Because the metabolism of a cell is essential for maintaining life and growth, metabolic enzymes have been successfully targeted by antibiotics inhibiting essential enzymatic processes in bacterial organisms [37, 68]. Thus, for directed drug development, computational methods are needed that support the prediction of essential genes, especially in metabolic networks.

It has been shown that analyzing the metabolic network *in silico* supports the identification of enzymes that are essential for the survival of an organism

(*e.g.*, [25, 54, 100, 131, 146, 167]). A general model for the metabolic network consists of alternating nodes of reactions and metabolites. Typically, using graph-based approaches, the cellular network is analyzed by removing a node in the network-model in order to mimic a specific drug treatment that inhibits the corresponding protein. A single node in the network is then characterized by estimating the robustness of the network when this node has been discarded. In this way, several computational techniques have been developed to identify essential genes *in silico*. For example, Flux balance analysis is widely used to assess the essentiality of genes [20, 55]. However, FBA approaches need clear definitions of nutrition availability and biomass production under specifically given environmental conditions [144]. The descriptors for enzymes in the metabolic network were provided by graph-based approaches and were used to identify drug targets in microorganisms; these descriptors included an identification of choke points and load points [54, 131, 167], an estimation of damaged compounds and reactions when inhibiting a possible target [100] and various descriptors for the centrality of a node in a network [1, 51, 63, 65, 129]. Furthermore, we invented a new descriptor that examines the ability of the network to obtain the products of a knocked-out reaction from its upstream substrates via alternative pathways [54]. In addition, gene sequence features like codon usage (frequency of base triplets coding for particular amino acids), GC-content (frequency of the bases guanine and cytosine) and phyletic retention (evolutionary sequence conservation), were used for predicting essential genes [64, 75, 147]. However, a single feature describing the topology and the sequence may often not yield a good essentiality estimate and intelligently combining these features can yield a far more comprehensive model.

In this thesis, we developed and applied an integrative machine learning method that combined these descriptors.

1.2 Objective and scope

The goal of this thesis is to analyze metabolic networks and to develop a machine learning technique for identifying potential drug targets in the metabolic networks of microorganisms and, in particular, pathogens. The study focuses on effecting single targets in metabolism. The main task was the investigation of metabolic network models, with respect to detecting the loss of the stabilization and robustness of the network after the removal of a component. This was followed by employing various graph-based analysis techniques, genome and gene expression analysis and a machine learning algorithm that related various characteristics of the nodes in the network with their essentiality. Genome-wide knockout screens served as the experimental data and the gold standard for evaluating the approach.

1.3 Outline of the thesis

Chapter 1 introduces the biological background and further basic topics related to this thesis; it also reviews the computational identification of drug targets and essential genes, especially graph-based analyses. Chapter 2 summarizes the methodologies and datasets applied in this thesis. Detailed descriptions of the methods and algorithms, including the machine learning technique used in this thesis, are provided. Chapter 3 reports the results of analyzing the metabolic network of knockout strains, validating the experimental screens in *Escherichia coli* (*E. coli*) and comparing the approach to Flux Balance Analysis. Additionally, the results of predicting essential genes across organisms are described for *E. coli*, *Pseudomonas*, and *Salmonella*. Chapter 4 provides the discussion and outlook.

1.4 Biological background

In this section, a brief overview of the related biological background is provided. The focus lies on essential genes and metabolism in bacteria and various biotechnological backgrounds, such as experimental knockout screens. First, I describe the concept of the process from DNA to proteins, which is essential to understanding our sequence analysis. Next, I briefly describe metabolism and its network reconstructions, which were central to our study. Thereafter, I explain the concept of antibiotic drugs in order to promote the understanding of the process of drug discovery. Finally, I explain the main idea of genome-wide knockout screens for detecting gene essentiality. For more detailed information on these topics, I refer to the standard molecular biology literature (*e.g.*, [4, 52, 118]).

1.4.1 DNA, genes and proteins

DNA stands for deoxyribonucleic acid, which is a polymer consisting of monomer units called nucleotides. Nucleotides are composed of a phosphate group, a sugar deoxyribose and a nitrogenous base. The four different bases of DNA are adenine (A), guanine (G), cytosine (C) and thymine (T). According to their chemical structures, these four bases are generally separated into two groups: purines and pyrimidines. A and G are the purine bases and form two rings, while C and T consist of one ring and are the pyrimidine bases. In a DNA molecule, these bases are connected via a sugar phosphate backbone through the 3' -hydroxyl group of a sugar and the 5'-phosphate group of the neighboring sugar. Therefore, one end of the DNA carries an unlinked hydroxyl group at the 3' position on the sugar ring (3' end), and the other end carries a free phosphate group at the 5' position on the sugar ring (5'

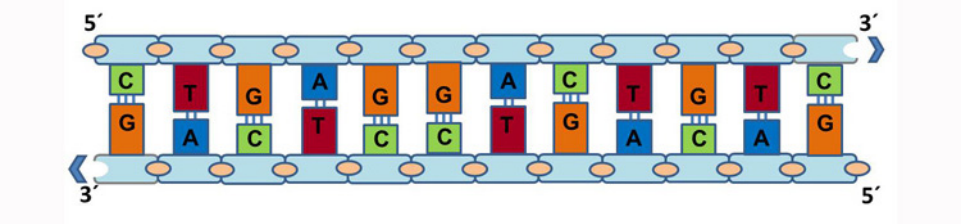

Figure 1.1: **Double-stranded DNA.** The structure of a normal DNA molecule complex is double-stranded and consists of two complementary strands. Each strand of the DNA molecule consists of nucleotides joined together by sugar-phosphate linkages in a specific manner such that A connects to T with two hydrogen bonds and C connects to G with three hydrogen bonds (A: adenine, T: thymine, C: cytosine and G: guanine).

end). It is the order of the base pairs from the 5' end to the 3' end that encodes the genetic information of a cell. Two strands of DNA are linked together by hydrogen bonds between purine bases and pyrimidine bases. A and T form two hydrogen bonds, while C and G form three hydrogen bonds. Because of this specificity of base pairing, the two strands of DNA are said to be complementary [4] (see Figure 1.1). From Figure 1.1, we see that the two strands run in opposite directions (upper: from left to right, 5' => 3'; lower: *vice versa*). The strands run antiparallel to each other. Thus, the chain in this example described from the 5' to 3' end would read C to T to G and so on. Another way to write this out is in as a condensed structural formula: 5'-CTGAGGACTGTC-3'. The sequence of the complementary strand (whose base order is 3'-GACTCCTGACAG-5') is shown here. Therefore, a DNA sequence can be considered a sequence from the four-letter alphabet A, C, G and T for information storage.

In a transcription process, the DNA sequence of a gene is used as a template to synthesize ribonucleic acid (RNA) molecules. RNA is chemically similar to DNA but contains a sugar ribose instead of a deoxyribose and the base uracil (U) instead of thymine. The RNA molecule copied from protein-coding genes is further processed into messenger RNA (mRNA), which is then translated into a protein, a long polymer chain of amino acids [4]. The information flow from DNA to proteins is shown in Figure 1.2. The genetic code in an mRNA molecule is the correspondence of three contiguous (triplet) bases, called a codon. During the process of translation, codons directly guide the insertion of a specific amino acid that is incorporated into a polypeptide chain for protein synthesis. For example, in an mRNA sequence, the codon CUG designates the amino acid *leucine*. Each codon is non-

overlapping so that each nucleotide base specifies only one amino acid or termination sequence. There are four possible nucleotide bases to be arranged into a three-base sequence codon. Therefore, there are 64 possible codons ($4^3 = 64$) encoded in a DNA sequence. Sixty-one of these codons code for the known twenty amino acids in protein; *i.e.*, a given amino acid can be specified by more than one codon. The remaining three codons act as stop signals to terminate protein synthesis. The 61 codons, which specify the 20 amino acids, and the 3 codons that lead to translation stopping can be found in Table 1.1.

The collection of DNA sequences, called a genome, comprises the genetic information that determines the structures and the functions of a cell. Genes are DNA sequences that are parts of the genome; genes can be transcribed and translated into proteins, which are macromolecules that perform functions in the cell. A large number of genomes have been successfully sequenced, and this number has been

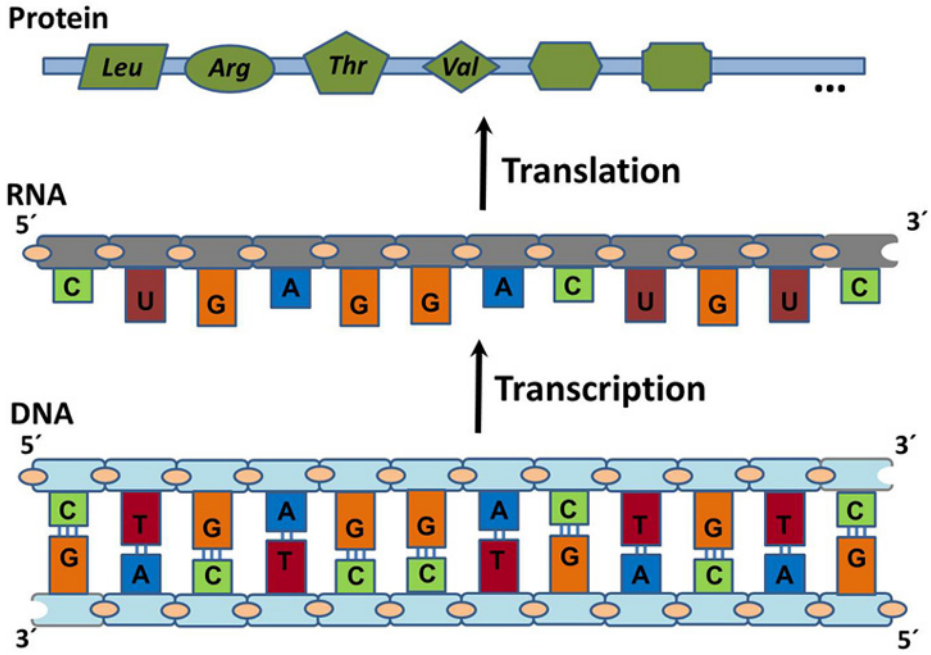

Figure 1.2: **From DNA to protein.** The genetic information is stored in the DNA and transferred to proteins. In the transcription process, the DNA sequence of a gene is used as a template to generate RNA. The RNA template is further processed and then used as a template to synthesize protein molecules in the ribosomes. These proteins are composed of amino acids transcribed from the gene sequence in which each triplet of bases (codons) encodes an amino acid.

Table 1.1: RNA codon table. The RNA codons in the table occur on the sense RNA sequence arranged at a 5' → 3' directionality.

1^{st} base	2^{nd} base							
	U		C		A		G	
U	UUU	Phenylalanine	UCU	Serine	UAU	Tyrosine	UGU	Cysteine
	UUC	Phenylalanine	UCC	Serine	UAC	Tyrosine	UGC	Cysteine
	UUA	Leucine	UCA	Serine	UAA	Stop	UGA	Stop
	UUG	Leucine	UCG	Serine	UAG	Stop	UGG	Tryptophan
C	CUU	Leucine	CCU	Proline	CAU	Histidine	CGU	Arginine
	CUC	Leucine	CCC	Proline	CAC	Histidine	CGC	Arginine
	CUA	Leucine	CCA	Proline	CAA	Glutamine	CGA	Arginine
	CUG	Leucine	CCG	Proline	CAG	Glutamine	CGG	Arginine
A	AUU	Isoleucine	ACU	Threonine	AAU	Asparagine	AGU	Serine
	AUC	Isoleucine	ACC	Threonine	AAC	Asparagine	AGC	Serine
	AUA	Isoleucine	ACA	Threonine	AAA	Lysine	AGA	Arginine
	AUG	Methionine	ACG	Threonine	AAG	Lysine	AGG	Arginine
G	GUU	Valine	GCU	Alanine	GAU	Aspartic acid	GGU	Glycine
	GUC	Valine	GCC	Alanine	GAC	Aspartic acid	GGC	Glycine
	GUA	Valine	GCA	Alanine	GAA	Glutamic acid	GGA	Glycine
	GUG	Valine	GCG	Alanine	GAG	Glutamic acid	GGG	Glycine

rapidly increasing as a result of new high-throughput sequencing techniques. Bacterial genomes can contain up to several million base pairs, and the number of genes found in bacteria ranges from 500 genes to over 5,000 genes [90].

In conclusion, genes are encoded in DNA and can be translated into proteins that perform biological functions for the growth, development and replication of the cell. While some genes are needed for the survival of a cell, other genes may only be used when responding to a particular environment. Although we have the genetic information encoded in DNA on hand for many microorganisms the challenge is how we can exploit this information in the understanding of microbial physiology and to the discovery of antibiotic drugs.

1.4.2 Cellular networks

Proteins are involved in a large variety of cellular functions comprising regulation, signal transduction and metabolism. These processes can be regarded as cellular networks that describe associations among proteins and other cell compounds. These cellular networks can conceptually be divided into three distinct parts: the metabolic network, the cell signaling network and the transcriptional regulatory network.

The metabolic network is currently the best-described cellular networks (more details described in Sections 1.4.3 and 2.3.1). Briefly, it consists of a series of biochemical reactions, which are catalyzed by enzymes (proteins that induce chemical

reactions) [4, 5, 86]. Metabolic reactions typically involve the conversions and mass flow of small molecules (*e.g.*, sugars); these reactions have been studied for several decades using enzyme kinetics and tracer experiments. In contrast to the metabolic network, the knowledge about signaling interactions is much less established on a general level, and models are often obtained from the functional context and potential wiring/rewiring aspects. The signaling network [5, 86] is a complex system of interactions between signaling molecules within the cell from receptors (proteins that receive and respond to a stimulus) to transcription factors (proteins that bind to specific DNA sequences to control transcription processes). The transcriptional regulatory network is a network model for the regulation of gene expression in which transcription factor proteins bind the regulatory DNA regions of a gene to stimulate or repress the transcription of a gene and, therefore, the production of the corresponding proteins [5, 86]. This topology of regulatory networks is less conserved and often adapts dynamically to the physiological situation [104]. A protein-protein interaction (PPI) network is a signal transduction model and usually refers to a physical interaction between proteins such as phosphorylation or binding. This term can refer to other associations of proteins, such as functional interactions, stable interactions to form a protein complex, and transient interactions, which are brief interactions that can modify a protein and can further change PPIs (*e.g.*, protein kinases). Moreover, PPI network models are often used as a simplification of more elaborate signaling networks [51, 63, 142] and as a global view of integrated cellular networks [1].

1.4.3 Metabolism and its network

Metabolism is a collection of biochemical reactions for food digestion and the maintenance of all cellular processes, in which large nutrient molecules, such as proteins, carbohydrates and fats, are broken down into smaller molecules to produce the constructing materials and components of a living cell; these processes involve energy transformation and conservation [4]. These chemical reactions are controlled by enzymes (see Figure 1.3). Enzymes are proteins that catalyze (*i.e.*, increase the rates of) chemical reactions; without them, most such reactions would not take place at a useful rate. In addition, an enzyme usually catalyzes only one of the many possible types of reactions that a specific molecule could possibly undertake. With these specific properties, enzymes can be used to regulate particular reactions. Enzymatic reactions are usually connected in a series, so that the product of one reaction becomes the substrate for the next reaction. These linear reaction pathways are, in turn, linked to each other, forming a complex system of interconnected reactions that enable the cell to survive, grow and reproduce (see Figures 1.3 and 1.4).

Metabolism is usually divided into two categories: catabolism and anabolism [4].

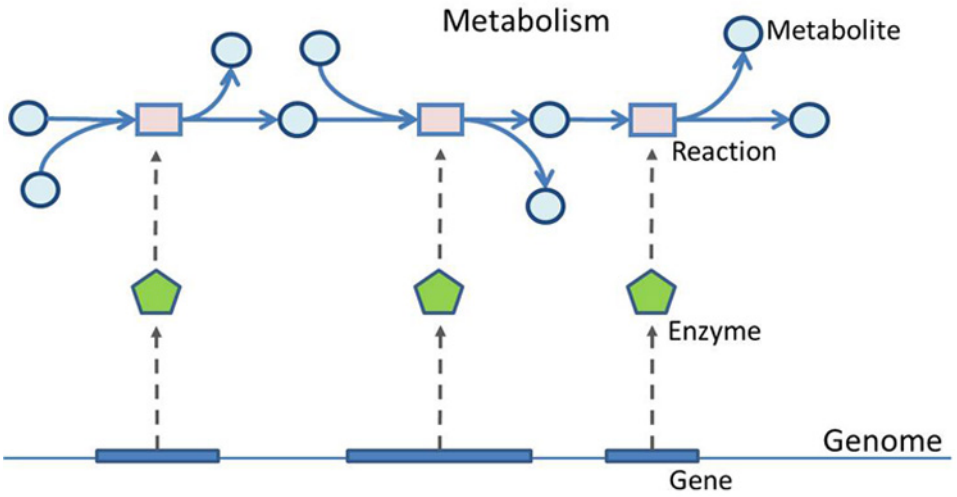

Figure 1.3: **Gene, enzyme and reaction associations in metabolism.** Genes are translated into proteins by transcription and translation processes. Then, the proteins that act as catalyzers (enzymes) control the biochemical reactions in the cell for maintenance, nutrient supply and energy production by consuming or producing metabolites.

Catabolism comprises the reaction pathways that digest food into smaller molecules, thereby creating useful materials and energy for the cell; some of those small molecules can be used as building blocks that the cell needs. Anabolism, or biosynthesis, involves the reaction pathways that use energy and small compounds to construct other needed components of the cell. Many enzymes are specific to one substrate such that the enzyme activity can be affected by inhibitors and activators. Inhibitors are molecules that decrease enzyme activity while activators are molecules that increase enzyme activity. Many drugs and toxins are enzyme inhibitors that reduce the activity of important reaction pathways involved in the cell's survival [37, 68]. There are other factors, such as temperature, chemical environment (*e.g.*, pH) and the concentration of substrates in the medium, that can affect the activity of enzymes. A variety of enzymes are utilized in medicine and industry. For example, some enzymes are used for the synthesis of antibiotics, and some enzymes are used in natural cleaning powders to break down protein or fat stains on clothes.

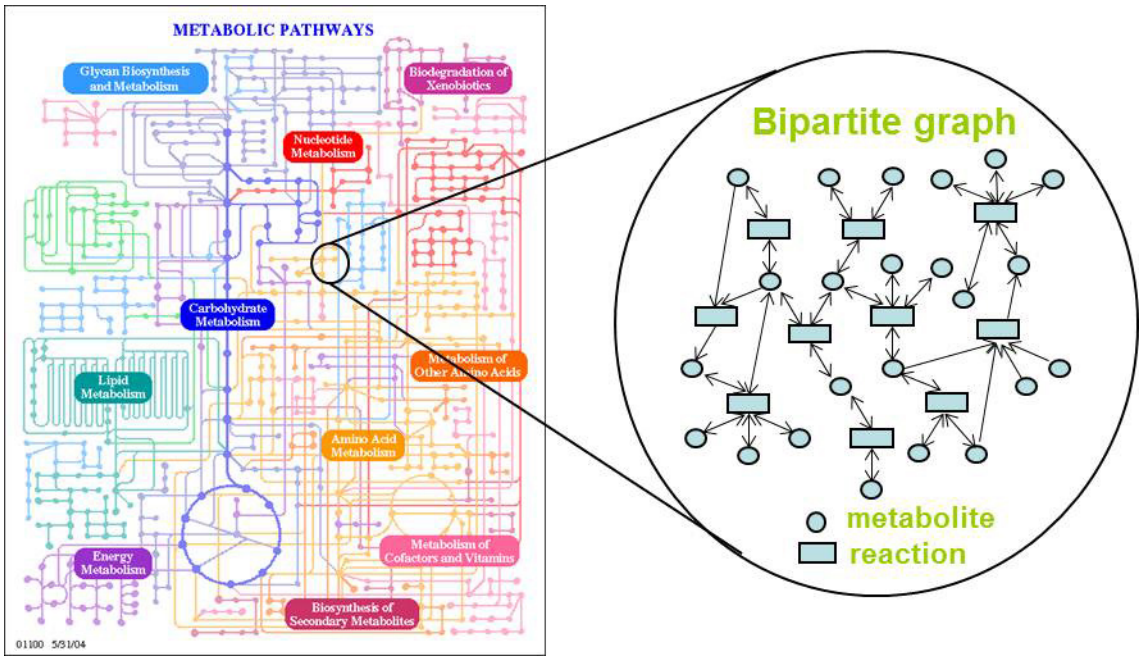

Figure 1.4: **Metabolic pathways in a cell.** The complex system of interconnected metabolic pathways consists of thousands of metabolic reactions and is linked by alternating nodes of reactions and metabolites. The depicted overview and more details of the metabolism for each organism can be found in the KEGG Pathway Database at www.genome.jp/kegg.

1.4.4 Treatment of bacterial infection with antibiotic drugs

Antibiotics are drugs that treat various infections caused by bacteria [10]. In general, these drugs are designed to kill pathogenic bacteria without harming the host organism. The effectiveness of an antibiotic treatment depends on several factors including the immune system of the host, the infection location, drug dispersion and concentration, and the resistance factors of the bacteria [10, 135]. Most antibiotics, such as *vancomycin* and *penicillin*, target the bacteria cell walls. Some interrupt protein synthesis (such as *erythromycin* and *tetracycline*), and some interfere with DNA replication, such as *quinolones* [37].

Antibiotics have been incredibly effective in treating and controlling bacterial infections. Many people have been saved, and morbidity has considerably decreased. However, antibiotic resistance has steadily increased in the last twenty years [16, 47]. Bacteria may be naturally resistant to different classes of antibiotics or may acquire resistance from other bacteria through the transfer of resistant genes. Antibiotic-resistant organisms lead to increased hospitalizations, health costs and mortality [8,

9, 13, 26, 27]. Thus, the discovery and development of new classes of antibiotics to treat these multi-resistant organisms has become one of the main challenges in the pharmaceutical industry.

1.4.5 Genome-wide knockout screens for detecting gene essentiality

With the availability of complete genomic sequence data, the systematic evaluation of the essentiality of each gene in a genome has become possible [52, 118]. The majority of the techniques used for these evaluations of global gene essentiality in biological experiments are focused on the growth of mutant strains; gene essentiality is deduced from the inability of the mutant cell to achieve a certain number of divisions. The approaches for identifying essential genes in the wet lab include several types of classical forward genetic screens and systematic targeted gene knockout [52].

Genetic footprinting
Genetic footprinting is a well-known technique to distinguish between essential and non-essential genes and was first developed in yeast. Later it was applied to bacteria, such as *E. coli* [60]. It employs transposon mutagenesis, the outgrowth of the mutagenized cell population and the analysis of the fate of cells carrying mutations in specific genes. Transposons (gene sequences that can be introduced and exchanged in the genome) of defined length are randomly integrated in the whole genome and therefore disrupt gene function. In one representative study of the identification of essential genes, half of the mutagenized population was grown in rich medium (Lysogeny Broth (LB)), containing additional vitamins and micronutrients. Fifty percent of mutagenzied cells were immediately frozen. After outgrowing the cells, the transposons were detected by nested polymerase chain reactions (PCR) and compared with the transposons in the frozen population. The regions where the transposons were absent in outgrown cells were considered essential because no cell survived if this region was disrupted. It is worth noting that footprinting technology does have some limitations, including the difficulty of assessing the essentiality of small genes (<400 bp) due to the lower number of transposition events per gene, the inability to assess duplicated genes or genes with functional paralogs, and the occurrence of regions where transpositions hardly occur [118].

Systematic targeted gene knockout
Systematic targeted gene knockout was developed to create single knockout mutants to investigate the effects of the loss of one gene, such as in the study *E. coli* by Baba *et al* [12]. The principal strategy is based on the method of one-step inactivation of chromosomal genes. To knock out a gene, the replacement for a target

gene with a kanamycin-resistant marker was generated by polymerase chain reaction (PCR) using oligonucleotide DNA primers homologous to the gene flanking regions. The start-codon and the up-stream translational signal were not replaced and fully intact. Inframe single-gene deletions were verified by PCR with kanamycin and loci specific primers. When cells were unable to create a mutant that formed colonies on a plate, the mutated gene was considered to be essential. The advantages of this method are the complete deletion of an entire open reading frame (gene) and the fact that the precise design reduces polar effects (non-integration of the replacement gene) for the downstream genes of the chromosome.

In conclusion, both methods (genetic footprinting and targeted gene knockout) are well-established and commonly used for genome-wide knockout screens. Genetic footprinting gives an idea about which genes are essential for vital growth; in contrast, deletion studies reveal lethal mutants more comprehensively [59].

1.5 Existing computational approaches

Considering the experimental constraints described above (in Sections 1 and 1.4.5), the development of a computational approach for predicting gene essentiality has become an important challenge in drug target identification. In this section, various existing approaches for computationally identifying potential drug targets are reviewed. Some of these predictors have been developed using the sequence features of genes with or without homology comparison [64, 147]. In addition, these predictors of gene essentiality have been developed based on the network topology features of a protein in a protein-protein interaction network [1, 51, 63, 81] or knocked-out enzymes from the metabolic network according to gene deletions [20, 55]. I first explain existing approaches using genomic data, especially those based on gene and protein sequences. Next, I briefly describe approaches using protein-protein interaction networks. Finally, I review various methods that analyze the topology of the metabolic network for detecting drug targets, such as choke points, damages and flux balance analysis.

1.5.1 Finding drug targets through the analysis of genomic data

The availability of complete genome sequences for many organisms has enabled the discovery of essential genes in several organisms by computational methods [64, 112, 147]. Some computational methods attempt to determine the "minimal gene set," *i.e.*, the set of core essential genes for cellular life [112]. To find the mini-

mal gene set, Mushegian and Koonin [112] performed comparative genome analysis based on the notion that conserved genes between species (so-called orthologous genes) are more likely to be essential [112]. The first two completely sequenced bacterial genomes, *Haemophilus influenzae* and *Mycoplasma genitalium*, were analyzed, and 256 genes that were orthologous between those bacteria were reported as an approximation of a minimal gene set for bacterial life [112]. After studying *Bacillus subtilis* and *E. coli* by inferring gene essentiality using homologs, Rocha and Danchin also reported that essential genes tend to be more conserved and essentiality may play a fundamental role in the distribution of genes in most bacterial genomes. Therefore, the identification of orthologous genes across species has become a major source for inferring gene essentiality.

Gustafson *et al.* [64] reported constructing a classifier of essential genes, by exploiting genomic features derived from the sequence data of *E. coli* and the yeast *Saccharomyces cerevisiae* (*S. cerevisiae*). They defined the "phyletic retention" feature, which is the number of other organisms in which a gene is conserved (the presence of an orthologous gene in other organisms) and found that this feature is a good indicator of gene essentiality. However, deriving the phyletic relationship of genes across organisms requires the attentive selection of related reference genomes. Characteristic sequence features, such as codon usage, codon adaptation, GC content and overall hydrophobicity, were used to train a classifier for predicting essential genes in fungal genomes in the study of Seringhaus *et al.* [147]. The classifier was developed using data from *S. cerevisiae* and was tested on the closely-related yeast species *Saccharomyces mikatae*. The predicted essential genes were verified by assessing their homology to the essential genes in *S. cerevisiae*, and some of those genes were tested experimentally. Although most of the studies were performed in the context of a single organism, this study demonstrates the ability to predict essential genes using machine learning based on genomic data, which can be applied to other novel organisms.

1.5.2 Finding essential genes in protein interaction networks

In recent years, high-throughput methods have been increasingly used to identify protein-protein interactions on the genomic scale resulting in interaction maps for entire organisms [1, 51, 63, 129]. Various descriptors of the centrality of a node in a network, such as connectivity and betweenness centrality, have been successfully applied in the detection of essential proteins in protein-protein interaction networks [1, 51, 63, 64, 65, 81]. For example, large-scale protein-protein interactions of yeast have revealed that "hub proteins" (proteins with high connectivity) are

more likely to be essential and evolve at a slower rate [83]. Batada *et al.* [17] used yeast protein interaction data that had been carefully collected from the literature. They found out that hub proteins are more likely to be essential, but they did not find that hub proteins are correlated to slow evolutionary rates [17]. Chen and Xu [34] studied the characteristics of essential genes in yeast, by integrating genomic and high-throughput experimental data from gene expression and protein-protein interaction networks. They showed that essential genes were correlated with evolutionary rates, interaction connectivity, gene duplication rates (a measure of gene homologs within the same species) and gene-expression cooperativity (a measure of gene co-expression) [34].

Although protein-protein interaction networks may provide a global view of cellular mechanisms, the biochemical implications of high-throughput interactions are not always obvious [74]. Therefore, we were interested in identifying drug targets in pathogens inferred from the properties of a mal-functional metabolism after having knocked out an enzymatic function because metabolism is the best-described cellular network and essential for responding to environmental constraints, maintaining the structure of a cell, and participating in cell growth and reproduction (see Section 1.4.3).

1.5.3 Analysis of metabolic networks

A general model for the metabolic network has been described by graph-based approaches and was applied to identify drug targets in pathogenic organisms [54, 55, 100, 131, 146, 167]. Several computational techniques have been developed to identify essential genes *in silico*. The concepts of choke points and load points were successfully applied to estimate the essentiality of an enzyme [54, 131, 167]. The term 'damage' was used to assess enzymes that may serve as drug targets when their inhibition influences a substantial number of downstream reactions and products [100]. In addition, one of the most widely-used techniques to assess gene essentiality in genome-scale metabolic network is flux balance analysis [20, 55], which considers mass balance analysis and other constraints with optimality conditions to predict steady-state reaction rates.

Evaluation of choke points and load points
In metabolic networks, Samal *et al.* found that most reactions identified as essential are reactions that are involved in the consumption or production of metabolites with low connectivity [138]. This is because they are more likely to be the limiting factor for consuming or producing these metabolites. In the extreme case, they uniquely consume or produce a certain compound in metabolic networks. Blocking these reactions may cause cell death through the accumulation of a large amount of toxic

compounds or the lack of important compounds. Rahman and Schomburg defined reactions with this property as 'choke points' [131]. This technique has been successfully applied to identify drug targets for *Plasmodium* [25, 167] and many other organisms through the use of their web-based Pathway Hunter Tool [130]. The website also provides an analysis of the shortest paths among metabolites and proposes the computational identification of potential drug targets by calculating the load scores of an enzyme in the network [131]. Thus, they define 'load points' as hot spots in the metabolic network (enzymes/metabolites) with high load scores based on the ratio of the number of shortest paths passing through a metabolite/enzyme (in/out), and the number of nearest neighbor links (in/out) attached to it. This ratio was compared to the average load value in the network [131]. Therefore, reactions with high load scores are more likely to be potential drug targets. The load score is a good measure of possible fluxes passing through a reaction. However, this technique does not consider the possibility of alternative pathways after removing a predicted essential reaction with a high load. When knocking out a reaction, the mutated network may use other reactions to achieve communication in cells. Therefore, a consideration of deviations, such as alternative pathways and the possibility of yielding some downstream metabolites, should be taken into account [54] as we explained in Section 2.4.3.

Estimation of metabolic damages

Lemke *et al.* proposed the term 'damage' which was used to assess enzymes that may serve as drug targets when their inhibition influences a substantial number of downstream metabolic reactions and products [100]. We implemented this term with generalization by counting possible damaged compounds and reactions when there was no possible alternative pathway by which to reach the compounds and reactions (see Section 2.4.2).

Flux balance analysis

Flux balance analysis (FBA) is a constraint-based approach for quantitatively analyzing the flow of metabolites through a metabolic network. Thereby, it is possible to predict the growth rate of an organism or the rate of production of an important metabolite [20, 49, 117]. The flux balance constraints are based on the assumption that the total amount of any inner compound in the cell that is produced must be equal to the total amount being consumed at the steady state. Allowable fluxes of any reaction are bounded as the maximum and minimum fluxes, and this is taken from experimental data from, *e.g.*, enzymatic assays. These flux balances and bounds define the space of the feasible flux distributions of a system. Fluxes represent the rates at which every metabolite is consumed or produced by the corresponding reactions. To optimize the fluxes out of these given constraints, a

biological objective function is defined. For example, in the case of microorganisms aiming for maximal growth, the objective is biomass production, which is the rate at which metabolic compounds are converted into biomass constituents, such as nucleic acids, proteins and lipids. Mathematically, an "objective function" is used to quantitatively define how much each reaction contributes to the phenotype. This is mathematically formulated as a system of linear equations. In flux balance analysis, these equations are solved using linear programming (see Section 2.4.4). By simulating the whole reconstructed metabolic network of an organism of interest, we obtain the wildtype growth rate under specific flux bounds and metabolic constraints (or conditions). When performing a single gene or reaction deletion under the same conditions by limiting its corresponding fluxes to zero (so-called knockout simulation), a mutant's growth rate is measured and compared to the wildtype's. A knocked-out gene or reaction is predicted as essential under the given condition if the mutant model yields much lower biomass production in comparison to the wildtype. Flux balance analysis is a widely-used and well-established method for assessing the essentiality of genes [20, 49, 55, 117]. For example, analyzing flux balances under the conditions of aerobic glucose (by limiting the glucose uptake rate) using the COBRA toolbox [20] and a newly reconstructed metabolic network of *E. coli* yielded 92% accuracy when predicting the essentiality of genes [55] under aerobic glucose conditions and yielded 88% accuracy for rich nutrient conditions. However, FBA approaches need clear definitions of nutrition availability and biomass production under specifically given environmental conditions, and it is difficult to characterize the uptake rates for each compound of a rich medium, especially for situations like the gut of a host of intestinal pathogens (for a good overview of these aspects see [56, 144]).

1.6 Main contributions of this thesis

In the following I summarize the main contributions of this thesis:

- **Analysis of metabolic networks**
  We developed an algorithm to examine the ability of the metabolic network to obtain the products of a knocked-out reaction from its substrates via alternative pathways. Basically, each reaction in the network was deleted (knocked out *in silico*), respectively. A breadth-first search algorithm tested whether the neighboring compounds of the knocked-out reaction could be produced by other reactions and pathways of reactions. With this approach, we tested whether deviations in the network could be used to replace the knocked-out reaction (see this method in Section 2.4.3 and the results in Section 3.1).

This was successfully applied to detect potential drug targets for *Plasmodium falciparum* [54] and used as one of our descriptors in our other investigations. This method was invented by us and reported for the first time in our article [54]. Furthermore, other descriptors based on metabolic networks, genomic data and transcriptomic data have been analyzed and examined for their potential to identify drug targets, and these are described in Sections 2.4 and 2.5.

- **Machine learning based approach to integrate the descriptors**
  In this thesis, we developed a workflow for a machine learning method that integrates a large variety of different descriptors to identify drug targets. First, the metabolic network was constructed using various qualitative and quantitative information from public databases and the literature (see Sections 2.2 and 2.3). With the technique of machine learning (explained in Section 2.7), a large set of features (explained in Sections 2.4 and 2.5) was integrated and used for a classification of gene essentiality. Finally, the results showed that our methods can be used to detect potential drug targets in pathogens and that these methods are feasible for validating experimental knockout data (see Sections 3.2.3 and 3.2.4). With this newly-developed, integrated approach, we showed that using a machine learning based approach made it possible to achieve 79% sensitivity and 97% specificity, which were comparable to those achieved by flux balance analyses (sensitivity: 51%, specificity: 97%, see Section 3.2.2). It is worth noting that, in contrast to FBA, our approach does not depend on any additional (in addition to the essentiality data serving as the gold standard) experimental information or elaborate literature study. Furthermore, we show that the method can be used to predict the essential genes of a query organism using the experimental information about essentiality from a related bacterial reference organism (see Section 3.3.1).

The results of our research have been published in a peer-reviewed conference proceedings article [128] and two original journal articles [125, 127] in a journal with a good reputation in our field of systems biology. Additionally, we described our approach in a book chapter [126]. The developed approach was also used in other related projects [53, 54].

Chapter 2

Methods

To integrate a variety of information for the purpose of gaining insight into the essentiality of a gene or protein, topology descriptors of metabolic networks, genomic data and transcriptomic data have been assembled for a machine learning approach. Our approach is based on a collection of methods from the areas of network analysis and machine learning. This chapter first summarizes the general workflow in Section 2.1. An explanation of the data, including the metabolic networks and knockout screens that we used, is given in Section 2.2. The construction of the network is addressed in Section 2.3, followed by the extraction of network descriptors, such as deviations and flux balance analysis features, which is given in Section 2.4. Genomic and transcriptomic analysis features are explained in Section 2.5, including homology analysis and gene expression analysis. Preprocessing and feature evaluation are explained in Section 2.6. Our classification method and learning techniques are described in Section 2.7. Finally, performance measures are explained in Section 2.8.

2.1 General workflow

An overview of our workflow is shown in Figure 2.1. First, the metabolic networks were constructed for the organisms that were investigated with biochemical reactions from public databases. For each gene, the features of the gene or the corresponding reaction were calculated to describe its topology in the metabolic network and its genomic and transcriptomic relations. These features were then normalized and statistically analyzed by comparing them to essentiality classes (used as a gold standard) taken from experimental genome-wide knockout screens. Next, Support

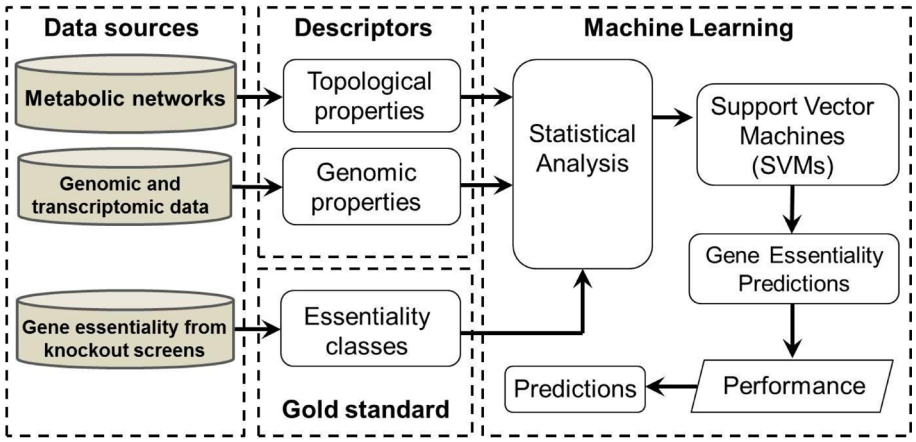

Figure 2.1: **The workflow**. The workflow for the prediction of essential genes by integrating network and genomic information using Support Vector Machines.

Vector Machines (SVMs) were trained based on the features to distinguish between essential and non-essential genes. The trained machines were evaluated and then used as a prediction model for gene essentiality. This model was then applied to identify potential drug targets and to predict new query genes.

2.2 Data sources

2.2.1 Lists of essential genes from knockout experiments

The list of the essential genes for microorganisms was downloaded from the National Microbial Pathogen Database Resource (NMPDR [108], www.nmpdr.org) and the literature [12, 60, 79, 93, 103]. In this thesis, we used information about essential genes from the knockout screens of well-studied organisms, *Escherichia coli* (*E. coli*) and *Pseudomonas aeruginosa* (*P. aeruginosa*), to evaluate the performance of our methods. The knockout screen of *Salmonella typhimurium* (*S. typhimurium*) was used to evaluate the prediction of potential drug targets. Table 2.1 summarizes the total numbers of tested genes, essential genes and non-essential genes for each organism and the source from which we obtained the data.

Experimental knockout screens of *E. coli*
E. coli is one of the most commonly studied organisms. Many experimental screens have been performed to test which genes are dispensable for *E. coli* under different conditions (such as different nutritional media). Recently, two knockout screens of

Table 2.1: Numbers of essential genes from the knockout experimental screens

Genome	Experimental condition	Total	N	E	U	Reference
E. coli	rich LB medium	4,390	3,985	303	102	Baba *et al.* [12]
E. coli	glucose minimal medium*	4,390	3,866	412	102	Baba *et al.* [12]
E. coli	rich LB medium	4,308	3,126	620	562	Gerdes *et al.* [60]
P. aeruginosa	rich LB medium	5,570	4,783	787	0	Jacobs *et al.* [79]
P. aeruginosa	rich LB medium	5,688	4,469	335	884	Liberati *et al.* [103]
S. typhimurium	rich LB medium	4,425	n/a	257	n/a	Knuth *et al.* [93]

N: Nonessential, E: Essential, U: Undetermined and n/a: not available
* This dataset was used to test the flux balance analysis.

E. coli have been reported by Baba *et al.* [12] and Gerdes *et al.* [60]. The collection of knockout mutants from the studies of Baba *et al.* is known as the "KEIO collection", and we will refer to this term again. The knockout experiments were performed in rich medium and in glucose minimal medium, resulting in two datasets (denoted as rich medium and glucose minimal medium). For the rich medium, out of the 4,288 tested genes, 303 genes were identified as having no living mutants, and these genes were defined as essential. Genes that were considered to be essential under the rich medium condition were also considered to be essential under the glucose minimal medium condition. In addition to these genes, 119 genes were designated essential in glucose minimal medium because they showed very slow growth in minimal media. Gerdes *et al.* performed random transposon insertions with population outgrowth on rich medium. They found 620 essential genes and 3,126 non-essential genes.

Experimental knockout screens of *P. aeruginosa*

For *P. aeruginosa*, we used the data of Jacobs *et al.* [79] and Liberati *et al.* [103]. Jacobs *et al.* created a library of transposon insertion mutants with clonal outgrowth on rich medium at room temperature. Approximately 12% of the predicted genes of this organism lacked insertions. Many of these genes are likely to be essential for growth on rich medium. They defined 787 essential genes and 4,783 non-essential genes. Using another strain of *P. aeruginosa*, Liberati *et al.* created a non-redundant library of transposon insertion mutants. They used different transposons to create the two libraries, which accurately define essential genes. Finally, they defined 335 essential genes and 4,469 non-essential genes.

Experimental knockout screens of *S. typhimurium*

The experimental dataset for *S. typhimurium* was from a study of Knuth *et al.* [93] and was based on insertion-duplication mutagenesis (IDM). Small, randomly generated genomic fragments were cloned into a conditionally replicating vector, and the resulting library of single *S. typhimurium* clones was grown under permissive con-

ditions. Upon switching to non-permissive temperatures, discrimination between lethal and non-lethal insertions following homologous recombination allowed the trapping of genes with essential functions. With this method, genes were detected that were indispensable for growth. However, a comprehensive classification of non-essential genes could not be determined. A total of 257 genes were found to be essential, 53 genes of which coded for enzymes.

2.2.2 Metabolic network databases

The knowledge about enzymes, reactions, compounds and pathways that was used to construct metabolic networks of organisms was obtained from the database of the Kyoto Encyclopedia of Genes and Genomes (KEGG [87, 115], www.genome.jp/kegg) and the BioCyc Database Collection ([32], http://biocyc.org). To perform flux balance analysis with a curated metabolic model of *E. coli* 'iAF1260' [55], the metabolic network of the Biochemical Genetic and Genomic (BiGG [141], http://bigg.ucsd.edu/) was used.

KEGG is a pathway database that consists of a comprehensive set of biochemical pathways for the systematic analysis of gene functions. It includes most of the known metabolic pathways and many of the known regulatory pathways. Similarly, BioCyc is a collection of genome/pathway databases. Each database contains the comprehensive genome and metabolic pathways of a single organism. For example, EcoCyc is a database for the bacterium *E. coli* with literature-based curation of the entire genome and of transcriptional regulation, transporters and metabolic pathways. BiGG is a resource of various published genome-scale metabolic network models with standard nomenclature for analyzing the metabolic capabilities of organisms using Constraint-Based Reconstruction and Analysis (COBRA) tool [20].

2.3 Definitions and construction of metabolic networks

Several properties and definitions of graphs can be used to explain our metabolic networks. Thus, in this section, definitions of the graphs that can be applied to define a mathematical representation of metabolic networks are first described. Some basic properties of graphs are also explained. Finally, characteristics of a graph can be explained in terms of the node degree distribution, which usually follows a power-law distribution in most real-world networks.

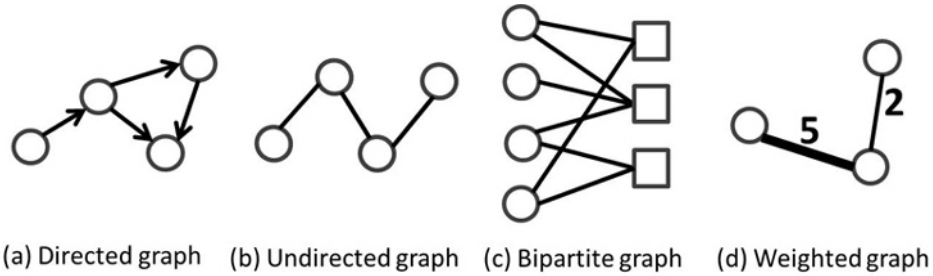

Figure 2.2: **Different types of graphs**. (a) A directed graph contains directed edges starting from one node pointing to another node. (b) An undirected graph has edges connecting two nodes without directionality. (c) A bipartite graph consists of edges connecting nodes from two different sets. (d) A weighted graph has weights for each edge.

2.3.1 Definition of graphs

In mathematical terms, a network is a *graph*. We use the term "graph" to refer to the mathematical concept of a set of vertices connected by links called edges, while the term "network" is used to refer to an application of a graph that explains an interconnection of entities such as, in our case, a metabolic network. We use the term "node" in an application network to refer to a vertex in the graph.

A **graph** $G = (V, E)$ consists of vertices $v \in V$ and edges $e \in E$ that connect those vertices, where $E \subseteq V \times V$. In a directed graph, each edge $e \in E$ is an ordered pair of vertices $e = (u, v)$, with $u, v \in V$, such that e consists of the starting vertex u and the terminating vertex v. The edges in a directed graph are depicted by arrows. In the case of an undirected graph, an edge $e \in E$ is represented by an unordered pair of nodes $\{u, v\}$, and it is depicted by a line between vertices u and v (see Figure 2.2). Undirected graphs are used if information about the direction is lacking or is not needed. In other words, in an undirected graph, there is no direction associated with an edge. Bidirectionality between vertices u and v can be represented by two edges, one leading from u to v and one in the opposite direction. In the following, if not specified, the definitions are applied to both the undirected and directed cases.

A graph is **bipartite** if the edges connect between the vertices of different sets. In a bipartite graph, the set of vertices V consists of two disjoint sets of vertices V_1 and V_2:

$$\begin{aligned} V_1, V_2 \subset V : (V_1 \cap V_2 = \emptyset) \wedge (V_1 \cup V_2 = V), \\ \forall (u, v) \in E : (u \in V_1 \wedge v \in V_2) \vee (u \in V_2 \wedge v \in V_1). \end{aligned} \tag{2.1}$$

A **weighted graph** $G = (V, E, W)$ has a set of vertices, V, and a set of edges, E. W represents the corresponding weights of those edges. **Unweighted graphs** are a special case of weighted graphs, with all of the weights set to 1 (see Figure 2.2).

Neighboring vertices and **neighborhood**: In a given graph $G = (V, E)$, two vertices u and $v \in V$ are said to be neighbors, or adjacent vertices, if $\{u, v\} \in E$. The neighborhood of a vertex v ($N(v)$) is a set of neighbors that is defined as the following:

$$N(v) = \{u \in V | \{u, v\} \in E\}. \tag{2.2}$$

For a set $S \subseteq V$, the neighborhood is defined to be

$$N(S) = \bigcup_{v \in S} N(v). \tag{2.3}$$

If G is directed, then we can distinguish between the incoming neighbors of v (those vertices $u \in V$ such that $(u, v) \in E$) and the outgoing neighbors of v (those vertices $u \in V$ such that $(v, u) \in E$) as

$$\begin{aligned} N_{in}(v) &= \{u \in V | \exists (u, v) \in E\} \\ N_{out}(v) &= \{w \in V | \exists (v, w) \in E\}. \end{aligned} \tag{2.4}$$

In this case, the neighborhood of vertex v is then defined as $N(v) = N_{in}(v) \cup N_{out}v$.

Paths and **path length**: Let u and v be two vertices in a graph G. A path from u to v, *path*(u,v), is a sequence of vertices where each vertex is a neighbor of the next vertex,

$$u = v_1, v_2, ..., v_b = v, \tag{2.5}$$

such that for $i = 1, ..., b - 1$:

$$\begin{aligned} &(i) \quad \{v_i, v_{i+1}\} \in E \quad \text{for the undirected case } \textit{or} \\ &\qquad (v_i, v_{i+1}) \in E \quad \text{for the directed case;} \\ &(ii) \quad v_i \neq v_j \text{ for } i \neq j \end{aligned}$$

where b is the number of vertices in the path and the length of this path is defined as the number of edges in the path, which is $b - 1$ (see Figure 2.3). The length of the shortest path from u to v, $|\text{path}(u, v)|$, is called the geodesic distance which is simply called "distance". The diameter is defined as the maximum value of the shortest paths taken over all of the pairs of vertices in V [169].

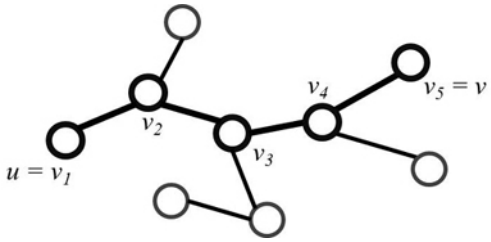

Figure 2.3: **An example of a path length**. A path from u to v in the graph is from v_1 to v_2, v_3, v_4, and v_5. Thus, the path length of this path is 4 ($|\text{path}(u, v)| = 4$).

2.3.2 Graph representations

Graphs can be represented as adjacency matrices that denote which vertices of the graph are adjacent to which other vertices.

Adjacency matrix of a simple graph: The adjacency matrix A of a weighted graph $G = (V, E, W)$ is an $N \times N$ matrix such that

$$A_{ij} = w_{ij} \text{ if } (i,j) \in E,\ 0 \text{ otherwise} \\ \text{for } i \text{ and } j \in 1, ..., N \tag{2.6}$$

where $N = \|V\|$, the number of vertices in V. The adjacency for an unweighted graph simply replaces w_{ij} with 1. Note that the adjacency matrix is symmetric in the undirected case.

Adjacency matrix of a bipartite graph: The adjacency matrix A of a weighted bipartite graph $G = (V, E, W)$ with $V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset$ is an $N_1 \times N_2$ matrix such that

$$A_{ij} = w_{ij} \text{ if } (i,j) \in E,\ 0 \text{ otherwise} \\ \text{for } i \in 1, ..., N_1 \text{ and } j \in 1, ..., N_2 \tag{2.7}$$

where $N_1 = \|V_1\|$ and $N_2 = \|V_2\|$. The adjacency for an unweighted bipartite graph simply replaces w_{ij} by 1.

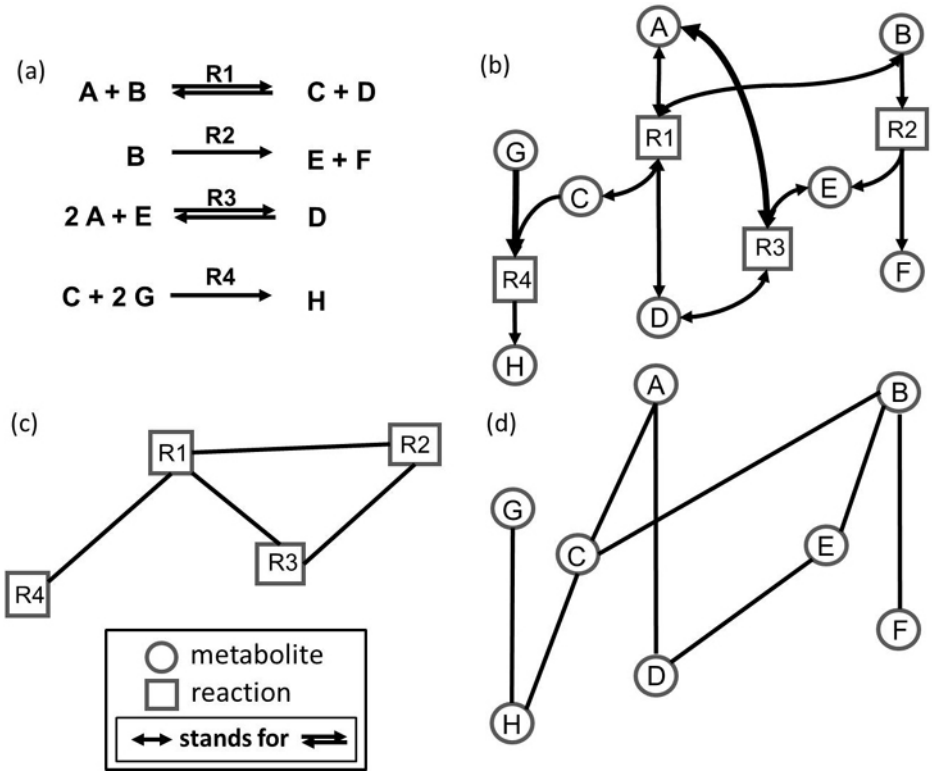

Figure 2.4: **Representations of the metabolic network**. Four representations of metabolic reactions are shown: (a) typical stoichiometric equations, (b) a bipartite graph with rectangles as reactions (or enzymes) and circles as metabolic compounds (c) a reaction-based representation of the metabolic network, and (d) a metabolite-based representation. Note that in (c) and (d) the direction of the edges are not taken into account.

Metabolic network representation

Metabolic networks are often represented as directed bipartite graphs (see Figure 2.4) that consist of two disjoint sets of vertices representing metabolites and reactions [94]. The directions of the edges in the metabolic networks are given by the relationship between the substrate and product of the biochemical reactions. An edge indicates that a metabolite is either a substrate or product of a reaction. The distinction between the substrates and products of a reaction is only possible if the graph is directed, *i.e.*, if the set of edges E consists of ordered pairs of vertices. This distinction is often useful when modeling metabolic fluxes [15]. For some applications, a reaction-based representation is needed in which the vertices of the network are the reactions, and the edges are present if a product of one reaction

is the substrate of the other. Similarly, in a metabolite-based representation, the vertices are the metabolites that are connected by reactions (see Figure 2.4). As a bipartite graph, the metabolic network can be represented as an adjacency matrix of $m \times n$ dimensions, where m is the number of metabolites and n is the number of reactions. More exact models of metabolic networks can be represented by adjacency matrices with weights for stoichiometric coefficients. The small example network of Figure 2.4 consists of four reactions and seven metabolites,

$$\begin{array}{lrcl} \text{R1:} & \text{A} + \text{B} & \rightleftharpoons & \text{C} + \text{D} \\ \text{R2:} & \text{B} & \rightarrow & \text{E} + \text{F} \\ \text{R3:} & 2\,\text{A} + \text{E} & \rightarrow & \text{D} \\ \text{R4:} & \text{C} + 2\,\text{G} & \rightleftharpoons & \text{H} \end{array}$$

The stoichiometric matrix, which is the adjacency matrix that contains stoichiometric coefficients of each reaction equation, is then given by the following:

$$S = \begin{bmatrix} -1 & 0 & -2 & 0 \\ -1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

where the rows correspond to metabolites A, B, C, D, E, F, G and H, respectively, and the columns correspond to reactions R1, R2, R3 and R4, respectively. Until now, no optimal and standardized method exists for the reconstruction of a cellular metabolic network [56, 95]. However, ubiquitous metabolites, such as water, oxygen, ATP and co-factors are often discarded to model only the most relevant metabolic fluxes [54, 94, 127].

2.3.3 Degree distributions and power laws

To understand network architecture and robustness, network topology properties can be considered. One of the most commonly used topological features in graphs is the *degree* (or connectivity) [169].

Degree

In undirected graphs, the **degree** k of a vertex $v \in V$ is defined as the number of edges between v and its adjacent vertices, which is the number of vertices in

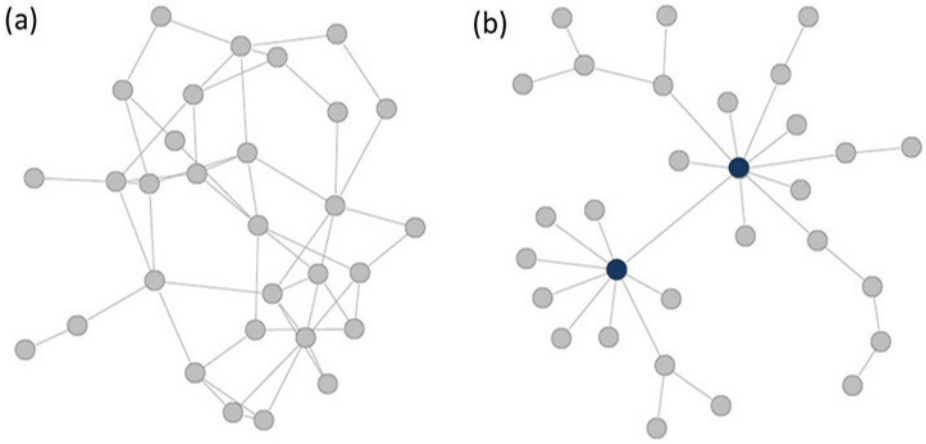

Figure 2.5: **Random and scale-free networks**. (a) The depicted random network contains 30 nodes with randomly chosen edges, according to a Poisson distribution. (b) The depicted scale-free network contains the same number of nodes, with a power-law distribution in which a few nodes have high connectivities (so-called hubs, shown in dark blue) and many have low connectivities.

the neighborhood set of v, $|Neighborhood(v)|$. A vertex with many connections has a higher degree, which reflects its importance in the network [65, 81, 169]. The degree for directed graphs can be divided into the **in-degree** and **out-degree** (see neighborhood in Section 2.3.1). The in-degree and out-degree of a vertex $v \in V$ in a directed graph is defined as the number of incoming neighbors and outgoing neighbors (as defined in Equation (2.4)), respectively:

$$\begin{aligned} d_{in}(v) &= |N_{in}(v)| \\ d_{out}(v) &= |N_{out}(v)| \end{aligned} \tag{2.9}$$

For metabolic networks as bipartite graphs, the in-degree of a reaction (node) can represent the number of its substrates, whereas the out-degree represents the number of its products.

Power-law distributions

Structures of graphs can be distinguished by their degree distribution. For example, a lattice grid has a simple degree distribution [11, 15]; all of the inner vertices have an identical degree, which is four for a square lattice. Erdös and Rényi [50] pointed out that the connectivity of simple random graphs follows a Poisson distribution. As defined by Erdös and Rényi [50], the traditional random network is an undirected graph with n vertices and randomly selected edges. There are $\frac{1}{2}n(n-1)$ possible

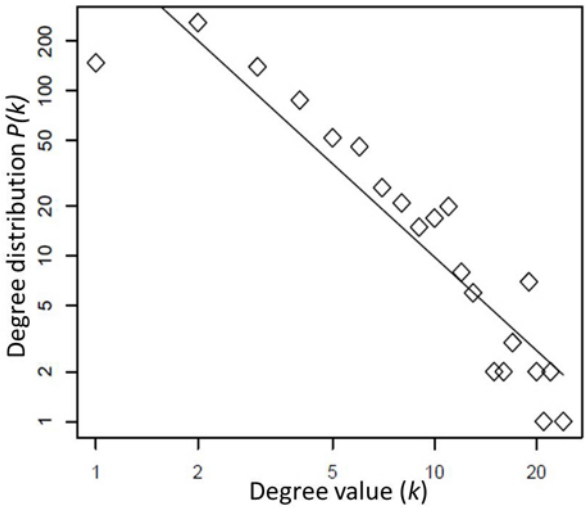

Figure 2.6: **An example of a power-law distribution** The degree distribution for the metabolic network of *E. coli* fitted with a log-log linear regression. The degree (k) of a reaction is the number of its next nearest neighbors in the network.

edges of these vertices. By choosing edges at random, each of the possible edges has the same independent probability p to be chosen. Thus, a single vertex in the random graph is connected to any of the other remaining $n-1$ vertices with the same probability p. Therefore, the probability of the specific vertex having the degree k, p_k, forms a binomial distribution [114]:

$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k} \tag{2.10}$$

For a large graph with large numbers of vertices, this binomial distribution becomes a Poisson distribution with the same mean. However, many degree distributions in the real-world follow power laws [15]. Let p_k be the fraction of vertices with degree k. Then, p_k is the probability that a randomly chosen vertex has degree k, $p_k = n_k/n$, where n_k is the number of vertices with degree k and n is the total number of vertices. Thus, the *degree distribution* is given by the following:

$$P(k) \sim \frac{n_k}{n}. \tag{2.11}$$

$P(k)$ follows a power-law distribution if

$$P(k) \sim k^{-\gamma} \tag{2.12}$$

where $\gamma > 0$ is a constant depending on the network and is usually in the range of $2 < \gamma < 3$ [11, 2, 14]. Graphs with a power-law degree distribution are called *scale-free* networks [14, 15]. These scale-free networks consist of a few highly connected vertices, so-called *hubs*, and many less connected vertices [5] (see Figure 2.5). Most of the real-world networks, including metabolic networks, are approximately scale-free networks [14, 15]. Figure 2.6 shows the degree distribution of the metabolic network of *E. coli* with a regression curve for the power-law distribution. Highly connected nodes in scale-free networks can reach other nodes by a short path. Therefore, these networks generally have a short diameter [109] and a high clustering coefficient [169]. The benefit of the scale-free architecture is its robustness against random attacks because it is statistically more probable that vertices with a lower degree are hit while the overall structure of the network is not affected. However, targeted attacks against the hubs can lead to devastating effects [3]. The scale-free topology therefore provides robustness to the network with increased flexibility to random perturbations. Nevertheless, it is susceptible to targeted attacks at heavily connected critical hubs [3], and mutations affecting hubs are more likely to cause a defect [169].

2.4 Descriptors for finding essential nodes in a network

This section explains various graph-based descriptors and approaches. In the following, the most relevant network descriptors are explained for estimating the essentiality of nodes in a network. This section is subdivided into four parts. Section 2.4.1 describes node features that are based on undirected graphs and that can be used for a reaction-based representation of a metabolic network. The other three sections explain features that have been specifically designed for metabolism as a bipartite graph. Section 2.4.2 explains features that are basic properties of a reaction; choke point analysis and damage estimation. Section 2.4.3 considers deviations of possible ways to produce compounds in metabolism. Finally, Section 2.4.4 describes the biomass production of bacteria after knocking out a reaction by the flux balance approach. We conclude the list of these features based on undirected graphs in Table 2.2 and based on directed bipartite graphs in Table 2.3.

2.4.1 Network topological features based on undirected graphs

As mentioned in Section 2.3 and Figure 2.4, a network may be represented as an undirected graph $G = (V, E)$, which consists of a set of nodes V and a set of edges E. Each node $i \in V$ represents a unique cellular entity such as enzymes, genes and proteins, while each edge $(i, j) \in E$ represents an observed interaction between two nodes i and j. To construct an undirected graph for metabolism, the network representation of a reaction-pair network can be used instead of a bipartite graph. Many descriptors of a node in the network describe the communication properties of the node. Node descriptors, also called features, for undirected graphs are described next.

Local topology
In a given network, the local connectivities of a reaction can be measured as the number of its neighboring reactions (NNR) and the number of neighbors of neighboring reactions (NNNR). Recall that the number of the neighboring reactions is the degree of a reaction node in the reaction-pair network (see Section 2.3.2). For a reaction node, v, the number of neighbors of neighboring reactions can be formulated as the following:

$$NNNR(v) = |\{u \in V | \{u, v\} \notin E \text{ and } \exists w \in V : \{u, w\} \in E \wedge \{w, v\} \in E\}| \quad (2.13)$$

The clustering coefficient is used to estimate the local density of the network. It explains the connection among neighbors, which helps to understand the possibility of local alternative communication. The clustering coefficient value (CCV) of a node

Table 2.2: Topological features for networks based on undirected graphs

Short form	Explanation
Local topology	
NNR	Number of Neighboring Reactions (NNR)
NNNR	Number of Neighbors of Neighboring Reactions (NNNR)
CCV	Clustering Coefficient Value (CCV): clustering coefficient of a reaction
Centrality	
BW	Betweenness centrality
CN	Closeness centrality
EC	Eccentricity centrality
EV	Eigenvector centrality

Table 2.3: Topological features for networks based on directed bipartite graphs

Short form	Explanation
Basic properties	
NS	Number of Substrates (NS)
NP	Number of Products (NP)
DIR	Directionality of reaction (DIR): reversible or irreversible reaction
Choke points and load scores	
CP	Choke Point (CP): a reaction is a choke point or not [131]
LS	Load Score (LS): load score of a reaction [131]
Damage	
NDR	Number of Damaged Reactions (NDR) [100]
NDC	Number of Damaged Compounds (NDC) [100]
NDRD	Number of Damaged Reactions having no Deviations (NDRD)
NDCD	Number of Damaged Compounds having no Deviations (NDCD)
NDCR	Number of Damaged Choke point Reactions (NDCR)
NDCC	Number of Damaged Choke point Compounds (NDCC)
NDCRD	Number of Damaged Choke point Reactions having no Deviations (NDCRD)
NDCCD	Number of Damaged Choke point Compounds having no Deviations (NDCCD)
Deviation	
RUP	Reachable/Unreachable Products (RUP): equals one if all products could be produced when blocking the reaction, otherwise zero
PUP	Percentage of Unreachable Products (PUP): the percentage of products which cannot be produced when blocking the reaction
ND	Number of Deviations (ND)
APL	Average Path Length (APL): the average path length of the deviations
LSP	Length of the Shortest Path (LSP): the length of the shortest path of the deviations
Flux analysis*	
BFV	Biomass Flux Value (BFV): biomass flux value when blocking a reaction (under aerobic glucose condition)

* This analysis was done only for a available *in silico* model of *E. coli* iAF1260.

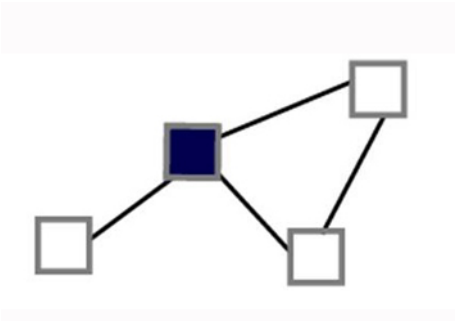

Figure 2.7: **A network example to illustrate the topological features based on undirected graphs**. Rectangles represent reactions and lines represent links between two neighboring reactions. Dark rectangles represent an observed reaction, v. Example of an undirected graph (reaction-based representation of the metabolic network) for computing degree, clustering coefficient and centrality features. The observed node has a degree of 3 ($NNR(v) = 3$) and a clustering coefficient of 1/3 ($CC(v) = 1/3$). The observed node is placed central and more pathways pass through the node (two out of six) compared to the other depicted nodes. Therefore, its betweenness centrality is higher in comparison with the other nodes (observed node: 2 ($BW(v) = 2$); other nodes: 0).

v is defined as follows as the ratio of the number of connecting edges m_v among all of the neighbors of v and the total number of all of the edges among them that could be possible:

$$CCV(v) = \frac{2m_v}{k_v(k_v - 1)}, \tag{2.14}$$

where k_v is the degree of node v. The clustering coefficient of the whole network is defined by the average of the local clustering coefficients of all of the nodes in the network [15, 164, 169]. Numerical examples of clustering coefficients are shown in Figure 2.7.

Centrality measures

In the context of cellular networks, descriptors for *node centrality* are quite powerful for describing the essentiality of the node. They describe not only the impact of the node to its direct vicinity but also the contribution of a node to the global structure of the network. The simplest of all of the centrality measures is connectivity, the degree k, as mentioned in Section 2.3.3, which is used in the feature NNR; NNR describes the local vicinity of the node. In a biological network, the degree is commonly used to describe the importance of a node because we know that most hubs (highly connected nodes) in the network are considered to be essential nodes.

According to the power-law behaviors of real-world networks, metabolic networks follow this rule in the same way. Not only is the degree, which is one of the centrality measures in the network, a good descriptors, but other centralities are also good descriptors in this circumstance. Next, we cover centrality measures that consider the entire network.

Betweenness centrality (BW) is the frequency at which a node has the shortest path that connects all of the pairs of nodes [51]. The betweenness centrality $BW(v)$ for node v is given by

$$BW(v) = \sum_{i \neq j \neq v \in V} \frac{\sigma_{ij}(v)}{\sigma_{ij}}, \tag{2.15}$$

in which σ_{ij} is the number of the shortest paths from node i to node j, and $\sigma_{ij}(v)$ is the number of shortest paths from i to j that pass through node v. The sum is composed of all of the pairs (i, j) of nodes of the network (see Figure 2.7).

Closeness centrality (CN) is defined by the inverse of the average length of the shortest paths from node v to all of the other nodes in the network, *i.e.*,

$$CN(v) = \frac{n-1}{\sum_{i \neq v, i \in V} d_{vi}} \tag{2.16}$$

where d_{vi} is a distance (path length) from v to i and n is the number of nodes in the network [51].

Eccentricity centrality (EC). The eccentricity of a vertex v is defined as the maximal distance from v to every other node in the network. Thus, the eccentricity centrality (EC) is the average of the reciprocal of the eccentricity [95],

$$EC(v) = \frac{n-1}{\max\limits_{i \neq v, i \in V} (d_{vi})} \tag{2.17}$$

where d_{vi} is a distance (path length) from v to i and n is the number of nodes in the network. This measure means that more central nodes have a higher value of EC because such central nodes are the nodes with the smallest eccentricity value [95, 166].

Eigenvector centrality (EV) is based on the assumption that the utility of a node is determined by the utility of the neighboring nodes [23]. This measure scores a node higher if it is connected to high-scoring nodes. This centrality is the principal eigenvector of the adjacency matrix of the network. Let x_i denote the score of a

node i. Thus, the eigenvector centrality $EV(v)$ is the score of x_v. Let A_{ij} be the adjacency matrix of the network, *i.e.,* $A_{ij} = 1$ if there is an edge between nodes i and j, and $A_{ij} = 0$ otherwise. For node i, the eigenvector centrality score is proportional to the average of the eigenvector centrality scores of the neighbors of i:

$$x_i = \frac{1}{\lambda} \sum_{j \in N(i)} x_j \quad = \quad \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} x_j, \tag{2.18}$$

where $N(i)$ is the neighborhood of node i (as defined in Equation (2.4), Section 2.3.1), n is the total number of nodes and λ is a constant. This equation leads directly to the well-known eigenvector equation, $Ax = \lambda x$. Normally, there are different eigenvalues λ for which an eigenvector solution exists. According to the Perron-Frobenius theorem, only the eigenvector of the largest eigenvalue is feasible to be used for the eigenvector centrality [24].

2.4.2 Network topological features based on directed bipartite graphs

Because metabolic networks are constructed by connecting reactions and metabolites in a bipartite graph, some special and specific properties of a node in the graph can be considered. Metabolites are nutrients or, in general, compounds that need to be synthesized or catabolized by the enzymes of a cell. For identifying drug-target enzymes, the network topology features are computed for reaction nodes. Let $G = (V, E)$ represent a directed bipartite graph, where V consists of two disjoint sets of nodes M and R that represent metabolites and reactions, respectively [94]. Each edge connects the nodes from M to R and *vice versa*, representing directed edges leading from the substrates of a reaction to the reaction and from the reaction to its products (see Figure 2.4).

Basic properties of reactions

Reactions can be reversible or irreversible, and this characteristic is also used to describe nodes in metabolic networks, *i.e.,* the directionality (DIR) of a reaction. The *number of substrates* and the *number of products* correspond to the number of different metabolites that are needed for the given reaction and that are produced, respectively. Note that for a node $v \in R$, the number of substrates $NS(v) = d_{in}(v)$ (the in-degree of a reaction node) and the number of products $NP(v) = d_{out}(v)$ (the out-degree of a reaction node) are described in Section 2.3.3.

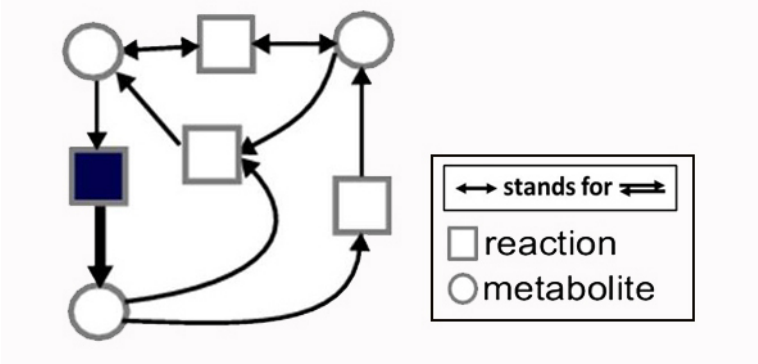

Figure 2.8: **Network examples to illustrate the topological features based on bipartite graphs**. Circles represent metabolites, rectangles represent reactions, and arrows represent directions of the metabolic flux. Dark rectangles represent the observed reactions. The observed reaction is a choke point ($CP(v) = 1$). Enzymes are choke points if they exclusively consume or produce a certain metabolite as depicted here for the filled reaction being the only reaction producing the lower left metabolite.

Choke points

A reaction that is the sole reaction that consumes or produces a certain metabolite in a metabolic network is considered to be a choke point [131, 167] (see Figure 2.8). This feature may make it irreplaceable. The choke point feature $CP(v)$ for a node $v \in R$ is given by the following:

$$CP(v) = \begin{cases} 1 \text{ if } (\exists s \in N_{in}(v) \wedge d_{out}(s) = 1) \vee (\exists p \in N_{out}(v) \wedge d_{in}(p) = 1), \\ 0 \text{ otherwise.} \end{cases} \tag{2.19}$$

where $N_{in}(v)$ and $N_{out}(v)$ are the incoming neighborhood and outgoing neighborhood sets of v, which represent the sets of substrates and products of the reaction, respectively.

Load scores

Load scores are defined to detect hot spots in the network and are based on the ratio of the number of shortest paths passing through a reaction and the number of nearest neighbor links attached to it [131]. This ratio is compared to the average load value in the network. The load score $LS(v)$ of a node $v \in R$ is given by the following:

$$LS(v) = \ln \left[\frac{\sigma_{ij}(v)/k_v}{\sum_{i,j \in M, t \in R} \sigma_{ij}(t) \Big/ \sum_{t \in R} k_t} \right] \tag{2.20}$$

where k_v is the degree of reaction v, and $\sigma_{ij}(v)$ is the number of shortest paths from metabolite i to metabolite j that pass through v.

Damage in global networks

The damage estimates the potentially effected metabolites (the number of damaged compounds (NDC)) and reactions (the number of damaged reactions (NDR)) downstream of the knocked-out reaction [100] (see Figure 2.9). For irreversible reactions, it can be calculated using the procedure in Algorithm 1 for an observed reaction v (if the reaction is reversible, the procedure is performed in both directions and the resulting damaged compounds and reactions are put together). Briefly, the procedure begins by deleting all of the metabolites that are produced by v, which are counted as damaged compounds. Next, all of the reactions for which at least one substrate is missing are deleted and counted as damaged reactions. All of the metabolites that are produced by the missing reactions are effected and are included in the set of damaged compounds. This process is repeated until no further nodes are deleted. All of the deleted metabolites and reactions are collected and are counted, yielding the feature values for damaged compounds (metabolites) and damaged reactions, respectively. Combined with the knowledge of choke points and alternative pathways, the definition of damage can be applied to define further features, such as the number of damaged choke points (see Algorithm 1) and the damaged nodes that cannot be produced by alternative paths [125, 127] (see Algorithm 2).

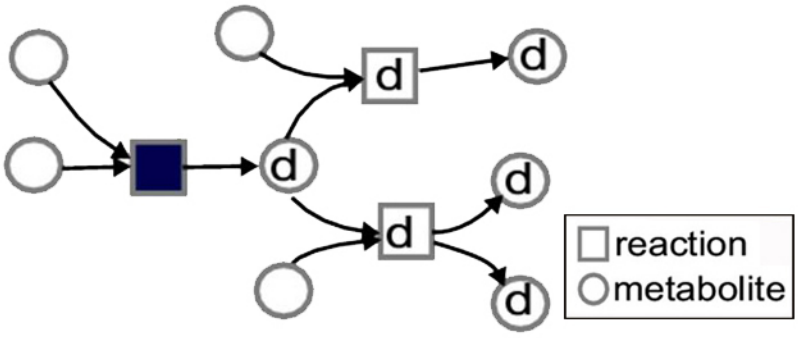

Figure 2.9: **A network example to illustrate the damage feature**. Circles represent metabolites, rectangles represent reactions, and arrows represent directions of the metabolic flux. Dark rectangles represent an observed reaction. When removing the observed reaction, damaged compounds and reactions (node d in circles and rectangles, respectively) are estimated.

Algorithm 1: FindDamage(G,v)

Input : A bipartite graph $G = (V, E)$ where V consists of two disjoint sets of vertices M and R representing metabolites and reactions, respectively. An observed reaction $v \in R$.

Output: Number of damaged compounds (NDC(v)),
Number of damaged reactions (NDR(v)),
Number of damaged choke point compounds (NDCC(v)),
Number of damaged choke point reactions (NDCR(v)).

begin
- $damagedM \longleftarrow N_{out}(v)$
- $damagedR \longleftarrow \emptyset$
- **while** $|N_{out}(damagedM) \backslash damagedR| > 0$ **do**
  - $damagedR \longleftarrow N_{out}(damagedM)$
  - $damagedM \longleftarrow damagedM \cup N_{out}(damagedR)$
- **end**
- $damagedCkpM \longleftarrow \emptyset$
- **for** $u \in damagedM$ **do**
  - **if** $CP(u) = 1$ **then**
    - $damagedCkpM \longleftarrow damagedCkpM \cup \{u\}$
  - **end**
- **end**
- $damagedCkpR \longleftarrow \emptyset$
- **for** $u \in damagedR$ **do**
  - **if** $CP(u) == 1$ **then**
    - $damagedCkpR \longleftarrow damagedCkpR \cup \{u\}$
  - **end**
- **end**
- $NDC(v) \longleftarrow |damangedM|$
- $NDR(v) \longleftarrow |damangedR|$
- $NDCC(v) \longleftarrow |damangedCkpM|$
- $NDCR(v) \longleftarrow |damangedCkpR|$

end

Algorithm 2: FindNoDeviationDamage(G,v)

Input : A bipartite graph $G = (V, E)$ where V consists of two disjoint sets of vertices M and R representing metabolites and reactions, respectively. An observed reaction $v \in R$.

Output: Number of damaged compounds without deviations (NDCD(v)), Number of damaged reactions without deviations (NDRD(v)), Number of damaged choke point compounds without deviations (NDCCD(v)), Number of damaged choke point reactions without deviations (NDCRD(v)).

```
begin
    damagedM ⟵ N_out(v)
    damagedR ⟵ ∅
    for s ∈ N_in(v) do
        for p ∈ N_out(v) do
            if no path(s, p) then
                damagedM ⟵ damagedM ∪ {p}
            end
        end
    end
    while |N_out(damagedM)\damagedR| > 0 do
        damagedR ⟵ damagedR ∪ N_out(damagedM)
        for s ∈ N_in(v) do
            for p ∈ N_out(damagedR) do
                if no path(s, p) then
                    damagedM ⟵ damagedM ∪ {p}
                end
            end
        end
    end
    damagedCkpM ⟵ ∅
    for u ∈ damagedM do
        if CP(u) = 1 then
            damagedCkpM ⟵ damagedCkpM ∪ {u}
        end
    end
    damagedCkpR ⟵ ∅
    for u ∈ damagedR do
        if CP(u) = 1 then
            damagedCkpR ⟵ damagedCkpR ∪ {u}
        end
    end
    NDCD(v) ⟵ |damangedM|
    NDRD(v) ⟵ |damangedR|
    NDCCD(v) ⟵ |damangedCkpM|
    NDCRD(v) ⟵ |damangedCkpR|
end
```

2.4.3 Deviations of nodes in the metabolic network as a bipartite graph

For estimating the feasibility of possible flux deviations if the node under observation is discarded, several descriptors have been established. These features describe possible alternative pathways from substrates of the knocked-out reaction to its products. To calculate these descriptors, a modified breadth-first search algorithm is used to simulate a qualitative metabolic flux from the substrates to the products of the observed reaction when the reaction is discarded. The network without the observed reaction will also be called a "mutated network" in the following. This graph-based investigation was developed to measure "producibility" of the mutated network [54] (see Algorithm 3). Thus, the feasibility of the alternative paths is analyzed. A reaction is more likely to be essential for survival when the mutated network cannot yield the products of the reaction from its substrates or if the mutated network has difficulties reaching the products. The procedure in Algorithm 3 has been proven to be useful for investigating this scenario and is used to identify drug targets for *Plasmodium falciparum* [54]. Briefly, the procedure begins by first

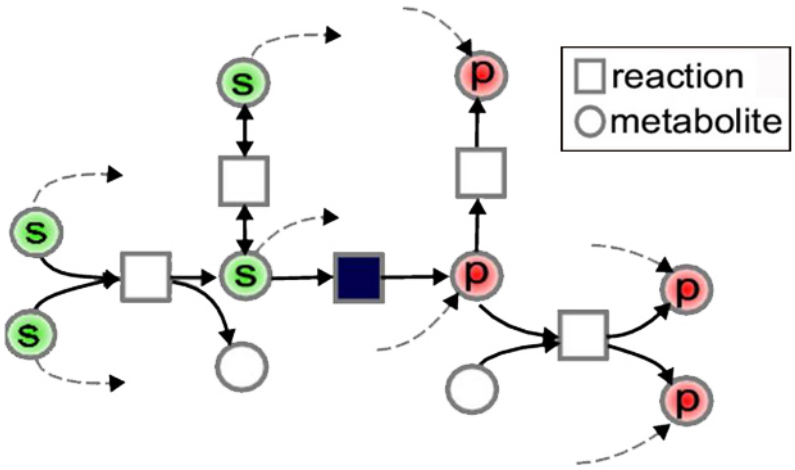

Figure 2.10: **A network example to illustrate the producibility feature**. Circles represent metabolites, rectangles represent reactions, and arrows represent directions of the metabolic flux. Dark rectangles represent an observed reaction v. Reactions nearby the observed reaction and its upstream substrates (S) and downstream products (P). Dash-line arrows represent possible alternative pathways to consume substrates S for producing products P. The producibility gives the percentage of products of the considered reaction that can be produced from the substrates via alternative pathways.

Algorithm 3: Producibility(G,v)

Input : A bipartite graph $G = (V, E)$ where V consists of two disjoint sets of vertices M and R representing metabolites and reactions, respectively. An observed knocked-out reaction $v \in R$.

Output: Reachable or unreachable all of the products (RUP(v)) Percentage of unreachable products (PUP(v)).

begin

$S \longleftarrow N_{in}(v)$ `// Direct substrates`

$P \longleftarrow N_{out}(v)$ `// Direct products`

$upstreamR \longleftarrow N_{in}(S)$
$S \longleftarrow S \cup N_{in}(upstreamR)$
`// include substrates of the reactions upstream of` S

$downstreamR \longleftarrow N_{out}(P)$
$P \longleftarrow P \cup N_{out}(downstreamR)$
`// include products of the reactions downstream of` P

$Rxn \longleftarrow (N_{out}(S) \cup N_{in}(P))\backslash\{v\}$
$S \longleftarrow S \cup N_{in}(Rxn)$
`// include substrates of reactions that have at least one of` S `and produce a metabolite` $\in P$ `into` S`.`

$availableS \longleftarrow S$ `// Set available substrates`

$discoveredR \longleftarrow \emptyset$
while $|N_{out}(availableS)\backslash discoveredR| > 0$ **do**
  $discoveredR \longleftarrow discoveredR \cup N_{out}(availableS)$
  $availableS \longleftarrow availableS \cup N_{out}(discoveredR)$
end
if $|availableS \cap P| = |P|$ **then**
  $RUP(v) \longleftarrow 1$
else
  $RUP(v) \longleftarrow 0$
end

end

$PUP(v) \longleftarrow (|P| - |availableS \cap P|)/|P|$

selecting all of the metabolites that act as incoming nodes (substrates) and outgoing nodes (products) of the knocked-out reaction. The set of substrates is defined as a set of available substrates S, and the set of products P is defined as a set of desirable downstream products P of the knocked-out reaction. To obtain a broader list of available substrates, the substrates of the reactions upstream of S and the products of the reactions downstream of P are included in the sets S and P, respectively. The substrates of the reactions that have at least one of the substrates S as a substrate are also added to S. Furthermore, the substrates of the reactions that have a metabolite of P as a substrate are also included in S. Next, reactions are selected that use compounds of S as substrates. These selected reactions and their products are incorporated into the list of discovered reactions and products. The products are set as newly available metabolites in the network. This process is repeated until no further reactions can be identified. Finally, metabolites of P that cannot be produced are counted (for unreachable products P, see Figure 2.10 and Algorithm 3). After finishing the process, we used the number of desired products that could be produced within the mutated network (defined as the producibility of the mutated network) for two features, *i.e.*, a quality feature defining whether all of the products could be produced (RUP, reachable/unreachable products) and the percentage of products that could not be produced (PUP, percentage of unreachable products). The breadth-first search algorithm was implemented internally in

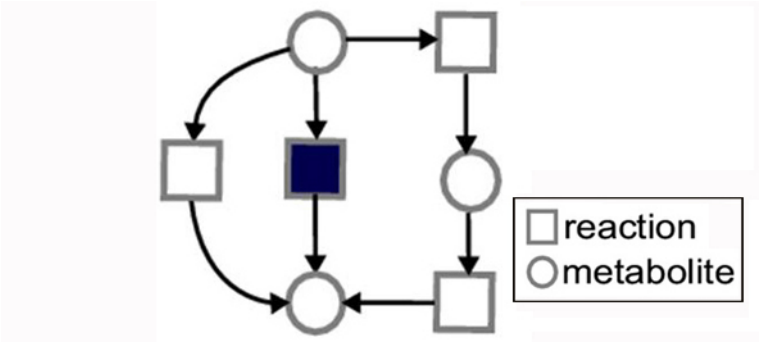

Figure 2.11: **A network example to illustrate the deviation feature**. Circles represent metabolites, rectangles represent reactions, and arrows represent directions of the metabolic flux. Dark rectangles represent an observed reaction v. The example shows alternative pathways for the observed reaction. There are two alternative paths to produce the product of the observed reaction and therefore the number of deviation is 2 ($ND(v) = 2$). Their respective lengths are one and two reactions. Thus, the length of the shortest path is 1 ($LSP(v) = 1$) and the average of alternative path lengths is 1.5 ($APL(v) = 1.5$).

the procedure of Algorithm 3.

We again run a breadth-first search on the network to estimate possible deviations. Starting from the direct substrates, the breadth-first search explored the network for finding the direct products of the knocked-out reaction. When the algorithm visited these products, it stored the corresponding pathway and continued its search to find further alternative paths until the network was entirely explored or a maximal path length of 10 reactions was reached. The organism may have many pathways to produce the products, causing the system to be more robust. Thus, we counted the number of possible alternative paths that yield feature ND (ND, number of deviations). We took the average path length (APL, average path length) and the shortest path length (LSP, length of shortest path) of the deviations as features for the classifier (see Figure 2.11). The deviation features were used to find alternative pathways to produce products of the knocked-out reaction by its substrates S. In the metabolic network, these substrates could also be consumed by other reactions, yielding their products. Therefore, we kept track of alternative paths in the network that had the potential to allow the organism to survive when a reaction was blocked.

2.4.4 Descriptors from flux balance analysis

As introduced in Section 1.5.3, flux balance analysis (FBA) is a mathematical modeling approach that often utilizes quantitative analysis of metabolic flows through microbial metabolisms. In this section, the basic concepts and the mathematical descriptions of FBA are explained.

Metabolic modeling

A metabolic network is a bipartite graph that contains two alternative nodes: reactions and metabolites (see the definition of a bipartite graph in Sections 2.3.1 and 2.3.2). Edges are represented by the consumer or producer relationships between reactions and metabolites.

Let s_{ij} be the stoichiometric coefficient of metabolite i in reaction j, which specifies the number of metabolites produced or consumed by reaction j. Then, $s_{ij} > 0$ indicates that reaction j produces metabolite i while $s_{ij} < 0$ indicates that reaction j consumes metabolite i. The equation $s_{ij} = 0$ means that metabolite i does not participate in reaction j. For example, consider the reaction $A + 2B \rightleftharpoons C$; the stoichiometric coefficients of A, B and C are -1, -2 and 1, respectively. The stoichiometric coefficients s_{ij} can be combined into the so-called *stoichiometric matrix* $S = (s_{ij})$, which is shown in Equation (2.8). The rate of the concentration change

of a metabolite can be formulated by a set of differential equations as follows:

$$\frac{dx_i}{dt} = \sum_j s_{ij} v_j \tag{2.21}$$

where x_i is the concentration of metabolite i, s_{ij} is the stoichiometric coefficient, and v_i is a consumption/production rate for reaction j.

Usually, we model the dynamic behavior of a system in which metabolite and enzyme concentrations change over time with *kinetic models*. However, kinetic models can become complex, with 15-20 parameters for a single complex enzyme; these models require a substantial amount of experimental data. Therefore, kinetic models work well for smaller models in which the kinetic information of the enzymes is known. The ultimate goal for the development of dynamic models is the simulation of the entire cellular metabolism. However, the success of such approaches has been severely hampered by a lack of kinetic information on the dynamics and regulation of metabolism [121, 158]. However, constructed metabolic networks are large, comprising hundreds or thousands of reactions and metabolites in the absence of kinetic information; it is still possible to assess the theoretical capabilities and operative models of metabolism using FBA [20, 49, 55, 56, 121]

Steady-state assumption
FBA is based on the assumption of mass conservation at a steady state, where internal metabolite concentrations are constant over time. Therefore, the concentration change of each internal metabolite i is zero ($\frac{dx_i}{dt} = 0$). With this assumption, Equation (2.21) can be formulated as follows:

$$\sum_j s_{ij} v_j = Sv = 0 \tag{2.22}$$

where S is the $m \times n$ stoichiometric matrix of m metabolites and n reactions in the network. The vector v represents all of the reaction rates (also called metabolic fluxes) in the metabolic network. The ranges of individual metabolic fluxes are then constrained by the following:

$$\alpha_j \leq v_j \leq \beta_j \tag{2.23}$$

where α_j and β_j indicate a minimal and maximal flux of reaction j, respectively. These inequality constraints allow us to model the reversibility of each metabolic reaction. If a reaction is reversible, the flux of the reaction v_j can be either negative or positive. In other words, $-\infty \leq v_j \leq \infty$. Positive v_j indicates a forward direction of the reaction, converting its substrates into its products; in turn, a negative v_j indicates a backward direction. These constraints allow both forward and backward

directions for the reactions. If a reaction is irreversible, its flux constraint would be $0 \leq v_j \leq \infty$. If we want to block a reaction, we can force the flux of this reaction to be equal to zero ($v_j = 0$). In addition, the benefit of these inequality constraints is to simulate metabolic capabilities under certain conditions such as the glucose minimal medium condition, for which we can constrain the flux of the glucose uptake rate within a specific range. Finally, the set of vectors that satisfies all of the set-up constraints in Equations (2.22) and (2.23) is a set of feasible fluxes that define the capabilities of the metabolic network under specific conditions.

For the construction of a metabolic model using stoichiometric constraints when predicting the growth of a cell, it is necessary to formulate the biomass production (see Section 1.5.3). This scenario can be defined as a set of reactions that directly produce the metabolites (*e.g.*, amino acids and nucleotides) into either biomass or macromolecules that form the biomass. Thus, analyzing these feasible fluxes, the production of biomass constituents can be formulated as the following:

$$\sum_j c_j v_j \tag{2.24}$$

where c_j indicates portions of selected fluxes for the biomass composition. If $c_j = 0$, the flux of reaction j is not taken into account. The biomass composition of a given organism comprises the relative amounts of the molecules and these compositions can be found in the literature [55, 121]. The flux that is associated with this biomass composition represents the specific growth rate of an organism. Finally, FBA formulation is a linear programming problem that optimizes the following:

$$max_v \sum_{j=1}^{n} c_j v_j \tag{2.25}$$

and is subject to

$$\sum_{j=1}^{n} s_{ij} v_j = 0 \tag{2.26}$$

$$V_{j,min} \leq v_j \leq V_{j,max} \tag{2.27}$$

where $V_{j,min}$ and $V_{j,max}$ are the boundary conditions of flux j. Comparing the growth rate of mutant and wildtype *in silico* strains reveals the set of essential reaction knockouts. These results can be used to suggest genes and enzymes that are essential and non-essential for a specific condition.

An *in silico* representation of *E. coli* has been well developed and studied to describe a bacterium's metabolic capabilities [49, 55, 56]. Analyzing flux balances under aerobic glucose conditions using the COBRA toolbox [20] and the latest reconstructed metabolic network of *E. coli* (iAF1260 model) has been successfully

applied to detect the essentiality of genes [55]. In this thesis, we performed a single reaction deletion on the network and calculated flux values by FBA using the COBRA toolbox [20] to assess essential reactions under aerobic glucose minimal medium conditions. A reaction was determined to be essential if the respective prediction of the mutated network's maximal biomass production was zero or less than one percent of the wildtype biomass production. The maximal biomass flux value (BFV) of each mutated network was used as our descriptor for each reaction as well.

2.5 Genomic and transcriptomic features

Apart from the topological features that are described in Section 2.4, we also used other features derived from genomic and transcriptomic data, such as information about gene sequences, gene homology and gene expression. An analysis of codon usage with respect to gene sequences was also performed. Homologous genes were counted to support the classification that a knocked-out gene may have homologs in the genome that can replace the function of the knocked-out gene. Gene conservation among various species was measured as a single feature, which is called phyletic retention. In addition, the properties of co-expressed genes using microarray data were also considered. Table 2.4 summarizes the list of all of the genomic and transcriptomic features. In this thesis, DNA sequences of all of the open reading frames (coding regions of genes) were taken from the NCBI database (http://www.ncbi.nlm.nih.gov/, *E. coli*: [GenBank:NC_000913], *P. aeruginosa*: [GenBank:NC_002516] and *S. typhimurium*: [GenBank:NC_003197]). Gene expression data for *E. coli* were obtained from a study in which the regulation during oxygen deprivation was investigated [39] and for *P. aeruginosa* from a study observing the response to agmatine and putrescine treatment [36] and from a study of the quorum-sensing response to environmental conditions [145]. For *S. typhimurium*, we used data from cells that were treated by limiting nutrients at different time points [91] and data from a study that captured the regulatory response in the environment of the host [44].

2.5.1 Genomic features derived from gene sequences

Analysis of codon frequencies and the length of a gene sequence
As explained in Section 1.4.1, genes consist of triplets of bases (codons) that encode amino acids that make up proteins. Codons were counted for each investigated gene from its coding region. We counted base compositions at silent sites (third position of the codons) yielding the features T3s, C3s, A3s and G3s for thymine, cytosine,

Table 2.4: Genomic and transcriptomic features

Short form	Explanation
Codon usage	
Nc	Number of codons
N3s	Base composition at silent sites (T3s, C3s, A3s, G3s)
glt	The frequency of amino acids glutamine (exemplarily)
Homologs	
NAR	Number of Associated Reactions (NAR): the number of reactions that base on the knocked-out gene
Hn	Homology at different expectation values: the number of homologous genes with e-value cutoff $10^{-30}, 10^{-20}, 10^{-10}, 10^{-7}, 10^{-5}, 10^{-3}$ (H30, H20, H10, H7, H5, H3)
Phyletic retention	
PR	Phyletic Retention (PR): the number of homologs in the other prokaryotes
Gene expression	
NGSE	Number of Genes having Similar Expression (NGSE): the number of genes that have similar expression (correlation coefficient > 0.8)
MCC	Maximum of Correlation Coefficients (MCC): maximum value of the correlation coefficients for all neighboring genes

adenine and guanine, respectively. Additionally, the number of codons coding for all of the encoded amino acids (phe, ser, tyr, cys, leu, trp, pro, his, arg, gln, ile, met, thr, asn, lys, val, ala, asp, glu and gly) were counted. All of the codon counts were normalized by the division of the total number of codons (Nc). Nc was also used as a feature.

Analysis of sequence similarity

Two major concerns were followed when the sequence homology was studied. Gene homologs were searched for within the same organism and across organisms. The analysis of gene similarity can be performed by comparing the DNA sequences of two genes using the Basic Local Alignment Search Tool (BLAST [7], http://www.ncbi.nlm.nih.gov/BLAST/). BLAST is an alignment algorithm that compares DNA sequences or amino-acid sequences of different proteins. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences and to identify library sequences that resemble the query sequence

above a certain threshold (so-called E-value). The E-value represents the expected value that describes the number of hits one can "expect" to see by chance when searching a database of the relevant specific size.

- **Homologs within the same species.** Assuming that two genes that have similar sequences may encode proteins with similar functions [34, 112], we calculated the number of homologous genes that may have assumed the function of the knocked-out gene. Homologous genes were searched for using BLAST [7] against all of the open reading frames (coding regions of genes) of the respective organism (*E. coli*, *P. aeruginosa*, *S. typhimurium*). We used different E-value cutoff values, *i.e.*, 10^{-3}, 10^{-5}, 10^{-7}, 10^{-10}, 10^{-20} and 10^{-30}, to obtain different numbers of homologs found within a genome that yielded the features H3, H5, H7, H10, H20 and H30, respectively.

- **Homology across species and phyletic retention.** Homologous genes may be conserved across organisms. It has been shown that conserved genes are more likely to be essential [64]. Therefore, a measure of the number of organisms in which a gene has homologous counterparts (phyletic retention) can be a very predictive feature for essentiality [64]. According to Gustafson's study [64], we selected 177 prokaryotic organisms (except for *E. coli*, *P. aeruginosa* and *S. typhimurium*) from which we counted the number of organisms that had an open reading frame that was homologous to the sequence of the knocked-out gene. This task was performed with *E. coli*, *P. aeruginosa* and *S. typhimurium* using bi-directional best BLAST hits (E-value cutoff of 0.1).

2.5.2 Transcriptomic features derived from gene expression analysis

The expression of thousands of genes can be quantitatively monitored at the same time using a DNA microarray. A DNA microarray is a large array of short DNA molecules that are bound to a glass slide [4, 113], which are specific for each gene to measure, employing Crick-Watson base pairing (for details [4]). To use a DNA microarray to monitor gene expression, mRNA is isolated from the cells being studied and is converted to cDNA, which is labeled with a fluorescent probe and then hybridized to the microarray. After hybridization, the array is imaged. The relative expression of a specific gene is then represented by the image intensity at a specific position, where the DNA sequence of a gene has a spot. To allow for the appropriate comparison of data obtained from different microarrays, normalization

must be performed. At present, there are many established methods for normalization [72, 113, 122], and we used variance stabilizing normalization (vsn) [72]. To obtain an estimate for the similarity of expression profiles, we calculated the Pearson correlation coefficient for all of the pairs of genes [113, 122].

Given the expression ratios for two genes under n conditions, $X = (x_1, x_2, x_3, \ldots, x_n)$ and $Y = (y_1, y_2, y_3, \ldots, y_n)$, and the correlation coefficient can be computed as follows:

$$R = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}} \tag{2.28}$$

where

$$cov(X,Y) = \sum_{i=1}^{n} (x_i - \bar{x}) \times (y_i - \bar{y}), \tag{2.29}$$

$$var(X) = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \quad var(Y) = \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{2.30}$$

$cov(X,Y)$ represents the covariance between X and Y and var is the variance of the data. Correlation coefficient values range from -1 to +1. A correlation coefficient that is close to 1 suggests that the genes behave similarly. If a correlation coefficient is equal to 1, then the two genes have exactly the same expression profiles, while a correlation coefficient value that equal to -1 suggests that the two genes have exactly the opposite expression profiles. A value of 0 means that no linear relationship can be inferred between the expression profiles of the genes. We used gene expression data concerning an unspecific regulation, *i.e.*, not from a small band but from a broad range of effected metabolic pathways. Genes in the same pathway often show co-regulation [138]. Therefore, the maximum correlation coefficient (MCC) of all of the neighboring reactions of the knocked-out reaction was used as a feature. This feature indicated that the knocked-out reaction had a strong connection to its neighbors and might be heavily used in a certain pathway (see Figure 2.12). Additionally, we calculated the number of reactions with similar gene expression (NGSE, correlation coefficient > 0.8) and used it as a feature for an estimate of co-regulated analogous genes.

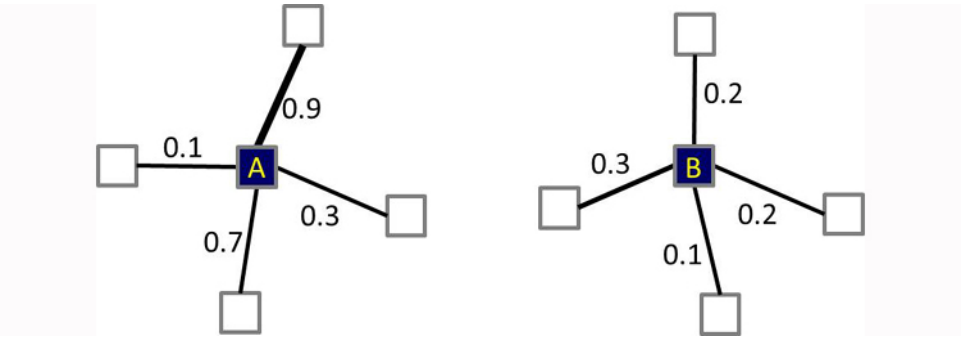

Figure 2.12: **The maximum correlation coefficients among the neighbors**. The figure depicts two subnetworks of two reactions, A and B, with their neighboring nodes in the metabolic networks, where edges are labeled with the correlation coefficients. The maximum correlation coefficient (MCC) of all of the neighboring reactions of the knocked-out reaction was used as a feature to estimate the possibility of the reaction to be used by its neighbors (indicating its importance). Therefore, MCC(A) is 0.9 while MCC(B) is 0.3, which means that reaction A is highly cooperative with its neighbors, assuming that it is more likely to be important than reaction B in the network.

2.6 Preprocessing and feature evaluation

2.6.1 Normalization of features

Features were normalized for the training and validation of the classifiers to bring them into similar orders of magnitudes, which is beneficial for classification [70]. In this work, we had to cope with various scales for features. For example, the feature CP is Boolean, indicating that an observed reaction is either a choke point or not. The value of this feature was set to be one or zero. The feature PUP, estimating percentage of producibility, was continuous and ranged from 0 to 1. The feature NNR, the number of neighboring reactions, ranged from 0 to a large number of nodes in the network. The features were scaled to a range from 0 to 1 using "Min-Max normalization". Let x_i represent all of the values within one set of features; $\acute{x}_i$ are normalized values computed by the following:

$$\acute{x}_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{2.31}$$

2.6.2 Feature evaluation and selection

Feature evaluation and selection is the process of choosing the features that are most relevant for discrimination. This task is useful for reducing the dimensionality of the data for the classifier and results in reducing the computational time and improving the prediction performance. We first evaluated our features using a correlation-based feature analysis [66], and then we selected the optimized set of features with a top-down approach.

Correlation-based feature analysis
We calculated the correlation coefficient between x_f, which represents the vector of all of the sample values of a feature f, and y, which is the vector of all of the sample classes as follows:

$$R(f) = \frac{cov(x_f, y)}{\sqrt{var(x_f)var(y)}} \tag{2.32}$$

where *cov* represents the covariance and *var* is the variance. The correlation coefficient gives the principle relationship between a feature and a class label. For each correlation analysis, we calculated a significance (p-value) for the relationship using $R(f)$ [77].

Top-down feature selection
Final feature selection was performed by a top-down approach. We trained classifiers in terms of maximizing the overall accuracy using all of the features. Each single feature was discarded from the dataset and the performance of the machine was observed. Testing the performance of the machine was performed by a cross-validation. The accuracies of the machines missing one feature were compared and the best machine was kept for the next iteration. This procedure was repeated until the accuracy did not increase. The machine with the best accuracy was selected as the best classifier, and its features were selected as the optimized feature set.

2.7 Machine learning model to identify potential drug targets

All of these features describe the network topology, genomics and transcriptomics and can support finding an essential node in the network. However, these features yield much stronger predictions when combined. There exists a variety of methods to combine such descriptors. We followed a supervised machine learning approach, specifically the Support Vector Machine (SVM), which we used to classify the essen-

tial and non-essential genes. For this task, we describe very briefly how (supervised) machine learning works in principle and provide additional details on SVMs. The machine learning algorithm or classifier requires prior knowledge about a set of objects. These objects are composed of values for their descriptors and a class label. In our case, the object is a node in the network and its descriptors include, for example, whether an object is a choke point, its connectivity or other features. The class label of the object is the property of whether it is essential. The classifier then "learns" from a given dataset for which the class labels are known, which, in our case, is the information of whether a node is essential and may have been observed experimentally by a knockout study for the coding gene of the protein. After learning, the classifier is applied to superimpose the class labels from the given descriptors (features) of new objects for which the class labels are not known. We divided this section into two parts. Section 2.7.1 explains the Support Vector Machine. Section 2.7.2 describes our machine learning approach in combination with a voting scheme technique.

2.7.1 Support Vector Machines

The support vector machine [30, 162] is an effective learning algorithm that is a widely used method for classification. This algorithm addresses the design of a linear classifier that is optimal in the sense that the distances to the points belong to separated classes.

Linear support vector machines
The simplest formulation of the classification task is the task for two possible classes of objects; the two classes could be, for example, -1 for non-essential genes and 1 for essential genes. The classifier attempts to find a function $f: \mathbb{R}^N \rightarrow \{-1, +1\}$, with N being the number of attributes, or features. The object $\vec{x} \in \mathbb{R}^N$with its N features is a feature vector, and $y \in \{-1, +1\}$ is its desired class. A training set that consists of features and class pairs $(\vec{x}, y)$ is used to estimate the function f. For M observations (or samples), the training set T can be written as the following:

$$T = \left\{(\vec{x}_i, y_i) \in \mathbb{R}^N \times \{-1, +1\}, i = 1, ..., M\right\}. \tag{2.33}$$

The N features of an object i are stored in the elements of each vector $\vec{x}_i$. To find a function f that correctly classifies objects, an easy way is to assign an object $\vec{x}$ to be in the class +1 if $f(\vec{x}) \geq 0$; the object would be in the class -1 otherwise. With linearly **separable data**, we can formulate a linear classifier function or a linear discriminant function as the following:

$$f(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b) \tag{2.34}$$

where $\vec{w} \cdot \vec{x}$ indicates the dot product. The *sign* function returns -1 or +1 depending on whether the argument is negative or positive. The object $\vec{x}$ is a feature vector, which is an N-dimensional vector, and $\vec{w}$ is an N-dimensional vector of weights, while b is a scalar. A separating hyperplane is $L = \{\vec{x}|\ \vec{w} \cdot \vec{x} + b = 0\}$. In two-dimensional space, an example of a straight linear separation of the feature vectors that belong to two classes, which are depicted by circles and crosses, is shown in Figure 2.13. To construct a linear classifier, we are looking for a vector $[\vec{w}\ b]$ that satisfies the following system of linear inequalities:

$$\vec{w} \cdot \vec{x}\ +\ b \begin{cases} < 0 & \text{for } y = -1 \\ > 0 & \text{for } y = +1 \end{cases} \tag{2.35}$$

The system in Equation (2.35) is homogeneous, which means that if $[\vec{w}\ b]$ is a solution of the problem, then any other vector $\alpha\,[\vec{w}\ \ b]$ obtained by multiplication by a positive constant $\alpha > 0$ is also a solution [30, 162]. Thus, $\vec{w}$ and b can be rescaled such that the data points closest to the hyperplane satisfy the following:

$$\vec{w} \cdot \vec{x}\ +\ b \begin{cases} \leq -1 & \text{for } y = -1 \\ \geq +1 & \text{for } y = +1 \end{cases} \tag{2.36}$$

which can be reformulated to

$$y(\vec{w} \cdot \vec{x}\ +\ b) - 1 \geq 0 \tag{2.37}$$

For finding a suitable weight $[\vec{w}\ \ b]$ of the system in Equation (2.37), Support Vector Machines use optimization methods by formulating linear or quadratic programming problems. SVMs yield an optimal weight in terms of maximizing margins [30, 102, 110], such as in the description that follows.

Consider two samples $\vec{x}_1$ and $\vec{x}_2$ from different classes, with $\vec{w} \cdot \vec{x}_1\ +\ b = 1$ and $\vec{w} \cdot \vec{x}_2\ +\ b = -1$ in the system of Equation (2.37), respectively. Then, the margin is given by the distance of these two points, which are measured perpendicular to the hyperplane, *i.e.*, $\vec{w}/\left\|\vec{w}\right\| \cdot (\vec{x}_1 - \vec{x}_2) = 2/\left\|\vec{w}\right\|$. Thus, by minimizing $\|\vec{w}\|$, the margin is maximized and the support vectors are the objects $\vec{x}$ that satisfy $y(\vec{w} \cdot \vec{x}\ +\ b) - 1 = 0$. By substituting $\|\vec{w}\|$with $\frac{1}{2}\left\|\vec{w}\right\|^2$ (the factor of $1/2$ is used for mathematical convenience) without changing the solution and recalling the idea of the construction of the system in Equation (2.37), the parameters $\vec{w}, b$ of an optimal hyperplane can be obtained from the solution to the quadratic programming problem

$$\min\ \frac{1}{2}\left\|\vec{w}\right\|^2 \tag{2.38}$$

subject to the constraints

$$y(\vec{w} \cdot \vec{x}\ +\ b) - 1 \geq 0. \tag{2.39}$$

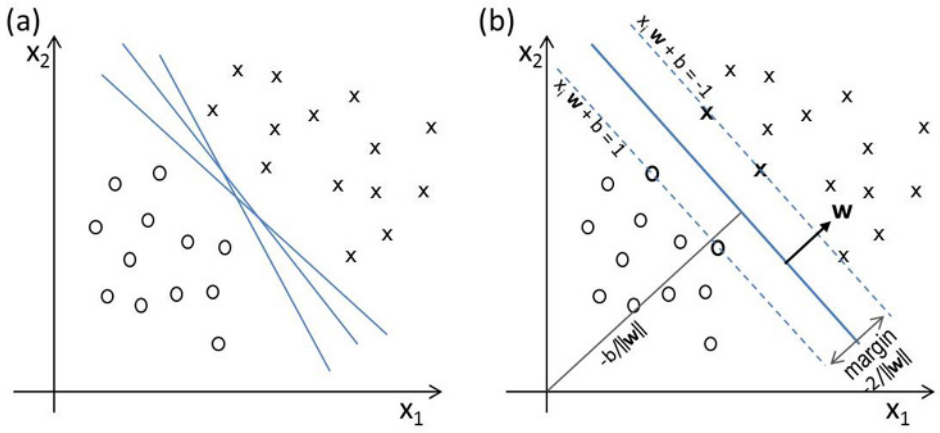

Figure 2.13: **Linear separating hyperplanes in a two-dimensional feature space**. In a two-dimensional space, the data points are plotted for one feature on the horizontal axis and the other feature on the vertical axis. There are two classes, which are marked by circles and crosses. (a) There is an infinite number of possible linear discriminant functions, or lines, for separating the two classes. (b) The SVM attempts to find the optimal separation by maximizing the margin. The dashed lines mark the margin and are chosen using the closest data points to the line. The vectors (points) that constrain the width of the margin are the support vectors and are shown in bold.

There are many examples in which the optimal discriminant function $f(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b)$ has better properties than a "randomly chosen" discriminant function, such as in Figure 2.13, in the optimal line in Figure 2.13(b) and in the randomly chosen line in Figure 2.13(a).
Because it is difficult to resolve this specific optimization with regard to $\|\vec{w}\|$ (the norm is calculated from a square root), $\|\vec{w}\|$ is replaced through $\frac{1}{2}\|\vec{w}\|^2$, which does not change the solutions for w and b. To solve the minimization problem with a multivariate function $f(\vec{w}, \vec{b})$ that is subject to a set of constraints, the technique of Lagrange multipliers can be applied by introducing positive Lagrange multipliers $\alpha_i, i = 1, ..., M$. With the function $f(\vec{w}, \vec{b})$ to be minimized and the constraints $c(\vec{w}, \vec{b})$ given in Equation (2.39) the Lagrangian is the following:

$$L_P = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{M} \alpha_i y_i (\vec{x_i} \cdot \vec{w} + b) + \sum_{i=1}^{M} \alpha_i, \tag{2.40}$$

where $\alpha_i \geq 0, \forall i$. Then, we set the partial derivatives $\frac{\partial}{\partial \vec{w}} L_P$ and $\frac{\partial}{\partial b} L_P$ equal to zero, which results in the following conditions:

$$\vec{w} = \sum_{i=1}^{M} \alpha_i y_i \vec{x}_i \tag{2.41}$$

$$\sum_{i=1}^{M} \alpha_i y_i = 0. \tag{2.42}$$

The L_P in Equation (2.40) forms the primal formulation of the optimization problem. By substituting Equations (2.41) and (2.42) into Equation (2.40), the L_P can be converted into a dual problem L_D, which is the following:

$$L_D = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1}^{M} \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j. \tag{2.43}$$

These two formulations, L_P and L_D, are derived from the same objective function but they address different constraints. By minimizing L_P or by maximizing L_D, both problems yield the same optimal solution.

For the training of the SVM, the original quadratic problem in Equation (2.38) with the constraints in Equation (2.39) can be transformed into the problem of maximizing L_D with respect to α_i subject to the constraints in Equation (2.42) and the positivity of the α_i. Notice that for every training point, there is a Lagrange multiplier α_i. In the solution, for some of those points, $\alpha_i = 0$ holds, and for some, $\alpha_i > 0$. The points with $\alpha_i > 0$ are the support vectors. The support vectors are the critical points of the training set because they lie nearest to the decision boundary, and they determine the separating hyperplane; if all of the other training points were removed and the training was repeated, the separating hyperplane would be the same. Obviously, the weight $\vec{w}$ is explicitly given by Equation (2.41), whereas the bias b is not. However, b can be determined by applying the Karush-Kuhn-Tucker (KKT) *complementary* condition for the primal problem L_P [30, 34]:

$$\alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b) - 1) = 0 \quad \forall i. \tag{2.44}$$

To apply the trained SVM for any test sample $\vec{x}_t$, the following hyperplane decision function has to be evaluated:

$$f(\vec{x}_t) = sign(\vec{w} \cdot \vec{x}_t + b) = sign(\sum_{i=1}^{M} \alpha_i y_i (\vec{x}_t \cdot \vec{s}_i) + b) \tag{2.45}$$

where $\vec{s}_i$ are the support vectors. Notice that in the dual formulation in Equation (2.43), the dot product $\vec{x}_i \cdot \vec{x}_j$ is presented, and here again, the dot product

$\vec{x}_t \cdot \vec{s}_i$ appears. For the formulation of non-linear SVMs, these dot products can be replaced by a non-linear kernel function (see Equation (2.51) below). For **non-separable data**, additional positive slack variables ξ_i that measure classification errors are introduced. Consequently, the quadratic problem in Equation (2.38) with the constraint in Equation (2.39) can be formulated in a relaxation version as the following:

$$\min \left(\frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{N} \xi_i \right), \tag{2.46}$$

which is subject to the following constraints:

$$y(\vec{w} \cdot \vec{x}_i \ + \ b) - 1 + \xi_i \geq 0, \ \forall i, \ \ \xi_i \geq 0 \ \forall i. \tag{2.47}$$

The sum of the slack variables $\sum \xi_i$ is an upper bound on the number of training errors where C is a parameter that controls the penalty for errors and has to be chosen by the user. This formulation is called a *soft margin* classifier, and it allows some training points to lie on the wrong side of the decision hyperplane. The formulation of the primal problem by applying the Lagrangian multipliers α_i and μ_i is the following:

$$L_P = \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{M} \xi_i - \sum_{i=1}^{M} \alpha_i \left[y_i(\vec{x_i} \cdot \vec{w} + b) - 1 + \xi_i \right] - \sum_{i=1}^{M} \mu_i \xi_i. \tag{2.48}$$

where the μ_i were introduced to enforce $\xi_i \leq 0$. The formulation of the dual problem L_D becomes the same as the formulation for the separable case of Equation (2.43) but with an additional constraint $0 \leq \alpha_i \leq C, \forall i$. The solution of the weight $\vec{w}$ is again given by Equation (2.41), whereas the threshold b can be computed when applying the KKT complementary conditions:

$$\alpha_i \left[(y_i(\vec{w} \cdot \vec{x}_i + b) - 1 + \xi_i \right] = 0 \ \text{ and } \ \mu_i \xi_i = 0. \tag{2.49}$$

Non-linear support vector machines

For non-linearly separable samples, we do not need to increase the complexity of the decision function, but rather, we need to increase the power of the classifier by performing the classification in higher dimensional space, for which the data become linearly separable [162]. Thus, the generalization of the above mentioned methods can be formulated for the non-linearly separable samples [162]. Consider the mapping $\Phi : R^N \rightarrow H$, which maps the N-dimensional training data into some generally

unknown but usually higher dimensional space H such that the data become linearly separable as can be expressed by a modified version of Equation (2.37):

$$y(\vec{w} \cdot \Phi(\vec{x}) \ + \ b) - 1 \geq 0. \tag{2.50}$$

As the mapping Φ is usually unknown and the explicit computation of mappings to H (which can be infinite dimensional) is very expensive in terms of resources, this formulation does not allow $\vec{w}$ to be accessed for computation. However, notice that because of the way in which the data appears in the training problem, the dual problem in Equation (2.43) is in the form of dot products, $\vec{x}_i \cdot \vec{x}_j$. With this property, we introduce a given kernel function $K(\vec{x}_i, \vec{x}_j)$ that represents a dot product in Φ [30, 162]:

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j). \tag{2.51}$$

Then, we replace the dot product $\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$ in Equations (2.43) and (2.45) by the kernel function K. Therefore, the actual mapping Φ of the data can be avoided. Next, exactly the same techniques that were developed for the linear case can be used to divide two non-linearly separable samples. The optimization problem in Equation (2.43) then becomes the maximization the following:

$$L_D = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1}^{M} \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j), \tag{2.52}$$

which is subject to

$$\sum_{i=1}^{M} y_i \alpha_i = 0 \ \text{ and } \ 0 \geq \alpha_i \geq C \ \ \forall i. \tag{2.53}$$

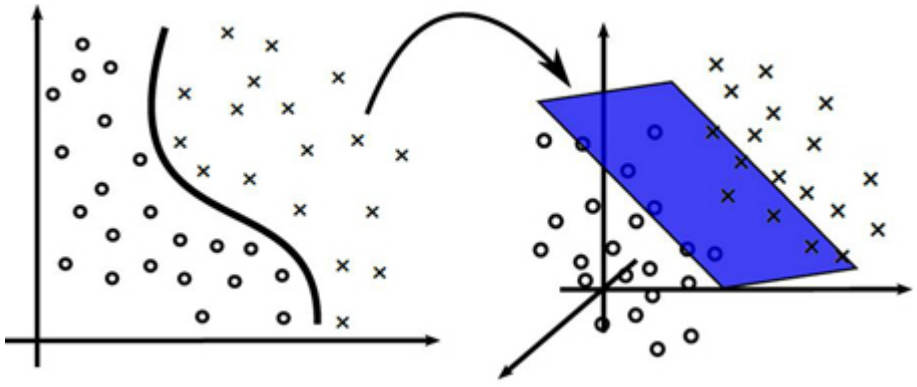

Figure 2.14: **Non-linear separating hyperplanes in a two-dimensional feature space**.

Thus, the hyperplane decision function in Equation (2.45) for a testing sample x_t becomes the following:

$$f(\vec{x}_t) = sign(\vec{w} \cdot \vec{x}_t + b) = sign(\sum_{i=1}^{M} \alpha_i y_i K(\vec{x}_t, \vec{s}_i) + b) \tag{2.54}$$

where $\vec{s}_i$ are the support vectors. A function that satisfies Mercer's condition can be used as a kernel function (see [30, 162]). This condition tests whether a prospective kernel is a dot product in some space but does not provide information on the mapping function Φ or the target space H. There exists a number of kernel functions that satisfy the property $K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$ and that satisfy Mercer's condition. Commonly used non-linear kernel functions are shown in Table 2.5. Figure 2.14 shows a set of data that are not linearly separable in two-dimensional space. The non-linear line classifier can be obtained with a non-linear kernel for which a corresponding mapping (shown in the right part of Figure 2.14) can be found such that the data become linearly separable. In our work, we used the Gaussian radial basis function kernel. Therefore, two parameters needed to be defined: the error penalty parameter C and the kernel parameter γ, which controlled the variance of the Gaussian kernel. To optimize C and γ, we performed a grid search on a limited parameter space [70]. Different parameter pairs ($C = 2^{-4}, 2^{-3}, ..., 2^4$, $\gamma = 2^{-4}, 2^{-3}, ..., 2^4$) were systematically tested to train different SVMs, and their classification accuracies were evaluated using separate validation sets. The pair (C, γ), which led the classifier to the best results, was then chosen as the optimal parameter set, and this task was performed by a cross-validation (see Section 2.8).

Table 2.5: Commonly used kernel functions

Name	$K(x, y)$
Linear	$x \cdot y$
Gaussian radial basic function (RBF)	$exp(\frac{-\|x-y\|^2}{\gamma^2})$
Polynomial	$(x \cdot y + \theta)^\delta$
Sigmoidal	$tanh(\kappa x \cdot y + \theta)$

Greek letters represent real-valued parameters.

One of the crucial issues in machine learning is the problem of unbalanced data for training a machine. This data contains significantly fewer training samples of one class compared to another class. For balanced datasets, support vector machines work well because they aim to optimize overall classification accuracy on training sets. For unbalanced data, the decision boundary tends to be biased towards the majority class; therefore, the minority class samples are more likely to be misclassified. To limit the influence of the majority class in favor of the minority class, the original formulation of SVMs can be extended to **weighted SVMs** [119]. The weighted SVMs allow systematic weighting of classes by introducing class-specific penalty parameters C^+ and C^- instead of using the same penalty parameter C for all of the classes, as introduced in Equation (2.46). Thus, the objective function to be minimized becomes the following:

$$\frac{1}{2}\|\vec{w}\|^2 + C^+\Big(\sum_{i:y_i=1}\xi_i\Big) + C^-\Big(\sum_{i:y_i=-1}\xi_i\Big) \tag{2.55}$$

which is subject to the same constraints as Equation (2.47). Because our data contained small numbers of positive samples, we applied the weighted SVMs with various weights on the positive class. The software library LIBSVM [33] under R environments, which was implemented as the package *e*1071 [45], has been used for our SVM classifications.

2.7.2 Voting scheme

Voting is a commonly used technique to combine the predicted outputs from different classifiers to produce better estimates in a final prediction. With this technique, it has been shown that for a two-class problem, if we have an ensemble with independent classifiers each with an accuracy greater than 0.5, *i.e.*, better than random guessing, the accuracy of the final classifier increases as the number of classifiers increases [48, 139, 140]. Furthermore, this technique can be used to cope with the problem of large data, very small data, and unbalanced data [48, 125]. Because it is often not feasible to train a classifier with the whole dataset, the approach is to divide the data into smaller subsets, train a classifier with each subset, and combine the outputs of these classifiers into a single prediction. With very small datasets, the trained classifier can be unstable if we add or remove just one or two samples. Using the voting technique, we can draw several overlapping subsamples from the original data, learn a classifier with each subsample, and then combine their output by summing over the positive predictions of each classifier (positive "votes"). Unbalanced data can cause problems because the classifier may be overwhelmed by the majority class. To train an unbiased classifier, we can draw balanced subsamples several times instead. To make a final prediction, different thresholds can be

selected, which enables the stringency to be adjusted. Performance measures are then derived by plotting a receiver operator characteristics (ROC), which sketches sensitivities versus specificities for a range of different stringencies (see the next section). Our data were unbalanced, with a small positive and a large negative class. Thus, we enhanced the performance of our predictions by generating several classifiers that were trained with balanced data selected by random sampling and the described voting technique (see Section 3.3).

2.8 Performance measures

We used several performance measures, and they can be determined from *confusion* matrices (also called a contingency table, see Table 2.6). The commonly used measure is the overall classification rate such as the *accuracy* (the number of correctly predicted samples / the number of samples). The accuracy is calculated by the following:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.56}$$

Table 2.6: Confusion matrix for a two-class classification task

		True class	
		Positive	Negative
Prediction class	Positive	*TP*	*FP*
	Negative	*FN*	*TN*

where TP are the true positives, TN are the true negatives, FP are the false positives and FN are the false negatives (see Table 2.6)

Other measures are sensitivity, specificity, positive predictive value and negative predictive value. The *sensitivity* reflects the accuracy on the positive examples while the *specificity* provides the accuracy on the negative examples:

$$\text{Sensitivity} = \frac{TP}{\text{Total Positives}} = \frac{TP}{TP + FN} \tag{2.57}$$

$$\text{Specificity} = \frac{TN}{\text{Total Negatives}} = \frac{TN}{TN + FP} \tag{2.58}$$

The positive predictive value (PPV) measures the percentage of positive predictions made by the classifier that are correct. The PPV is also called *precision*. Similarly,

the negative predictive value (NPV) provides the percentage of correctly negative predictions,

$$\text{Positive predictive value} = \frac{TP}{TP + FP} \tag{2.59}$$

$$\text{Negative predictive value} = \frac{TN}{TN + FN} \tag{2.60}$$

Estimating the performance for the data that the classifier has been trained with leads to an overestimation of its performance. Therefore, another dataset (called the validation set) with known class labels is used and the trained function of the classifier is applied to predict these labels. The predictions are then compared to the true values and the above described performance measures are calculated. This procedure provides a more objective estimate of how the classifier performs when applied to data for which the class labels are not known. After this performance estimation, the classifier is applied to new data. For these data objects, only the feature values are known and the classifier then predicts to which class each object belongs using this information. To achieve a general and reliable performance estimation when the dataset with known class labels is small, the training and testing procedure can be repeated on several independent datasets, which is called a cross validation [154].

Cross validation

The simplest cross-validation is the leave-one-out cross-validation; and this cross-validation scheme was mostly used in our analyses. For this procedure, except for one object, all of the objects with known class labels are included for learning, and the classifier is validated with the object that was taken out for learning. As this procedure does not give a very precise performance estimate, the whole procedure is repeated for every object (with a known class label) to be excluded and the performance is estimated by taking the mean prediction performance of all of these validations. This procedure can, of course, also be performed by excluding more than one object.

For a k-fold cross-validation, the training data are randomly divided into k equally sized non-overlapping subsets, where k indicates the number of subsets, which can be any number from two to the number of training examples. Systematically, one subset is used as a test set and the other subsets are used together as the training set for the training of the classifier. The trained classifier then predicts the classes of the test set, and the predictions are compared to the true class labels to assess their performance. This procedure is systematically repeated, each time a different subset is chosen as a test set and the concatenation of the other $k-1$ subsets as a training set, respectively. The overall performance is estimated by averaging

over all of the performances of each subset used as a test set. In this way, such a cross-validation uses the entire training data in a more efficient manner by increasing the size of the test set, as the entire training data can be used for testing. The overall accuracy can be measured as making predictions on the entire training data. To assess a more general estimate of the overall performance, k-fold cross-validation can be repeated several times using different randomly divided subsets.

ROC curve and AUC

A receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) are the two most common measures for assessing the overall classification performance [124]. The ROC curve was used to measure the performance for a classifier system with various thresholds. The curve is a graph showing the relationship between benefits (correct detection rate or true positive rate) and costs (false detection rate or false positive rate) as the decision threshold varies (see in Figure 2.15). The ROC curve shows that for any classifier, the true positive rate (TPR) cannot increase without also increasing the false positive rate (FPR). The true positive rate is the same as sensitivity, and the false positive rate is equal to

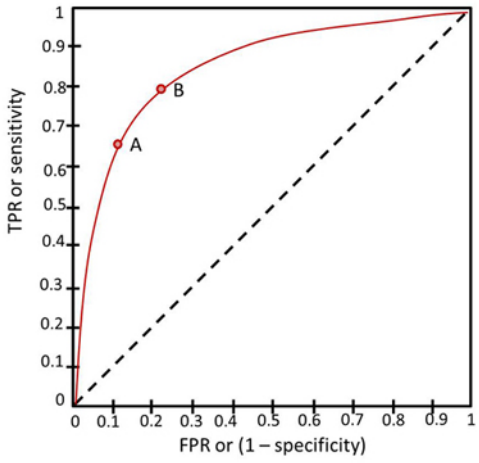

Figure 2.15: **An example of an ROC curve**. Each threshold yields a prediction result and one point in the ROC curve. For example, a certain threshold at point A yields a sensitivity of 0.65 and a specificity of 0.9, while point B (with another threshold) yields a sensitivity of 0.8 and a specificity of 0.8. The best possible prediction method would yield a point in the upper left corner at the coordinate (0,1), which represents 100% sensitivity (no false negatives) and 100% specificity (no false positives). The dashed diagonal line represents a completely random prediction, which yields an area under the curve (AUC) of 0.5. The higher the AUC is, the better the performance of the classifier is.

$$\begin{aligned} FPR &= \frac{FP}{\text{Total Negatives}} = \frac{FP}{TN + FP} \\ &= 1 - \text{specificity}. \end{aligned} \tag{2.61}$$

Hence, the ROC curve depicts the sensitivity versus 1 - specificity for various thresholds (see Figure 2.15). The area under this curve (AUC) is a measure to yield a performance estimate for the entire range of thresholds.

Chapter 3

Results

This chapter describes our results by analyzing the topological features of metabolic networks using a machine learning based analysis for drug target identification. It consists of three sections. Section 3.1 shows the results of analyzing the metabolic network with a new feature that we developed to identify novel potential drug targets. Section 3.2 shows the results of machine learning based integration of various descriptors for a single organism model (*E. coli*). Section 3.3 provides the results of inferring gene essentiality across organisms.

3.1 Analyzing the metabolic network of knockout strains

We investigated the essentiality of a reaction in the metabolic network by deleting (knocking out) such a reaction *in silico*. The algorithm selected products of the investigated reaction that had to be produced by alternative biochemical pathways when the reaction was knocked out. Using a breadth-first search algorithm, we tested qualitatively whether these products could be generated from the substrates of the knocked-out reaction by other reactions that produced potential deviations of the metabolic flux (see Section 2.4.3 deviations). We called this feature "producibility". The producibility yielded two measures of the mutant: RUP (reachability of all of the products) and PUP (percentage of unreachable products). We analyzed the metabolic network of *E. coli* with these new features by knocking out each single reaction *in silico* and comparing the results to a comprehensive experimentally derived list of essential genes (KEIO collection, [12]). Table 3.1 shows the comparison of our producibility features and the other graph-based investigating methods, which em-

ploy the representation of the network as a bipartite graph (see Section 2.4.2), such as choke points (CP), load scores (LS) and damages (originally counting the number of damaged reactions (NDR) and the number of damaged compounds (NDC) [100]). The producibility yielded higher accuracy, sensitivity and precision than the other methods (see Table 3.1). It is worth noting that this approach alone was inferior to flux balance analysis (FBA). For a comprehensive analysis of various network descriptors and genomic features, refer to Sections 3.2 and 3.3, which present the machine learning approach results and its comparison to the FBA approach. As explained in Section 2.4.4, FBA is a comprehensive mathematical modeling method that concerns the stoichiometry for each reaction and the environmental conditions. FBA requires information that has been determined accurately determined, such as the exact constitution of the biomass, the available nutrients and also a sufficiently detailed reconstruction of the network. Furthermore, similar to other graph-based approaches (CP, LS, NDR and NDC), our method can be applied for estimating the potential drug targets of organisms for which less experimental information is available. After setting up the technology, we applied the method to analyze the metabolism of the malaria pathogen *Plasmodium*, for which the detailed network reconstruction needed for FBA analysis would be difficult.

The identification of novel targets for antimalarial drugs remains a difficult task. At present, the genome-wide mutagenesis system in *Plasmodium* is technically challenging [40]. With a computational choke point analysis for *Plasmodium*'s metabolic network, Yeh *et al.* (2004) identified 216 enzymatic activities as catalyzing choke point reactions, assuming that each enzyme has only one active site, unless annotated as multifunctional [167]. If an enzyme catalyzed at least one choke point reaction, it was classified as a potential drug target. Within the 216 identified poten-

Table 3.1: Comparison of our producibility feature with other graph-based features

	Accuracy	Sensitivity	Precision
Our producibility feature			
RUP	69%	86%	29%
PUP	65%	57%	22%
Damage features			
NDC	57%	38%	13%
NDR	59%	38%	14%
Choke points and load scores			
CP	56%	56%	17%
LS	54%	52%	16%

Table 3.2: Results assessing a known drug target for *P. falciparum* to be essential

	Accuracy	Sensitivity	Precision
Choke point feature	68%	74%	19%
Our producibility feature	80%	32%	17%
Intersection of both methods	**88%**	**24%**	**29%**
Union of both methods	60%	82%	16%

tial targets, they identified three targets of clinically proven drugs and 24 proposed drug targets with biological evidence (such as *in vitro* growth inhibition of the parasite with target inhibition). However, the precision (the number of true predictions out of all of the predictions) of their approach is limited, which makes it difficult for a researcher to choose the appropriate potential drug target when developing inhibitors as effective therapeutics. We applied our approach together with the choke point analysis for estimating novel potential drug targets for *Plasmodium* [54], and we improved the specificity when comparing our results to a well elaborated list of known drug targets for *Plasmodium*.

Estimating novel potential drug targets for *Plasmodium*
For this task, we used the data of the metabolic reaction database PlasmoCyc [167]. To estimate the performance of our method, we assembled a list of proposed drug targets from the literature as our gold standard. We used the list of Yeh *et al.* (2004), comprising three targets of clinically proven drugs and 24 proposed drug targets with biological evidence, such as *in vitro* growth inhibition of *Plasmodium falciparum* (*P. falciparum*) [167]. Additionally, we found further drug targets when scanning a variety of established databases, *i.e.*, DrugBank [165], TDR Target Database (www.tdrtarget.org) and the database for Malaria Parasite Metabolic Pathways by Hagai Ginsburg (http://www.sites.huji.ac.il/malaria/, [61]). To equally compare all of the predictions with a gold standard, every reaction of the network was mapped to its corresponding enzyme classification number (EC-number). For performance estimations, the reactions without an EC-number were not taken into account. Thus, our network contained 38 reactions from the gold standard and consisted of a total of 411 reactions. To yield a valid comparison of the algorithms, we did not take the reactions into account that were not in the network. As our network was constructed as a connected graph, we discarded all of the reactions that were not joined with the bulk of the graph.

Table 3.3: Novel potential drug targets for *P. falciparum*

EC-number	Genes	Reaction	Evidence	Human homologs	E-value
2.1.2.9	MAL13P1.67	Methionyl-tRNA formyltransferase	[107]	ENST00000373665	0.19
2.4.1.119	PFI0960W	Dolichyl-diphosphooligosaccharideprotein glycosyltransferase		ENST00000306726	0.034
2.4.2.11	MAL6P1.137	Nicotinate phosphoribosyltransferase		ENST00000370856	0.19
2.4.2.30	PFI1005W	NAD(+) ADP-ribosyltransferase	[132]	ENST00000282892	0.052
2.5.1.	MAL6P1.78	Glutamyl-tRNA(Gln) amidotransferase		ENST00000340159	0.009
2.5.1.46	PF14_0125	Deoxyhypusine synthase	[111]	ENST00000352853	0.54
2.7.1.1		MAL6P1.189 Hexokinase	[99]	ENST00000240487	0.14
2.7.1.35	MAL6P1.266	Pyridoxal kinase	[43, 69]	ENST00000234179	0.011
2.7.1.50	PFL1920C	Hydroxyethylthiazole kinase		ENST00000346134	0.005
2.7.4.7	PFE1030C	Phosphomethylpyrimidine kinase		ENST00000382103	0.086
2.7.4.9	PFL2465C	Thymidylate kinase	[161]	ENST00000340245	0.90
2.7.7.2	PF10_0147	FMN adenylyltransferase		ENST00000256103	0.091
2.7.8.-	MAL6P1.97	Cardiolipin synthetase		ENST00000233710	0.66
2.7.8.11	MAL13P1.82	CDP-diacylglycerolinositol phosphatidyltransferase	[106]	ENST00000321998	0.19
3.1.2.6	PFL0285W	Hydroxyacylglutathione hydrolase	[120]	ENST00000389580	0.35
3.5.1.19	PFC0910W	Nicotinamidase		ENST00000294671	0.12
4.1.2.4	PF10_0210	Deoxyribose-phosphate aldolase		ENST00000356689	0.29
4.2.1.17	PF10_0167	Enoyl-CoA hydratase		ENST00000335407	0.16
4.2.1.60	PF13_0128	3-Hydroxydecanoyl-[acyl-carrier protein] dehydratase	[137]	ENST00000297933	0.25
4.2.1.70	PFB0890C	Pseudouridylate synthase		ENST00000370920	0.37
6.1.1.19	PFL0900C	Arginine-tRNA ligase	[22]	ENST00000231572	2e-04
6.2.1.3	PF14_0761 PFB0695C	Long-chain-fatty-acid-CoA ligase	[156]	ENST0000038037 ENST00000373480	0.23 0.063

Compared to the choke point analysis [167], we were able to improve the prediction results when combining our method with the choke point method (see Table 3.2). Using the choke point analysis alone yielded an accuracy of 68% and a precision of 19%, whereas applying the choke point analysis together with our method yielded an increased accuracy of 88% and a precision of 29%. Finally, we analyzed the "false" positives. This list may serve as candidates for new drug targets. We tested the sequences of these candidates for sequence homology to the human genome to exclude severe physiological side effects when targeting. We identified a refined list of 22 new potential candidate targets for *P. falciparum*, half of which had reasonable evidence that they could be valid targets against microorganisms and cancer [54]. These candidate targets with evidence references are listed in Table 3.3. We compared all of the corresponding genes of those targets to all of the transcripts of the human genome using BLAST [7] and the ENSEMBL database [71]. Arginine-tRNA ligase showed some homology with E-value 4×10^{-4} and may need more detailed homology investigations of its active domain. For the rest, we did not find any significant homologies (all E-values > 0.01). E-values of the best hits and the best hits are given in the last two columns of Table 3.3. In conclusion, our approach is computationally inexpensive and simple to implement and has the potential to serve

as a valid technique to be combined with other established graph-based investigations of metabolism. However, for predicting drug targets *in silico*, a useful model for metabolism is needed.

In another study, we evaluated different network models for *P. falciparum* by estimating their robustness. The networks were constructed using either automatically inferred enzymes from the database PlasmoCyc or only enzymes for which the coding genes were known. Additionally, networks were constructed considering the enzymes of the human host cell. Comparing the modeling results of our network features applied to these four different network constructions showed that we had the best discovery success of known drug targets with a network model consisting only of enzymes from the parasite alone and for which coding genes were known [53]. In principle, such *in silico* investigations can be performed for any organism if its genome and its inferred metabolic network have been discovered in sufficient detail.

3.2 Machine learning analysis to identify drug targets in a single genome model

In this analysis, a machine learning system was trained to distinguish between the essential and non-essential reactions. It was trained and validated by a comprehensive experimental dataset, which consisted of growth rates of single knockout mutants of *E. coli* (KEIO collection [12]). We yielded an overall accuracy of 93% for predicting the essential reactions under rich medium conditions (see Section 3.2.1). Comparative analysis between the flux balance approach and our machine learning approach showed that we yielded a better prediction performance compared to FBA alone (see Section 3.2.2). Predictions that contradicted the KEIO collection were experimentally tested and successfully used to detect errors in the experimental data (see Section 3.2.3). Predicted reactions matching the experimental screen strengthen their candidacy as potential drug targets (see Section 3.2.4). Conclusions and beneficial outcomes of this analysis are in Section 3.2.5.

3.2.1 Performance of the machine learning algorithm

We used a well-established metabolic network of *E. coli* from Feist *et al.* (2007) [55], which is known as the "iAF1260" model. Genes were mapped to the corresponding proteins, enzymes and reactions using the gene-protein-reaction table from Feist *et al.* [55]. The reaction(s) associated with each gene were defined as essential or non-essential if there was no other way to activate the reaction(s) by other genes and if the coding gene was experimentally essential or non-essential, respectively.

Table 3.4: The number of known essential reactions for training a classifier under different conditions

Condition	Reactions		
	Essential	Non-essential	Total
rich medium	231	1,125	1,356
glucose minimal medium	338	1,018	1,356

Otherwise, they were discarded from our training and testing analysis. Furthermore, 133 reactions were discarded from the analysis, as the corresponding genes could not be defined. Finally, from 303 essential genes in the KEIO collection [12] (see Section 2.2), we determined a set of 231 essential and 1,125 non-essential reactions under rich medium. Out of the 1,125 non-essential reactions under rich medium conditions, 107 reactions were defined as essential under glucose minimal medium conditions. In total, 1,356 reactions were used and the experimental results (KEIO [12]) for their essentiality were considered class labels of the reactions (samples) for training and validating the classifiers (see Table 3.4). We used all of the features that were explained in Sections 2.4 and 2.5 except for the phyletic retention and centrality measures.

Because the set of 1,356 reactions for training and validation was relatively small, we performed a leave-one-out cross-validation to measure the effectiveness of our machine learning system. Using the KEIO collection data from the rich medium as the

Table 3.5: Performance of machine learning based predictions on rich medium condition

	Machine Learning (all 30 features)	Machine Learning (25 optimized features)
true positives	168	174
true negatives	1078	1092
false positives	47	33
false negatives	63	57
sensitivity (recall)	72.73%	75.32%
specificity	95.82%	97.07%
positive predictive values (precision)	78.14%	84.06%
negative predictive values	94.48%	95.04%
overall accuracy	91.89%	93.36%

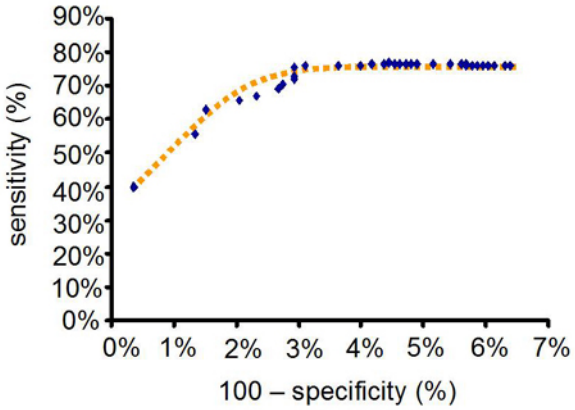

Figure 3.1: **ROC-curve showing our prediction results with different weight factors for positive instances.** Each (blue) diamond shows the result for a different weight. From left to right, the weight was increased from 0.1 to 5.0 by a step size of 0.1. When the weight factors were higher than 1.0, the sensitivity remained constant. The dotted line was manually fitted.

reference, we gained an overall accuracy of 92% when all of the features were considered (Table 3.5). To increase the performance, we conducted a systematic feature reduction within a top-down procedure. We yielded a better result with an optimized feature set of 25 features (accuracy = 93%, see Table 3.5). These 25 features may be regarded as the dominating factors for leading to a good performance. To find out which of them are more relevant, we again started the top-down procedure, stopping at the first step. For all features, we compared the accuracy for each classifier lacking one feature. Losing the feature NNNR yielded the worst classification performance (accuracy -0.89 compared to the classifier with all of the features) and therefore could be the most relevant feature. This feature was followed by NRSE (-0.82), BFV (-0.74), NNR (-0.52) and H10 (-0.52). Interestingly, these first five features already span the whole set of our feature categories (NNNR and NNR: network topology; NRSE: gene expression, genomics; H10: homology, genomics; and BFV: flux balance analysis).

In our dataset, the sizes of the two classes "essential reactions" and "non-essential reactions" differed significantly (essential: 17%, non-essential: 83%). To obtain different stringencies, we weighted the positive instances by a factor ranging from 0.1 to 5.0 with a step size of 0.1 (Figure 3.1). The sensitivity increased significantly from smaller to higher weights, reaching a plateau for weight factors of 1.0 or more. As

expected, with a smaller weight the classifier tended to be overwhelmed by the large negative class. More positive instances were recognized when their weight factor increased. The highest specificity (99%) and the best precision (95%) was yielded by the first data point with a weight factor of 0.1. This scenario is beneficial when predicting drug targets with high reliability. Alternatively, to avoid overlooking potential targets, increased sensitivity can be achieved by raising weight factors to at least a 1.0 (sensitivity = 75%). In the following, all of the analyses were performed with a weight factor of one.

3.2.2 Comparing the performance to flux balance analysis

We performed a single reaction deletion on the network and calculated flux values by FBA using the COBRA toolbox [20] to assess essential reactions under aerobic glucose minimal medium conditions (as described in the supplementary material of Feist *et al.* [55]). In this analysis, a reaction was considered to be essential if the respective prediction of the mutated network's maximal biomass production was $<$ 1% of the wildtype's biomass production. The biomass objective function used in the analysis was explained in [55]. Note that simulating rich medium conditions is challenging because it is difficult to characterize the uptake rates for each compound of a rich medium (Adam Feist, personal communication 2008). For this reason, we compared the performances of our approach with FBA on glucose minimal medium.

Table 3.6: Comparison of our machine learning method and flux balance analysis (FBA) for glucose minimal medium condition

Performance	ML\BFV[1]	ML[2]	FBA[3]
true positives	192	266	174
true negatives	932	968	971
false positives	64	28	25
false negatives	146	72	164
sensitivity	56.80%	78.70%	51.48%
specificity	93.57%	97.19%	97.49%
positive predictive values	75.00%	90.48%	87.44%
negative predictive values	86.46%	93.08%	85.55%
overall accuracy	84.26%	92.50%	85.83%

[1] Machine learning without the feature BFV (biomass flux value from the FBA)

[2] Machine learning including the feature BFV

[3] Flux balance analysis

A total of 338 reactions were found to be essential in glucose minimal medium according to the experimental criteria for gene essentiality under glucose minimal medium [12, 55, 85]. A total of 996 reactions were identified as non-essential. The remaining reactions had no associated gene, were exchange reactions, or could not clearly be identified. The FBA approach detected the essentiality of a reaction under aerobic glucose minimal conditions with an accuracy of 86%, a sensitivity of 52% and a specificity of 98%. We performed our machine learning under glucose minimal conditions with and without BFV (Biomass flux value from FBA simulation) and found BFV to improve the results (Table 3.6). With BFV, our approach yielded 90% precision and 79% recall of the experimental results, compared to the FBA results of 88% precision and 52% recall.

By categorizing the results according to the KEGG pathways [87], a comparison of our method and FBA is shown in Figure 3.2. The essential reactions that were found by our machine learning approach but not by FBA were mostly reactions in amino acid metabolism and lipid metabolism. In amino acid metabolism, the tRNA transferases of almost all of the amino acids were found to be essential by the machine learning approach but not by FBA. We improved FBA simulations by adding the corresponding aminyl-tRNA reactions and their products to the biomass objective function. This process ensured that all of the tRNA transferases would be predicted to be essential by the FBA method. The simulations subsequently gained better results by correctly predicting the essentiality of the aminyl-tRNA reactions.

3.2.3 Validation of the experimental knockout screen

Predicting a different outcome from experimental high-throughput screening (KEIO [12]) may be due to either an error in our algorithm *or* an error within the experimental knockout screen. We examined our lists of false positives and false negatives with two experimental set-ups. Our list of false negatives contained 71 genes, which our algorithm predicted to be non-essential under glucose minimal conditions in contradiction to the outcome of the KEIO experiment [12]. For 33 of them, we obtained corresponding knockout clones from the KEIO library (growing on rich media) and grew them on M9 glucose medium. Indeed, we were able to grow 9 out of 33 clones with good growth rates (OD600 = 0.2 after 48 hours) and 3 clones with reasonable growth rates (OD600 between 0.07 and 0.2 after 48 hours). The complete list is given in Table 3.7.

We also tested the list of false positives, for which our algorithm predicted 33 genes to be essential, in contrast to the experimental high-throughput screen. We assumed that some of these genes were not knocked out correctly. Baba *et al.* [12] provided a validity estimation for their clones. We compared our results to their

estimations and selected 6 genes for which mutants were estimated to be less than or equal to 37.5% correct. The knockout mutations were verified by PCR amplification of genomic loci expected to contain the 1327 base pair gene replacement cassette with specific primers (see Table 3.8). Primers were chosen to have equal predicted melting temperatures of 60 °C and were hybridized at specific distances upstream and downstream of the target gene. PCR reactions were performed directly from freshly grown bacterial colonies for 30 cycles at the annealing temperature of 54 °C. The product sizes obtained from the KEIO collection strains were compared to those from the wildtype *E. coli* strain MG1655 on 1% agarose gels. For 5 out of 6 of these genes (*alaS*, *coaA*, *coaE*, *glyS* and *hemE*), PCR with specific primer pairs

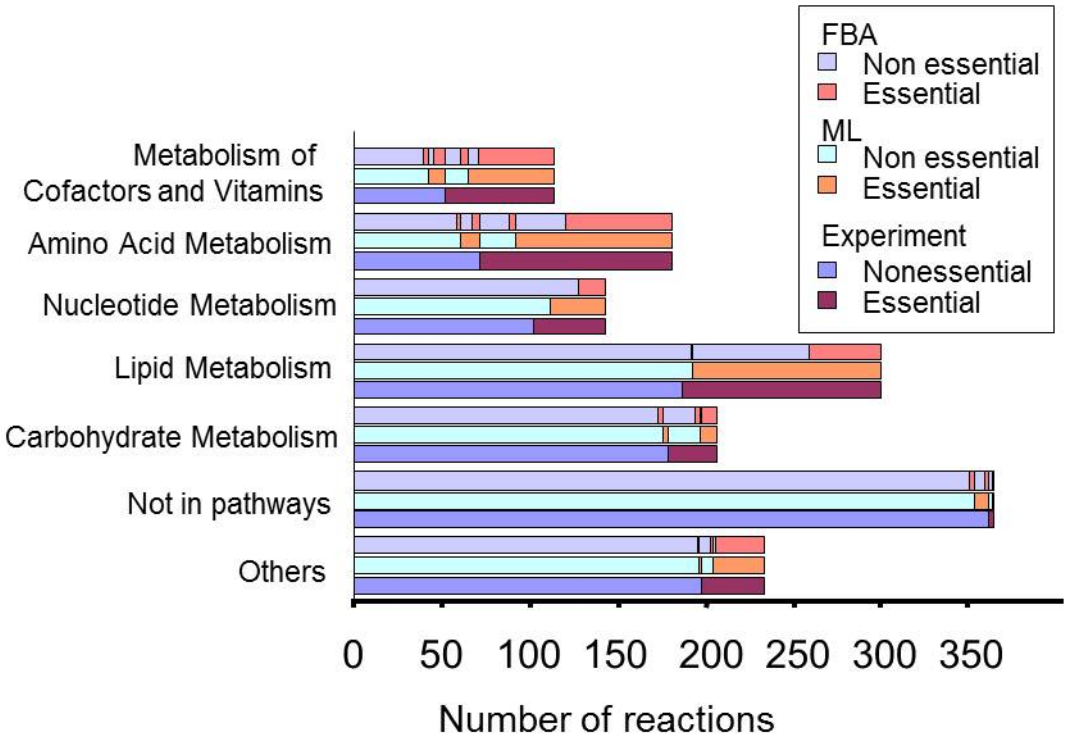

Figure 3.2: **Comparison of our machine learning predictions, FBA and the experimental data, according to different pathways**. For each pathway of KEGG [87, 115], the lowest bars represent the experimental result (KEIO [12]), and the reactions are grouped into non-essential (left, blue) and essential (right, magenta). The middle and top bars show the prediction results of our machine learning approach and flux balance analysis, respectively. Larger differences between the machine learning prediction and the FBA approach were in the amino acid metabolism and the lipid metabolism.

Table 3.7: Results from our growth experiments

ORFs	Gene	Our experimental result[1]	Our OD600 (48 hours)[2]	OD600 of Baba *et al.* (24 hours)[3]
b0002	thrA	-	0.029	0.023
b0032	carA	+	0.432	0.003
b0033	carB	-	0.029	0.000
b0115	aceF	+/-	0.070	0.091
b0116	lpd	-	0.000	0.061
b0242	proB	-	0.045	0.003
b0243	proA	-	0.006	0.007
b0720	gltA	-	0.032	0.014
b0775	bioB	+	0.959	0.049
b0776	bioF	+	0.795	0.025
b0778	bioD	+	1.150	0.037
b1136	icd	-	0.003	0.029
b1260	trpA	-	0.029	0.008
b1261	trpB	-	0.003	0.008
b1264	trpE	-	0.016	0.020
b1638	pdxH	-	0.001	0.005
b2415	ptsH	+	0.747	0.066
b2416	ptsI	-	0.039	0.018
b2508	guaB	-	0.005	0.005
b2530	iscS	+/-	0.084	0.028
b2551	glyA	-	0.000	0.002
b2913	serA	-	0.002	0.007
b3008	metC	+/-	0.152	0.029
b3281	aroE	-	0.001	0.007
b3731	atpC	+	0.833	0.038
b3737	atpE	+	0.671	0.016
b3738	atpB	+	0.989	0.014
b3772	ilvA	-	0.001	0.014
b3829	metE	-	0.008	0.008
b3870	glnA	-	0.000	0.005
b3916	pfkA	-	0.018	0.087
b3940	metL	-	0.000	0.011
b4388	serB	-	0.019	0.009
wildtype		+	0.702	

[1] Our experimental results are classified according to OD600 (minus: < 0.07, no growth; plus/minus: between 0.07 and 0.2, slow growth; plus: > 0.2: growth)

[2] OD600 after 48 hours (measured in duplicate)

[3] OD600 in glucose MOPS medium with 2nM Pi conditions after 24 hours as given by Baba *et al.* (2006) [12]

(see Table 3.8) yielded two products, which had sizes that correspond to wildtype and knockout alleles, respectively. This result indicated that the genes were not correctly knocked out and the wildtype gene was still present. No PCR product was observed for the *ileS* knockout. Additionally we tested another 4 genes from our list, for which mutations were stated to be 100% correct by Baba *et al.* Indeed, for all of those genes (*aspC*, *epd*, *luxS* and *thiE*), only the correct PCR product corresponding to the knockout allele was observed.

Table 3.8: List of the applied primer pairs and our experimental results of testing for correctly knocked-out genes

ORFs	Gene	%correct	Experimental results	fwd primer	rev primer
b0026	ileS	25.00%	no PCR product	5'-gttgcaatggacctttacgg-3'	5'-gctaataccaatcgcaataccg-3'
b0103	coaE	25.00%	2 PCR-bands	5'-aagggtaagagcgcaactcc-3'	5'-tggcaatccaggtttctacc-3'
b2697	alaS	25.00%	2 PCR-bands	5'-ccgactgaacgcatacgg-3'	5'-tacctggttgccctttaccc-3'
b3559	glyS	25.00%	2 PCR-bands	5'-acattcagggcgtagacagc-3'	5'-tctgcctttcggtgaatacc-3'
b3974	coaA	25.00%	2 PCR-bands	5'-aagtagcgcgcattctatgg-3'	5'-acgcggaatagacaaacagg-3'
b3997	hemE	37.50%	2 PCR-bands	5'-gccgtgagcgttactaccc-3'	5'-agagcggttcgaatttaccg-3'
b0928	aspC	100.00%	correct k.o.	5'-gacaacaaactgggcgtagg-3'	5'-ctggatttctggcaaagtgc-3'
b2687	luxS	100.00%	correct k.o.	5'-cccgatctgactttctctgc-3'	5'-ctatcggcacgtcgataacc-3'
b2927	epd	100.00%	correct k.o.	5'-gccggtatcacttcacaagc-3'	5'-cttctcgcctttgttgaagc-3'
b3993	thiE	100.00%	correct k.o.	5'-tacctgcgtaaggaggaagc-3'	5'-actgtgtcagtcgctgttgg-3'

3.2.4 Drug target identification

To propose novel potential drug targets, our predicted reactions were mapped to enzymes that are considered to be possible targets for antibacterial drugs. We examine these predicted targets with known targets that have been used to kill pathogens. Assembling a list of known drug targets, we selected drugs and their corresponding drug targets from the drug database DrugBank [165]. We took drugs into account that affected any organisms except humans and other mammals. Entries that were found as metabolites for a reaction in the KEGG database [87] were discarded to restrict our drug list to non-endogenous compounds. The target annotated enzyme classification numbers (EC-number) of the remaining drugs were collected as our validated drug targets.

We compared the enzymes of our predictions and the results from the KEIO collection to this comprehensive list of valid drug targets (see Figure 3.3). Surprisingly, 80% of the drug target enzymes were not found by the KEIO high-throughput screen and were not found by our machine. This result may be because these drug targets

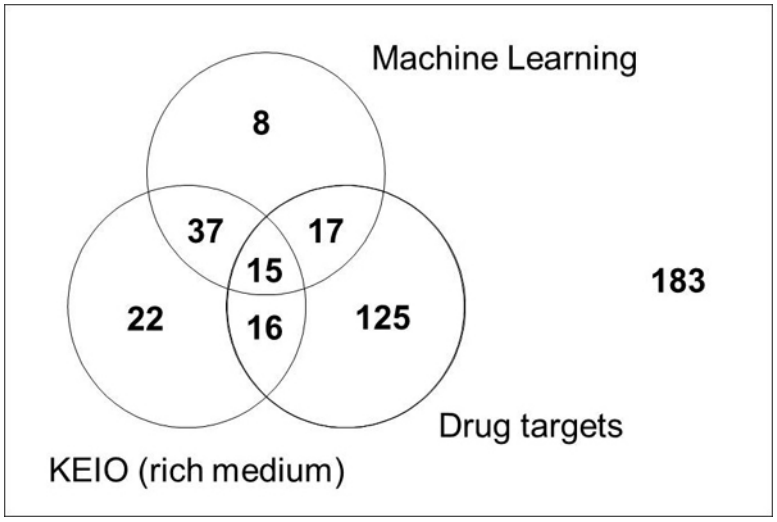

Figure 3.3: **The investigated enzymes.** Three sets are shown: essential enzymes found by the machine learning approach (top circle), essential enzymes found by the KEIO knockout screen [12] (bottom-left circle) and enzymes that are valid drug targets taken from the database DrugBank [165] (bottom-right).

are for a broad range of organisms that have different topologies in their metabolic networks and that may have different alternative pathways for the corresponding drug targets. It should be noted that this study focused on reactions that are essential under rich medium conditions. We suggest 37 promising drug target enzymes that are not in the DrugBank database, which are validated by the intersection of our predictions with the results from the experimental KEIO screen. A list of these enzymes with their enzyme classification numbers (EC-number), open reading frame (ORF) ids, gene symbols and references for reported experimental evidence is given in Table 3.9.

3.2.5 Conclusions

This section presented the results of the machine learning strategy to study and validate essential enzymes of the metabolic network of *E. coli*. Each single enzyme was characterized by its local network topology, its gene homologies and co-expression, and its flux balance analysis. The machine learning system was trained to distinguish between essential and non-essential reactions. It was validated by a comprehensive experimental dataset, which consists of the phenotypic outcomes from single knockout mutants of *E. coli*. We yielded very reliable results with high accuracy (93%) and

Table 3.9: Novel potential drug targets of *E. coli*

EC-number	ORF	Gene	Evidence from the literature
1.1.1.158	b3972	murB	In *B. antracis*, antisense dependent MurB2 expression gave synergistic response to betalactam antibiotics [89].
1.3.1.10	b1288	fabI	FabI is a well known drug target against microorganisms [92].
2.3.1.15	b4041	plsB	Mutations in plsB occur in *E. coli* strains with multi-drug tolerance [152].
2.3.1.51	b3018	plsC	nothing found
2.7.1.130	b0915	lpxK	It was shown that growth of *E. coli* is inhibited when lpxK is inactivated [58].
2.7.1.26	b0025	ribF	nothing found
2.7.4.8	b3648	gmk	*Salmonella* with gmk mutations showed growth dependence on adenine [19].
2.7.7.18	b0639	nadD	nothing found
2.7.7.2	b0025	ribF	nothing found
2.7.7.3	b3634	coaD	nothing found
2.7.7.41	b0175	cdsA	nothing found
2.7.8.13	b0087	mraY	MraY inhibitors serve as novel antibacterial agents [46].
2.7.8.5	b1912	pgsA	PgsA codes for an essential enzyme of *Mycobacterium smegmatis* that shows promise as a drug target for anti-tubercolosis therapy [78].
2.7.8.8	b2585	pssA	nothing found
3.5.1.18	b2472	dapE	*Heliobacter* strains lacking dapE were dependent on diaminopime--lic acid [88].
3.5.4.16	b2153	folE	nothing found
3.5.4.26	b0414	ribD	nothing found
3.5.4.9	b0529	folD	nothing found
4.1.1.65	b4160	psd	Psd null mutants of *E. coli* were non-motile (Karita et al, 1997 [88]).
4.1.2.16	b1215	kdsA	*E. coli* containing missense mutations in kdsA stopped cell growth [57].
4.2.1.52	b2478	dapA	Dihydrodipicolinate synthase is essential for lysine biosynthesis in *E. coli* [163].
4.3.1.8	b3805	hemC	nothing found
4.3.2.2	b1131	purB	PurB mutants of Lotus japonicus exhibit purine auxothrophy [116].
6.1.1.11	b0893	serS	nothing found
6.1.1.12	b1866	aspS	nothing found
6.1.1.16	b0526	cysS	Cysteinyl-tRNA synthetase is essential for protein synthesis [29].
6.1.1.17	b2400	gltX	nothing found
6.1.1.18	b0680	glnS	nothing found
6.1.1.19	b1876	argS	Mutations of argS and leuS are found in *E. coli* strains which are resistant to the antibiotic novobiocin [84].
6.1.1.2	b3384	trpS	nothing found
6.1.1.20	b1713, b1714	pheT, pheS	nothing found
6.1.1.21	b2514	hisS	Mutants expressing a structurally altered HisS protein require external histidine [155].
6.1.1.22	b0930	asnS	AsnS inhibitors are used as anticancer drugs [133].
6.1.1.4	b0642	leuS	Mutations of argS and leuS are found in *E. coli* strains which are resistant to the antibiotic novobiocin [84].
6.1.1.9	b4258	valS	nothing found
6.3.2.13	b0085	murE	*S. aureus* strains which are resistant against the antibiotic methicillin show mutations in murE which is needed for cell wall synthesis [42].
6.3.2.15	b0086	murF	4-phenylpiperidine was reported to inhibit the MurF enzyme and may contribute to antibacterial activity by interfering with cell wall biosynthesis [18].

precision (90%). We showed that topologic, genomic and transcriptomic features describing the network are sufficient for defining the essentiality of a reaction. These features did not substantially depend on specific media conditions and enabled us to apply our approach for less specific media conditions, such as lysogeny broth rich medium. Our analysis is feasible to validate experimental knockout data of high-throughput screens, can be used to improve flux balance analysis, and supports experimental knockout screens to define drug targets.

3.3 Predicting essential genes across bacteria

In Section 3.2, we showed that the method was successfully applied to predict potential drug targets and to validate an experimental knockout screen of *E. coli* [127]. In this section, we used the basic concepts of this strategy to enable predicting essential genes in an organism for which no experimental training data are available. To develop a classification system that is readily applicable for predicting essential genes of a new query organism, the system needs to make accurate predictions for an organism on which it was not trained. Therefore, we performed a cross validation across the organisms of *E. coli* and *P. aeruginosa*, *i.e.* we trained with *E. coli* and validated with *P. aeruginosa* (and *vice versa*) to obtain the quality of the performance of this approach (see Sections 3.3.1 and 3.3.2). We applied the trained and validated classifiers to the pathogenic bacterium *S. typhimurium* (see Section 3.3.3). Furthermore, we analyzed our predictions with gene set enrichment tests for metabolic pathways (Section 3.3.4). We conclude that our machine learning approach is a useful tool to infer essential genes from one organism to another related organism (see Section 3.3.5)

Metabolic networks and gold standards

The metabolic networks of *E. coli*, *P. aeruginosa* and *S. typhimurium* were reconstructed using the database of KEGG [87, 115]. The reactions were mapped to enzymes, and enzymes were mapped to their corresponding genes using the association tables from KEGG. Genes that corresponded to dead-end reactions in the network were not included in the datasets for training and validation. If a gene corresponded to more than one reaction, the mean value of the reaction features was taken. For the Boolean features (RUP, DIR and CP, see Sections 2.4.2 and 2.4.3) we used the Boolean OR-operation, *i.e.*, a gene feature was set to one if at least one reaction feature equaled to one.

To train and validate our predictions for *E. coli*, we used the KEIO collection of Baba and co-workers [12], which we denoted as 'ecoB'. The collection consisted of 104 essential and 641 non-essential genes for the metabolic network. The other dataset

for *E. coli* was from Gerdes and co-workers [60], which we denoted as 'ecoG'. This dataset consisted of 147 essential genes and 533 non-essential genes for our network. For *P. aeruginosa*, we used the data of Liberati *et al.* [103] denoted as 'paeL'. It consisted of 92 essential genes and 615 non-essential genes for the network. The other dataset, for *P. aeruginosa*, was taken from the study by Jacobs *et al.* [79]. We denoted this set as 'paeJ'. It consisted of 150 essential genes and 579 non-essential genes. The experimental dataset for *S. typhimurium* was from Knuth and co-workers [93]. For the metabolism, 53 genes were found to be essential and for the remaining 711, the essentiality could not be determined (see Table 3.10).

The machine learning system
The sizes of the two classes differed considerably in our datasets (essential genes: 8 - 15%, non-essential genes: 85 - 92%). For a broad spectrum of different sensitivities and specificities, we applied a voting scheme. We trained 100 Support Vector Machines (SVMs) with all of the essential genes and an equal amount of randomly selected non-essential genes. With these, we stratified the training data. For the classification of a query gene, the output of all machines was summed up and used as a voting score for the gene to be essential for the cell.

3.3.1 Performance of prediction across organisms

We initiated the process of predicting essential genes for *E. coli*. For this task, we trained classifiers (machines) with the experimental data of two genome-wide knock-out screens of *P. aeruginosa* (datasets paeJ and paeL from experimental studies of Jacobs and co-workers [79] and Liberati and co-workers [103], respectively). These datasets were considered to be our gold standard defining true positives and true negatives of essential genes in the metabolism of the training organism (*P. aeruginosa*). We trained several classifiers with all of the essential genes and an equal amount

Table 3.10: The number of known essential genes in the metabolic network of each dataset

Organism	Dataset	Genes found in metabolic networks		
		Essential	Non-essential	Total
E. coli	ecoB	104	641	745
	ecoG	147	533	680
P. aeruginosa	paeJ	92	615	707
	paeL	150	579	729
S. typhimurium	stm	53	711	764

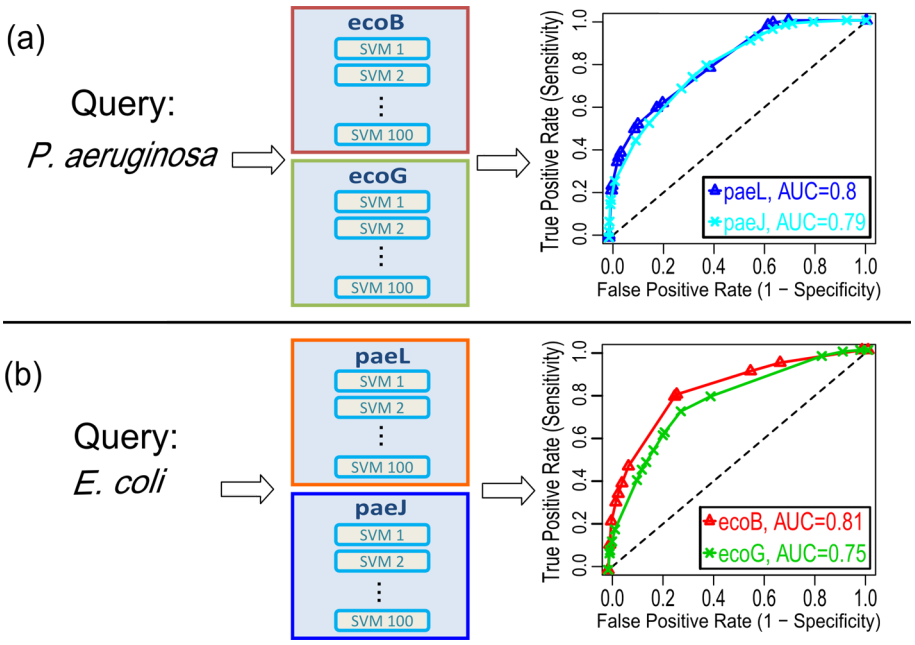

Figure 3.4: **ROC curves of the prediction performances**. (A) A total of 100 Support Vector Machines were trained with the datasets ecoB and ecoG, respectively, and were then queried using the datasets from *P. aeruginosa* (union of the datasets paeL and paeJ). The number of machines predicting essentiality was summed up (voting score). The results from varying thresholds of the voting score were compared to the experimental results of paeL and paeJ, yielding the ROC curves (area under the curve: 0.80 and 0.79, respectively). (B) Similar to (A) only that the machines were trained with the datasets of *P. aeruginosa* and queried with the datasets of *E. coli*, resulting in ROC curves with AUC = 0.81 and 0.75 for the datasets ecoB and ecoG, respectively.

of randomly selected non-essential genes (stratification of the training data). The trained machines were then applied to predict essential genes for the query organism (*E. coli*). The output of all of the machines was summed up and used as a voting score that represented the propensity of a gene to be lethal for the cell. In turn, the same scheme was applied to predict essential genes for *P. aeruginosa* with classifiers that were now trained with two datasets from *E. coli* (ecoB from Baba *et al.* [12] and ecoG from Gerdes *et al.* [60], respectively).

This organism-wise validation was applied to estimate the performance of the classifiers. We compared the datasets for each genome. A total of 79 of the essential genes were common in ecoB and ecoG, while 92 were common in paeL and paeJ. One hundred machines were trained with different training sets for each knockout screen. Votes from both training sets for an organism were summed up and defined the strin-

gency. A high number of votes for essentiality led to a high specificity, while lower numbers led to higher sensitivity. The resulting receiver operator curves (ROC) of the classifiers are shown in Figure 3.4(a) for predicting *P. aeruginosa* and in Figure 3.4(b) for predicting the *E. coli*. For predicting essential genes for *P. aeruginosa* we yielded an area under the curve (AUC) of 0.80 and 0.79 when compared to the experimental datasets paeL and paeJ, respectively. For *E. coli*, we yielded an AUC of 0.81 and 0.75 when compared to ecoB and ecoG, respectively. We wanted to obtain a reliable list of potential drug targets. For this task, predictions of essential genes required a low number of false positives. Hence, we set a high stringency and calculated the precision (true predictions from all of the predictions for essentiality) with a high selection criterion (more than 195 out of 200 votes). We yielded a precision of 67% (accuracy: 87%, sensitivity: 7%, validating with paeL) and 100% (accuracy: 80%, sensitivity: 3%, validating with paeJ) when predicting essential genes for *P. aeruginosa*. We yielded a precision of 61% (accuracy: 87%, sensitivity: 27%, validating with ecoB) and 65% (accuracy: 80%, sensitivity: 18%, validating with ecoG) for *E. coli*. Table 3.11 shows the results of different criteria. We further analyzed different groups of features to examine the best set of features. We grouped features according to their measured properties: a) Deviation (described in section 2.4.3), b) Local topology (in sections 2.4.1 and 2.4.2), c) Choke points and load scores (in section 2.4.2), d) Damage (in section 2.4.2), e) Centrality (in section 2.4.1), f) Homologs (in section 2.5.1), g) Gene expression (in section 2.5.2), h) Phyletic retention (see section 2.5.1) and i) Codon usage (see section 2.5.1). We yielded the best classifier results when using all of the features, compared to the classification performance when using individual sets of features (see Figure 3.5). Table A.2 in Appendix A.2 contains the AUCs for all of the features.

3.3.2 Examining the features

We wanted to obtain an estimate of the correlations of our features to the essentiality of a gene. Therefore, we calculated Pearson's correlation coefficients of the essentiality class of each gene (1 = essential, 0 = non-essential) and the corresponding feature values. Figures 3.6 and 3.7 give an overview for all of the features (see Table A.1 in Appendix A.2 for the correlation coefficients of all of the features). In the following, we describe the major results of our correlation study.

Topology features
The efficiency of flux deviations was estimated by the features RUP and PUP which gave an estimate if all of the products of the knocked-out reaction could be produced without the reaction (RUP) and how large the percentage of non-producible products (PUP) was. RUP was a Boolean feature to observe whether the mutant

Table 3.11: Prediction results for different criteria

	ecoB			ecoG		
Vote	accuracy	sensitivity	precision	accuracy	sensitivity	precision
200	87%	12%	75%	80%	7%	79%
195	**87%**	**27%**	**61%**	**80%**	**18%**	**65%**
190	87%	38%	56%	79%	22%	56%
185	86%	42%	51%	79%	26%	51%
⋮						
150	77%	64%	33%	76%	54%	45%
100	62%	85%	25%	65%	78%	36%
50	42%	94%	19%	47%	86%	27%
0	14%	100%	14%	22%	100%	22%
	paeL			paeJ		
Vote	accuracy	sensitivity	precision	accuracy	sensitivity	precision
200	87%	1%	100%	80%	1%	100%
195	87%	7%	67%	80%	3%	83%
190	**88%**	**15%**	**74%**	**80%**	**5%**	**88%**
185	89%	23%	75%	81%	7%	85%
⋮						
150	87%	41%	48%	82%	27%	69%
100	75%	62%	29%	78%	48%	47%
50	58%	83%	21%	66%	79%	35%
0	13%	100%	13%	21%	100%	21%

could produce all of the products of the knocked-out reaction. RUP was set to one if all of the downstream products could be produced by the mutant while RUP was set to zero if at least one downstream product could not be produced. RUP was highly negatively correlated, and PUP highly positively correlated to the essentiality of the genes (P = 1.2E-10 and P = 2.4E-09, respectively), as shown in Figure 3.6. If the (*in silico*) mutant could not produce one or more downstream products, RUP was zero whereas the percentage of unreachable products was increased compared to the situation in which all of the products could be produced. The higher the percentage was of unreachable products of the mutant, the fewer products of the knocked-out enzyme could be covered by alternative pathways. The number of substrates and products of the reactions of the knocked-out gene (previously denoted as NS and NP) were positively correlated to gene essentiality (P = 4.3E-06 and

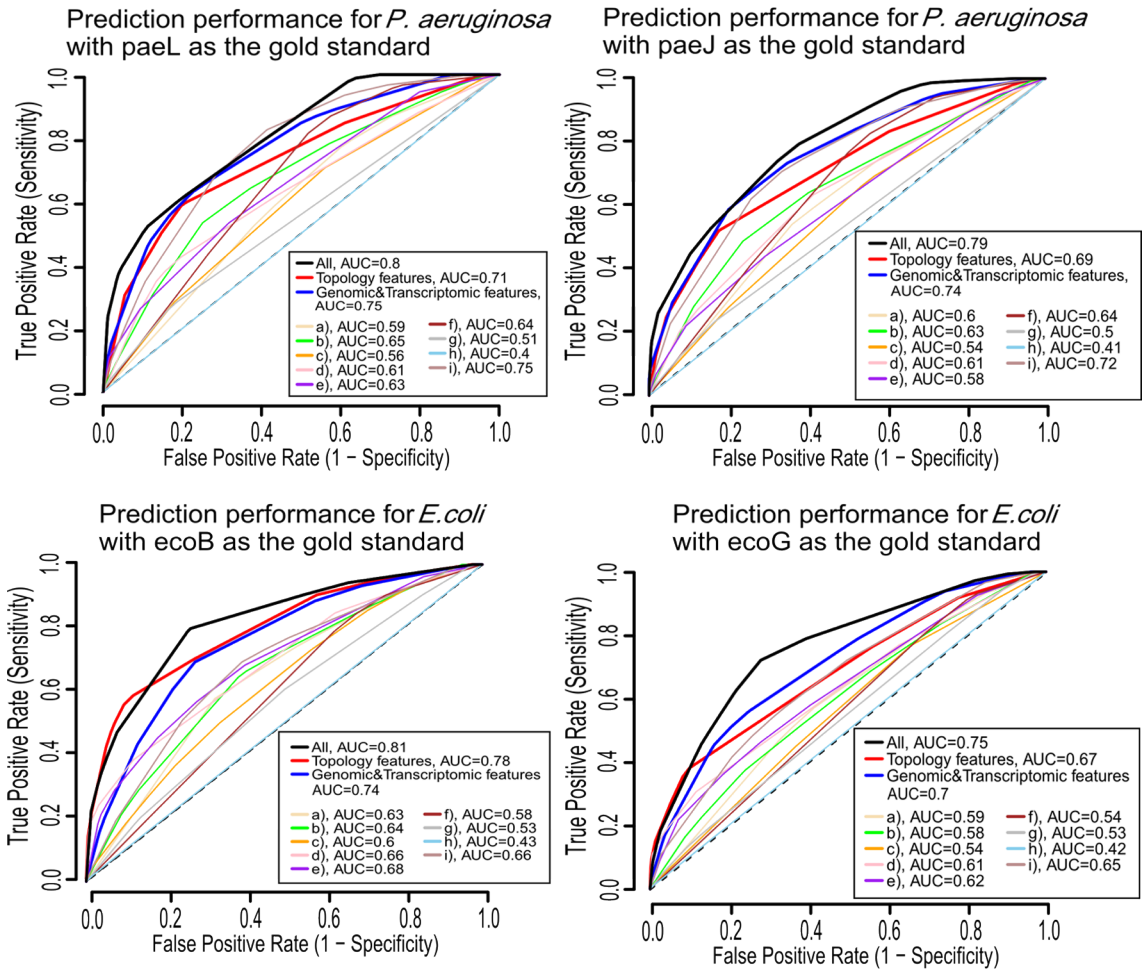

Figure 3.5: **ROC curves for the essential gene predictions with subsets of features**. To evaluate the performance of different subsets of our features, we trained the machines with subsets of features according to their basic groupings to predict essential genes. The figure shows their performances for *P. aeruginosa* with *E. coli* for training (upper row, the left (right) diagram shows the performances with paeL (paeJ) as the gold standard) and vice versa (lower row, the left (right) diagram shows the performances with ecoB (ecoG) as the gold standard). To estimate the overall performances, we calculated the area under the curves (AUC, see figure inserts). The machines that used all of the features performed best (black curves) followed by the set of all of the topology features.

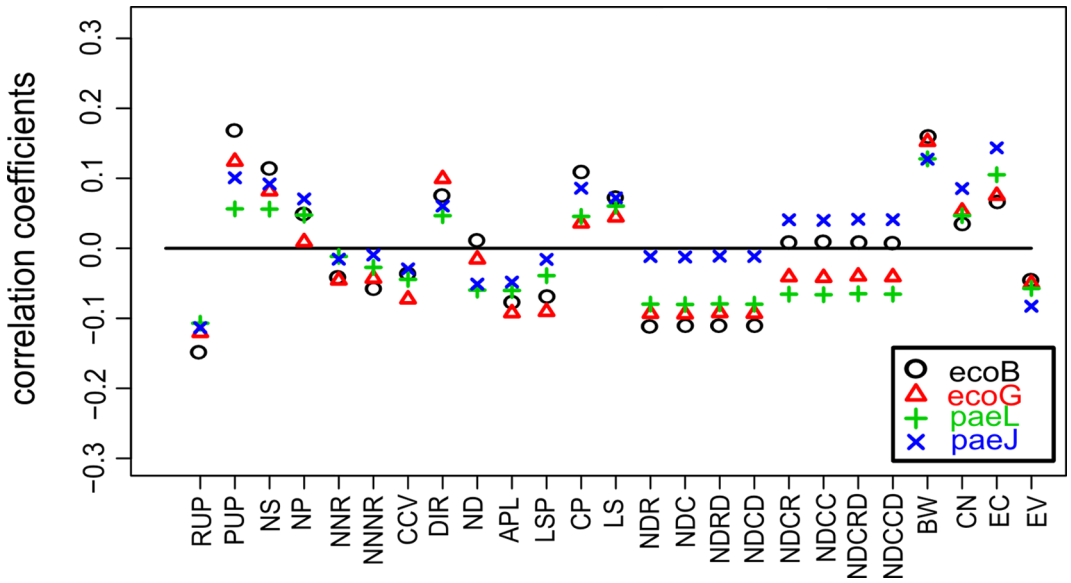

Figure 3.6: **Correlation coefficients for the correlation between essentiality and the topology features**. The feature values of each gene were correlated with the essentiality of the gene (1 = essential, 0 = non-essential). High values indicate that the feature was positively correlated to essentiality (see Table A.1 in Appendix A.2 for all of the correlation coefficients). These values were obtained for all of the gold standards (ecoB and ecoG for *E. coli* and paeJ and paeL for *P. aeruginosa*).

P = 0.0172, respectively) showing that essential enzymes metabolize more different compounds. Interestingly, the number of neighboring reactions (NNR) and the number of neighbors of neighboring reactions (NNNR) showed a weak negative correlation to essentiality (P = 0.14 and P = 0.091, respectively). This construct is reasonable because a reaction with a high number of neighboring reactions may have more metabolites as products that can be produced by alternative enzymes. The clustering coefficients (CCV) showed the same tendency (negatively correlated, P = 0.018), which also pointed to advantageous alternative pathways.

We estimated the feasibility of possible flux deviations by a set of features that describe alternative pathways. The number of alternative pathways (ND), the average path length of the deviations (APL) and the length of the shortest alternative path (LSP) describe the feasibility of the alternative pathways. As expected, all of these pathways were negatively correlated to essentiality (P = 0.15, P = 3.4E-04 and P = 0.0063, respectively), *i.e.*, knocked-out enzymes for which alternative pathways existed were less likely to cause a lethal phenotype if knocked out. Choke points (CP) are reactions that were uniquely consumed or produced compounds in

the metabolism and showed a positive correlation with essentiality (P = 2.8E-04) because choke points are often difficult to replace by the rest of the metabolism. Load scores (LS) give an estimate of how often a reaction is involved in metabolic processes. They were also positively correlated to essentiality (P = 9.4E-04). Betweenness centrality (BW) and eccentricity (EC) were strongly positively correlated to essentiality (P= 1.3E-14 and 7.6E-08, respectively), showing that enzymes have a higher influence on vitality if placed in the center of the network. Closeness centrality (CN) also showed a positive correlation (P = 0.0020). Interestingly, the eigenvector centrality (EV) showed a negative correlation (P = 0.0013). Betweenness, closeness and eccentricity centrality are global centrality measures that consider the whole network, while the eigenvector centrality is a measure for local centrality and is computed from its neighbors. Note that, typically, a node with a high value of eigenvector centrality is a hub (a node with high connectivity) with other hubs connected to it. Hence, flux deviations may be more likely for local hubs that have hubs in their vicinity, making the node replaceable, whereas global central nodes seem to be generally substantial for maintaining the metabolic flow in the network. Therefore, the eigenvector centrality may describe the network topology more in the sense of the clustering coefficient, specifically with respect to the likelihood of alternative pathways.

Genomic and transcriptomic features

As expected, the number of homologous genes (H30, H20, H10, H7, H5 and H3) showed a negative correlation to essentiality (P = 3.2E-04, 6.3E-04, 1.4E-06, 4.7E-09, 1.1E-10 and 1.5E-09, respectively). Interestingly, an E-value cutoff of 10^{-5} (H5) worked best, showing that also non-perfectly matching sequences may take over functions of the knocked-out gene. The number of genes having similar expression (NGSE) also exhibited a negative correlation to essentiality (P = 1.7E-04), which may be due to the co-expression of genes with analogous functions. For the feature phyletic retention (PR), the number of prokaryotes with orthologs of the knocked-out gene showed a positive correlation to essentiality (P = 2.1E-16), supporting the findings of a previous study that the conservation of genes during evolution appear to imply their essentiality [64].

We analyzed the codon usage for each gene and related these to the essentiality of the gene. We found that genes with a high number of the nucleotide thymine at the third position of the codons were more likely to be essential for cell viability (feature T3s in Figure 3.7 and in Figure 3.8 for the histograms). The third codon position is the most redundant position in the genetic code. The matching of mRNA to tRNA codon nucleotides is less robust at the third position, and translational errors are, therefore, more likely to occur at that position. However, essential genes need to be stable and need to be protected in the sequence. Thymine in the genetic code might

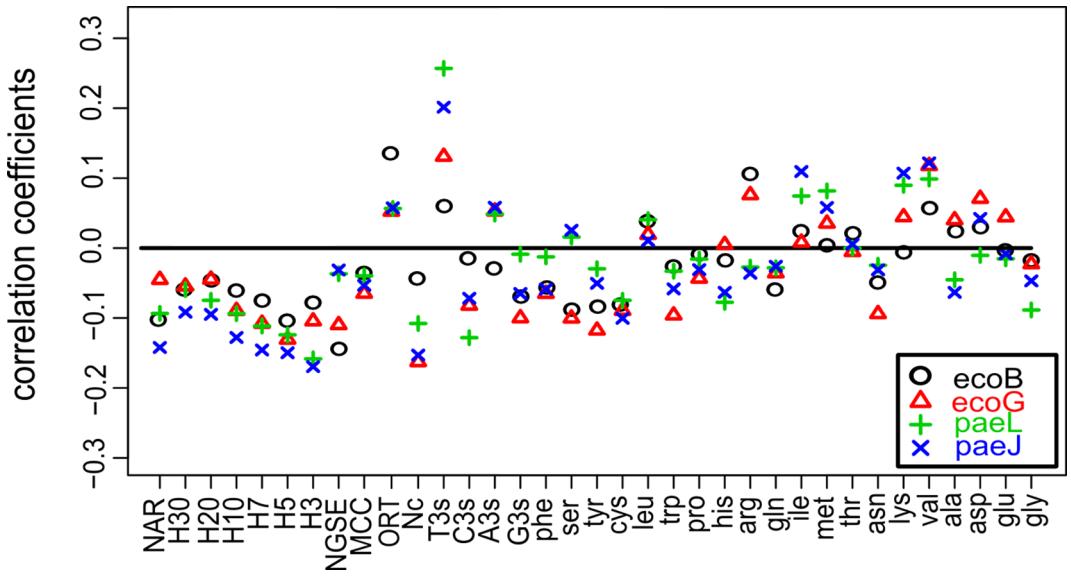

Figure 3.7: **Correlation coefficients for the correlation between essentiality and the genomic and transcriptomic features**. The feature values of each gene were correlated with the essentiality of the gene (1 = essential, 0 = non-essential). High values indicate that the feature was positively correlated to essentiality (see Table A.1 in Appendix A.2 for all of the correlation coefficients). These values were obtained for all of the gold standards (ecoB and ecoG for *E. coli* and paeJ and paeL for *P. aeruginosa*).

address these needs because it has been shown that thymine protects DNA and improves the efficiency of DNA replication [101]. Conserved genes are more likely to be essential [64], and a thymine at the 3rd codon position facilitates stable genetic inheritance into the off-spring and cellular replicates. Interestingly, we observed a larger difference of T3s in *E. coli* when compared to *P. aeruginosa*. It was found that a large average of G and C content at the third codon position is common for all of the genes in *P. aeruginosa* [62]. This observation results in a low T content at the third codon position, which we observed, and may explain the larger difference of T3s for essential and non-essential genes in *E. coli* compared to *P. aeruginosa* (see Figure 3.8).

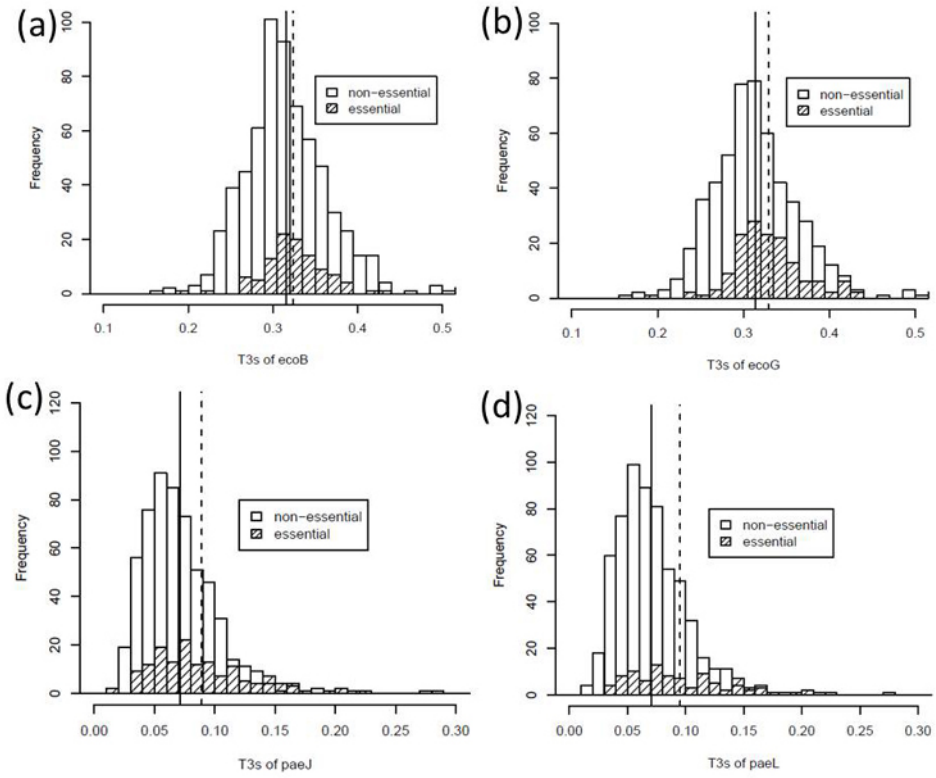

Figure 3.8: **Histograms for the frequency of T3s in essential genes and non-essential genes of *E. coli* (upper row) and *P. aeruginosa* (lower row).** To examine the relationship between the number of thymines at the 3rd codon position (T3s) and the gene essentiality, the figures show the T3s distributions of essential and non-essential genes in *E. coli* (upper row, left and right for the datasets ecoB and ecoG, respectively) and in *P. aeruginosa* (lower row, left and right for the datasets paeL and paeJ, respectively). Dashed lines indicate the average of T3s in essential genes, solid lines indicate the average of T3s in non-essential genes. *E. coli* and *P. aeruginosa* are gamma-proteobacteria, which are not closely related. It has been observed that *P. aeruginosa*'s genome is GC-rich, while *E. coli*'s genome shows an ordinary GC content. The large average of G and C at the third codon position is common for all of the genes in *P. aeruginosa* [62]. This observation results in a low T content of the third codon position, which we can also observe here and which may explain the larger difference of T3s for essential and non-essential genes in *E. coli* compared to *P. aeruginosa*.

3.3.3 Identifying drug targets for *S. typhimurium*

We applied our trained machines from all of the four datasets (ecoB, ecoG, paeL and paeJ) to predict essential genes for *S. typhimurium*, and we obtained votes from four hundred machines for each gene of *S. typhimurium*, to predict a gene as essential. To obtain a reasonable threshold for the number of votes needed to predict that a gene would be essential, we compared the number of genes predicted to be essential with the numbers in the training sets for *E. coli* and *P. aeruginosa*. For *E. coli*, 104 and 147 genes were essential, corresponding to the datasets ecoB and ecoG, respectively, and for *P. aeruginosa*, 92 and 150 (corresponding to datasets paeL and paeJ, respectively). Therefore, we set a threshold of 350 votes (out of 400 machines) to classify a gene as essential for *S. typhimurium*, and we obtained a comparable resulting amount of 128 predicted essential genes. We then compared our results to the experimental data from Knuth and co-workers, who performed a large knockout study for *S. typhimurium* [93]. They detected 6% of all of the open reading frames as being essential including 53 essential genes coding for enzymes in metabolism. For the remaining open reading frames of the genome they did not make any prediction, including 711 genes for metabolic enzymes.

We compared the list of essential genes of Knuth and co-workers with our predictions and found 27 of our predicted genes in their list yielding a precision of 21%, an accuracy of 83% and a sensitivity of 51%. It is worth noting that the experimental screen of Knuth and co-workers was not comprehensive; the authors stated in their article that for the genes that were predicted to not be predicted as essential, they could not conclude that these genes were definitively non-essential. Therefore, our novel predictions may be suitable as potential new targets for further investigations. As a conservative and robust estimate of essential genes for *S. typhimurium*, we defined the corresponding enzymes of genes that were experimentally determined (by Knuth and co-workers) *and* were recognized by our classifiers. We then searched in the literature to find drug treatments of these enzymes for other microorganisms. We compared the open reading frames of the predicted genes with the human transcripts and did not detect significant homologs (using BLAST [7] and ENSEMBL cDNA transcripts [71]). The results are listed in Table 3.12 with open reading frame (ORF) ids, gene symbols, enzyme classification numbers (EC-number), enzyme names and references for reported experimental evidence. The last two columns of the table provide E-values of the best hits and the best hits.

Table 3.12: Novel potential drug targets for *S. typhimurium* from the intersection of our predictions with the experimental knockout screen

ORF	Gene	EC-number	Enzyme	Evidence	Human homologs	E-value
STM0123	murE	6.3.2.13	UDP-N-acetylmuramoylalanyl-D-glutamate-2,6-diaminopimelate ligase	[28]	ENST00000364688	9.6
STM0128	murG	2.4.1.227	N-acetylglucosaminyl transferase	[96, 67]	ENST00000408249	0.73
STM0129	murC	6.3.2.8	UDP-N-acetylmuramate-L-alanine ligase	[168]	ENST00000408663	0.61
STM0154	lpdA	1.8.1.4	Dihydrolipoamide dehydrogenase		ENST00000411150	2.1
STM0218	pyrH	2.7.4.22	Uridylate kinase	[134]	ENST00000410942	4.7
STM0221	uppS	2.5.1.31	Undecaprenyl pyrophosphate synthase	[123]	ENST00000386578	0.18
STM0222	cdsA	2.7.7.41	CDP-diglyceride synthase		ENST00000362327	5.5
STM0228	lpxA	2.3.1.129	UDP-N-acetylglucosamine acyltransferase		ENST00000386484	1.3
STM0232	accA	6.4.1.2	Acetyl-CoA carboxylase	[160, 159]	ENST00000410499	6.1
STM0489	hemH	4.99.1.1	Ferrochelatase	[6]	no hit	
STM0535	lpxH		UDP-2,3-diacylglucosamine hydrolase		ENST00000386574	4.6
STM0542	folD	1.5.1.5,	Bifunctional 5,10-methylene-tetrahydrofolate dehydrogenase		no hit	
STM0988	kdsB	2.7.7.38	CTP:CMP-KDO cytidylyltransferase	[80, 97]	ENST00000386088	1.2
STM1194	fabD	2.3.1.39	Acyl carrier protein S-malonyltransferase	[85]	ENST00000385201	5.9
STM1195	fabG	1.1.1.100	3-ketoacyl-(acyl-carrier-protein) reductase	[151]	ENST00000388337	0.3
STM1200	tmk	2.7.4.9	Thymidylate kinase		ENST00000387015	4.1
STM1700	fabI	1.3.1.10	Enoyl-(acyl carrier protein) reductase		ENST00000387331	1.3
STM2483	dapE	3.5.1.18	Succinyl-diaminopimelate desuccinylase		ENST00000408717	7.2
STM2652	pssA	2.7.8.8	Phosphatidylserine synthase	[149]	ENST00000365512	8.7
STM3090	metK	2.5.1.6			ENST00000388372	7.4
STM3415	rpoA	2.7.7.6	DNA-directed RNA polymerase subunit alpha		ENST00000385068	1.1
STM3724	kdtA		3-deoxy-D-manno-octulosonic-acid transferase	[21]	ENST00000363352	2.1
STM3730	dfp	4.1.1.36	Pantothenate kinase	[98]	ENST00000410954	0.5
STM3912	rep	3.6.1.-	ATP-dependent DNA helicase Rep	[82]	ENST00000222567	0.2
STM3978	yigC				ENST00000364285	0.61
STM4153	rpoB	2.7.7.6	DNA-directed RNA polymerase subunit beta	[35]	ENST00000362682	6.5
STM4154	rpoC	2.7.7.6	DNA-directed RNA polymerase subunit beta'		ENST00000388141	1.7

3.3.4 Pathway enrichment with essential genes

The non-mevalonate pathway and fatty acid biosynthesis are highly enriched with the essential genes of *S. typhimurium*. We performed gene set enrichment tests (Fisher's exact tests) with all of the pathways from KEGG [115], and we found a significant enrichment of essential genes in the non-mevalonate pathway (P = 9.2E-06) and in the fatty acid biosynthesis pathway (P = 3.8E-04). Most of the genes in these pathways were essential (8 out of 9 genes in the non-mevalonate pathway and 8 out of 12 genes in the fatty acid biosynthesis pathway). The non-mevalonate pathway (Figure 3.9) produces isopentenyl diphosphate (IPP) and dimethylallyl pyrophosphate (DMAPP), which serve as a basis for the production of sterols, dolichols and ubiquinone, as well as components of macromolecules, such as the prenyl groups in proteins [73]. The pathway for non-mevalonate biosynthesis has been considered previously to be attractive targets of novel antibiotics against bacteria [76, 150], including *S. typhimurium* [38, 157]. Figure 3.9 shows the non-mevalonate pathway and its essential enzymes for *S. typhimurium*. Note that the arrows in the figure do not represent information about the irreversibility of these reactions but rather show the direction of the overall flux. This pathway, which is mostly linear, starts at 1-deoxy-D-xylulose-5-phosphate-synthase (EC-number: 2.2.1.7), which has a corresponding gene *dxs* that has been identified to be essential by the experimental knockout study of Knuth and co-workers [93]. The next six enzymes downstream were predicted to be essential by our method. The last enzyme we found in this pathway was geranyltranstransferase (EC-number: 2.5.1.10), which catalyzes a reaction to produce farnesyl-diphosphate. Recently, Cornish and co-workers performed an elaborate mutagenesis study of the non-mevalonate pathway in *S. typhimurium* and found five genes to be essential (*ispD*, *ispE*, *ispF*, *ispG* and *ispH*) [38]. We propose that all of the eight enzymes in this pathway are promising potential drug targets for *S. typhimurium*. These genes and their evidence are listed in Table 3.13. We did not detect significant homologs of the human transcript when using BLAST [7] and the ENSEMBL database [71]. E-values of the best hits and the best hits are given in the last two columns of Table 3.13.

3.3.5 Conclusions

In this section, we presented the results of the machine learning technique to identify essential genes using the experimental data of genome-wide knockout screens from one bacterial organism to infer the essential genes of another related bacterial organism. We used a broad variety of topological features, sequence characteristics and co-expression properties that are potentially associated with essentiality, such as flux deviations, centrality, codon frequencies of the sequences, coregula-

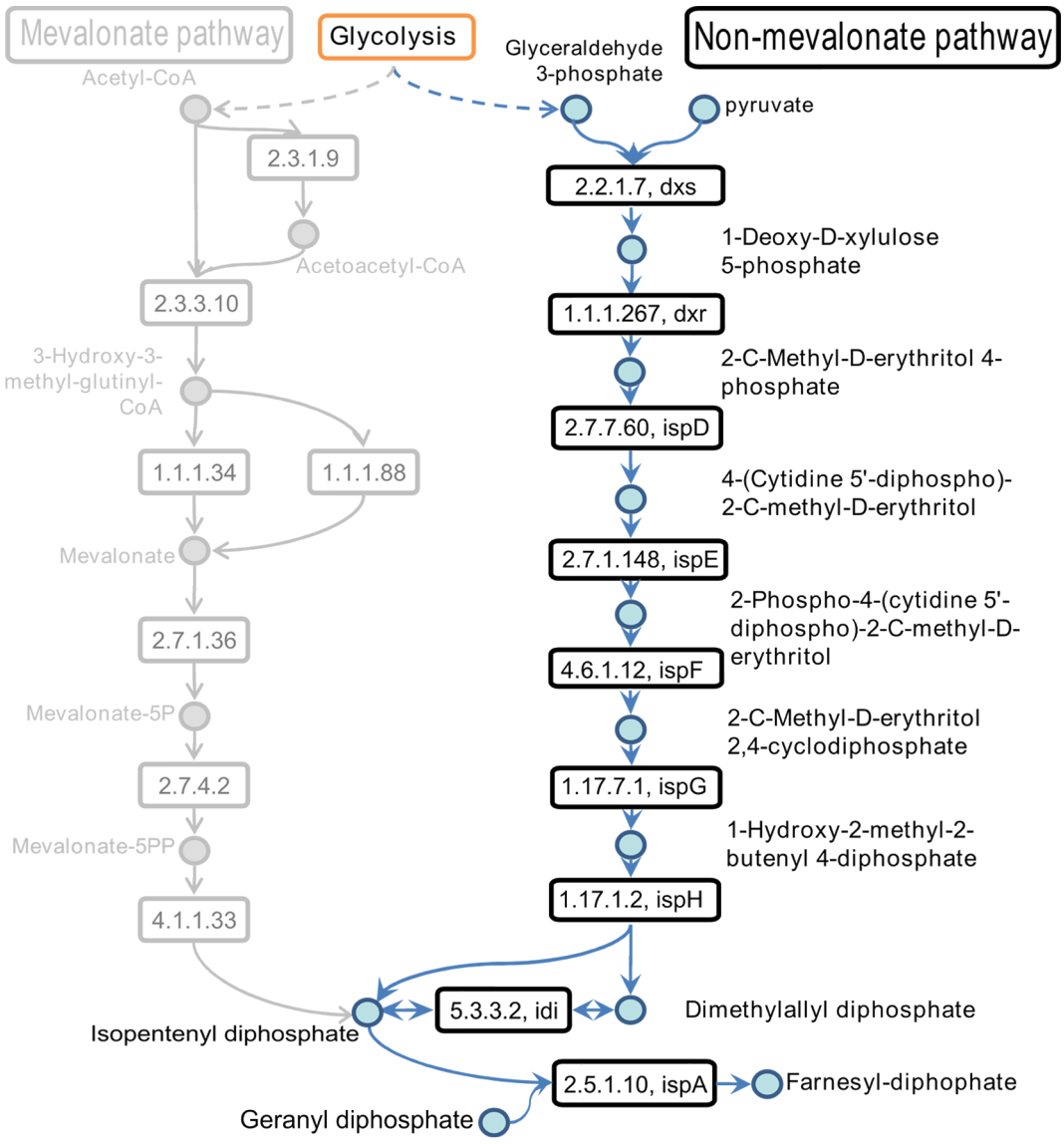

Figure 3.9: **The non-mevalonate pathway**. The non-mevalonate pathway produces isopentenyl diphosphate (IPP). This pathway is an alternative pathway in bacteria and does not exist in the human host, which uses the mevalonate pathway to produce IPP. The non-mevalonate pathway was highly enriched with genes that were predicted to be essential. The reactions are given by their EC-numbers and the gene symbols of the genes of the corresponding enzymes.

Table 3.13: Novel potential drug targets for *S. typhimurium* from the non-mevalonate pathway

ORF id	Gene	EC-number	Enzyme	Evidence	Human homologs	E-value
STM0049	ispH, lytB	1.17.1.2	4-hydroxy-3-methylbut-2-enyl diphosphate reductase	[38]	ENST00000408450	0.41
STM0220	dxr	1.1.1.267	1-deoxy-D-xylulose 5-phosphate reductoisomerase	[150]	ENST00000384092	2
STM0422	dxs	2.2.1.7	1-deoxy-D-xylulose-5-phosphate synthase	[105]	ENST00000387061	3
STM0423	ispA	2.5.1.10	geranyltranstransferase	[153]	ENST00000410444	1.3
STM1779	ispE, ipk	2.7.1.149	4-diphosphocytidyl-2-C-methyl-D-erythritol kinase	[38, 76, 148]	ENST00000386764	5.5
STM2523	ispG, gcpE	1.17.7.1	4-hydroxy-3-methylbut-2-en-1-yl diphosphate synthase	[38, 136]	ENST00000410400	7.2
STM2929	ispF	4.6.1.12	2-C-methyl-D-erythritol 2,4-cyclodiphosphate synthase	[38, 76]	ENST00000384847	12
STM2930	ispD	2.7.7.60	2-C-methyl-D-erythritol 4-phosphate cytidylyltransferase	[38]	ENST00000364025	0.07

tion and phyletic retention. An organism-wise cross-validation on bacterial species yielded reliable results with good accuracies (area under the receiver-operator-curve of 75% - 81%). Finally, the procedure was applied to drug target predictions for *S. typhimurium*. We compared our predictions to the viability of experimental knockouts of *S. typhimurium* and identified 35 enzymes that are highly relevant to be considered as potential drug targets. Specifically, we detected promising drug targets in the non-mevalonate pathway. Using elaborated features characterizing network topology, sequence information and microarray data enables us to predict essential genes from a bacterial reference organism to a related query organism without any knowledge about the essentiality of genes of the query organism. In general, such a method is beneficial for inferring drug targets when experimental data from genome-wide knockout screens is not available for the investigated organism.

Chapter 4

Discussion

4.1 Summary and discussion

In this thesis, I describe our development of a graph-based analysis system for drug target identification in the metabolic network of microorganisms. Experimental data from knockout strains were employed to set up a machine learning system that integrates a variety of features describing network topology and functional genomic properties in an elaborated way.

The graph-based investigation of the mutated network
We developed a new graph-based investigating tool, named "producibility," and we showed that this approach performed well in the prediction of potential drug targets. For *P. falciparum*, we used the target reactions of approved drugs as a gold standard. In comparison to an established choke point analysis, we yielded more precise predictions when combining our method with the established approach of choke point analysis. This makes sense, because in addition to the choke point analysis, we carefully checked for alternative pathways that could generate the products of the tested knockout reactions. However, our method alone yielded a lower precision and accuracy compared to the combined approach. This may be due to the fact that our algorithm simply searches for any kind of deviation in the network that biochemically may not always serve as a valid replacement. This is especially the case when analyzing higher connected reactions. This finding indicates that a combination of both approaches should be applied; assembling the union of the predictions from both methods yielded a good sensitivity (82%), whereas their intersection yielded an improved precision (29%) and a good accuracy of 88%. However, the precision

of 29% was also not very high. The reason for this is two-fold: first, we needed to improve our algorithm, and, second, some of our false positive predictions were indeed valid new drug targets. With at least reasonable evidence, this could be experimentally shown for half of these targets. In addition, for the KEIO knockout collection for *E. coli*, in which every open reading frame was knocked out and tested for its essentiality, around 300 ORFs were observed to be essential [12]. We mapped these ORFs to their corresponding reactions and found around 100 metabolic reactions to be essential in *E. coli*. Transferring this order of magnitude to our case, we estimate that only about half of all essential reactions in *P. falciparum* have been experimentally validated and targeted thus far. In comparison to the choke point analysis, we yielded a rather low sensitivity (our method: 32%, choke point: 74%, combined: 24%). However, it was not our aim to predict a large list of potential targets with a high number of false positives. We calculated a smaller defined list of concrete candidate targets to be tested in the lab. For our concept, no initial settings are needed. This makes its application easier in comparison to that of flux balance analysis, for which the environmental conditions need to be defined, such as the availability of nutrients, the carbon sources and the temperature. In contrast, we restricted our method to scan over the local topological properties of the network around the investigated reaction. Such a concept may be combined with a flux balance analysis by defining the investigated region of the network and restricting it to local subnets, making FBA more independent from environmental settings.

In summary, this approach is computationally inexpensive and simple to implement, and it limits the time and costs associated with wet-lab experiments. Also, it has the potential to serve as a valid technique in combination with other established graph-based investigations of metabolism. The ability to detect essential nodes in the network depends on how well the network of interest was reconstructed. Specifically, for *P. falciparum*, there is a larger gap in information and the reconstructed networks were far from being complete. Still, it is necessary to have *ad hoc* solutions that tackle the increasing demand for drug targets for the severe diseases caused by this parasite.

Machine learning based approach

A single feature describing the topology may often not yield a good essentiality estimate, but intelligently combining these features can yield a far more comprehensive model. Thus, we used a machine learning concept integrating various network topological, genomic and transcriptomic features for two applications. The first application involved training the machine to identify the essential genes and reactions of *E. coli*. For the gold standard, we used the KEIO collection, which is a comprehensive experimental dataset comprising phenotypic outcomes from single knockout mutants of almost every open reading frame of *E. coli* [12]. The trained machine ac-

curately predicted the experimental outcomes, which were, in the case of essentiality under the glucose minimal condition, comparable to the findings the flux balance analysis. Thus, the approach can, in principle, handle all media conditions. Rich medium conditions may better reflect the conditions experienced by pathogens in their hosts. Flux balance analysis needs clearly defined nutrient compositions, but providing these compositions can be difficult in such less-defined media as the gut of the host. An advantage of machine learning approaches is the ability to easily change the stringency parameter; for example, to increase precision to avoid losing potential candidates, the weight factor for the positive instances can be increased.

The predictions of the trained classifier were used to detect errors in the experimental knockout screen. A difference between the experimental data and the *in silico* predictions may either be due to an erroneous prediction by the algorithm or an error in the knockout experiment. Thus, genes that were predicted as false positives and false negatives were experimentally re-investigated. Five out of the 6 selected false positive genes were found to be not correctly knocked out, and 9 out of the 33 false negative genes were found to not be essential. In the intersection of our results and the KEIO collection, we found 37 potential targets for novel drugs; for 19 of these targets, we could find some reported experimental evidence in the literature.

Another application of the developed method is the training of the machine with one organism (*e.g.*, *E. coli*) to predict essential genes for another organism for which an experimental high-throughput screen may be too expensive, *e.g.*, in the case of a pathogen that needs elaborate control and isolation conditions in the laboratory. We used five experimental datasets from high-throughput knockout screens, two sets of different studies of *E. coli* and two sets of different studies of *P. aeruginosa*. Similar to *E. coli* in the gut, *P. aeruginosa* is an abundant bacterium in the soil. Additionally, we employed a smaller, non-comprehensive dataset from *S. typhimurium* in which 53 knockouts were described to be essential. The classifiers were trained with essentiality information for the genes of one organism (*e.g.*, *E. coli*) and were employed to predict essential genes for the other organism (*e.g.*, *P. aeruginosa*). These predictions did not depend on the essentiality information of the query organism for which the predictions were made, but solely on features that were calculated from the metabolic network and genomic and transcriptomic information of the query organism. Such data are abundantly available for many pathogenic bacteria. We applied the trained machines to predict the essential genes of *S. typhimurium* as the query organism of interest and proposed 35 potential drug targets. Twenty-seven targets resulted from the intersection between our predictions and an experimental study [93], and 8 targets were found in the non-mevalonate pathway through a statistical enrichment analysis. The non-mevalonate pathway is non-existent in the human host which makes it very attractive for designing a specific treatment. Some

of the enzymes of this pathway are known targets for other pathogenic microorganisms [125]. We discovered interesting correlations between our features and the essentiality of a gene. Various features describing the network topology allowed the machine to select reactions that showed no possible pathways for flux deviations, *e.g.*, in the linear non-mevalonate pathway.

Through this analysis, we gained three valuable insights. First, we could see that the topologic, genomic and transcriptomic data describing the network attributes were sufficient for defining the essentiality of a certain reaction under all media conditions. Secondly, the method could be used to validate the experimental knockout screens by means of reducing the number of false, experimentally obtained class labels, specifically for the positive predictions; this supports the estimation of potential drug targets. Finally, the method performed well when inferring the essentiality information for an organism from another, related organism.

Feature analyses

Most of our network descriptors aimed at discovering weak points in a metabolic network. Because they qualifying the uniqueness of the consumption or production of a metabolite, choke points have been reported to be well-suited to the detection of possible drug targets [131]. However, some false positive targets may be identified because of gaps in an incomplete metabolic network that come along with some dead-end reactions. After deleting an investigated enzyme in the network, an estimation of damages is also suitable for detecting possible drug targets [100]. The results of these damage analyses support the idea of network robustness because the removal of the majority of the enzymes, when individually deleted, exerts little damage to the network. It is worth noting that damage analyses may yield many false positives, specifically when investigating reversible reactions. We used four different measures for defining central nodes in the networks. As global centrality measures concerning the whole network, betweenness, closeness and eccentricity centralities were positively correlated with essentiality, while eigenvector centrality, which describes local connectivity, was negatively correlated. Both types of centrality measures (local and global) help to detect flux deviations. The eigenvector centrality detects deviations for hubs locally because high eigenvector centrality indicates highly connected adjacent nodes, which raises the possibility for flux deviations. This explains the negative correlation to essentiality. In contrast, high values for the other centralities indicate the central position of the node in the whole network and therefore suggests that a node is more likely to be essential. For example, high betweenness centrality indicates that a node is likely to participate in many paths of any pairs of nodes in the network.

Our genomic and transcriptomic features aim to analyze similar properties and functions for essential genes. The genomic sequence is not be the limiting factor for

most applications because a remarkable number of genomes have been sequenced or will be sequenced in the near future. Furthermore, our approach uses unspecific gene expression data, which can be obtained from publicly available resources or from straightforward experiments. For example, for *E. coli*, we used gene expression data from wildtype and single knockout strains. The single knockouts targeted regulators of respiration affecting a large number of genes and the treatment was rather unspecific (growth in oxygen-rich and oxygen-deprived conditions). Hence, a large portion of the network pathways of the metabolic network were differentially expressed [143]. Within the presented approach, data from such pathway-unspecific examinations allowed the classifier to learn which neighboring enzymes jointly worked together. Therefore, multiple gene co-expression datasets for a variety of conditions may be well-suited to our approach. However, which type of gene expression data would optimize performance still needs to be determined.

In conclusion, we developed a machine learning approach for *in silico* predictions of drug targets. It intelligently combined all these features and may be seen as an alternative approach to the established methods of flux balance analysis (FBA) and elementary flux modes (EFM) if detailed growth and nutrient information are lacking (which is needed for FBA [144]) and if an in-depth refinement of the metabolic network is considered to be too labor intensive (in EFM, the enzymes need to be separated into internal nodes and external nodes to reduce the computational complexity [41]). For pathogens, it is often hard to define these environmental parameters, which are complex and changeable (*e.g.*, for intestinal infections). The machine learning approach described in this thesis can, in principle, handle various environmental conditions without detailed specification, but it may benefit from experimental essentiality screens that performed under similar environmental conditions as those of the application (*e.g.*, in oxygen-deprived conditions when mimicking the environmental conditions of the gut).

4.2 Outlook

A machine learning algorithm combining descriptors of network topology for predicting essential genes can be broadly applied to systems seeking potential drug targets for a variety of important bacterial and other pathogenic infections. Until now, most studies investigated the prediction of essential nodes for the same organism, and the inference of essentiality information from one organism for another organism has been achieved only for closely-related strains of an organism (yeast). We extended this approach to different, but related, organisms (*E. coli*, *P. aeruginosa* and *S. typhimurium*). It will be a challenge to open this up to a wider variety of organisms for training and application and to use the descriptors for other applications,

specifically for multicellular organisms and human cells, *e.g.*, to predict potential driver mutations in cancer or host factors for viruses. To apply this method to other microorganisms, the metabolic pathways may need to be well-characterized, and the method may need to be adapted for less studied organisms or those with special metabolic capabilities. To apply this method to eukaryotic genomes, the compartments in the cell where a reaction occurs may need to be considered. Such a prediction across more distant organisms will be interesting, specifically with respect to studying the differences between conserved and evolved genes.

Because the parasite may employ host enzymes, network reconstructions considering the network of the host may improve target predictions for parasitic organisms. We demonstrated this concept in the context of *P. falciparum* in the human blood cell. But such a reconstruction can be improved using more detailed experimental information about host-parasite interactions and metabolic exchanges. Furthermore, because genes are expressed differentially, such as during the cell cycle or under different conditions, it will be important to dynamically analyze conditionally-specific networks as opposed to using a static network. Moreover, it will be very challenging to apply our method to infer multiple drug targets using experimental double knockout screens, *e.g.*, in the case of the synthetic lethal project of eSGA [31]. For this, attributes that are related to single players (*e.g.*, sequence features) might be of less relevance, while specific network features might be more relevant to synergistic knockout effects. In the future, it will be critical to integrate such topological descriptive approaches with genetic information to systematically explore the network effects of enzyme treatments and combinations thereof in a realistic and dynamic environment.

References

[1] Acencio, M. L. and N. Lemke: *Towards the prediction of essential genes by integration of network topology, cellular localization and biological process information.* BMC Bioinformatics, 10:290, 2009.

[2] Albert, R. and A. L. Barabasi: *Statistical mechanics of complex networks.* Reviews of Modern Physics, 74(1):47–97, 2002. 533UL Times Cited:4589 Cited References Count:214.

[3] Albert, R., H. Jeong and A. L. Barabasi: *Error and attack tolerance of complex networks.* Nature, 406(6794):378–382, 2000. 337WC Times Cited:1561 Cited References Count:24.

[4] Alberts, B., A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter: *Molecular biology of the cell.* Garland Science, 5. edition, 2008.

[5] Almaas, E.: *Biological impacts and context of network theory.* Journal of Experimental Biology, 210(9):1548–1558, 2007. 165JN Times Cited:31 Cited References Count:68.

[6] Almiron, M., M. Martinez, N. Sanjuan and R. A. Ugalde: *Ferrochelatase is present in Brucella abortus and is critical for its intracellular survival and virulence.* Infect Immun, 69(10):6225–30, 2001.

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

[8] Amsden, G. W.: *Pneumococcal resistance in perspective: how well are we combating it?* Pediatr Infect Dis J, 23(2 Suppl):S125–8, 2004. Comparative Study Journal Article Review United States.

[9] Apfalter, P.: *[MRSA/MRSE-VISA/GISA/VRSA-PRP-VRE: current gram positive problem bacteria and mechanism of resistance, prevalence and clinical consequences].* Wien Med Wochenschr, 153(7-8):144–7, 2003. Comparative Study English Abstract Journal Article Review Austria.

[10] Archer, GL. and RE. Polk: *Treatment and prophylaxis of bacterial infections.* Harrison's Principles of Internal Medicine. McGraw-Hill, New York, 16 edition, 2004.

[11] Arita, M.: *Scale-freeness and biological networks.* J Biochem, 138(1):1–4, 2005. Arita, Masanori Research Support, Non-U.S. Gov't Review Japan Journal of biochemistry J Biochem. 2005 Jul;138(1):1-4.

[12] Baba, T., T. Ara, M. Hasegawa, Y. Takai, Y. Okumura, M. Baba, K. A. Datsenko, M. Tomita, B. L. Wanner and H. Mori: *Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: the Keio collection.* Mol Syst Biol, 2:2006 0008, 2006.

[13] Baggett, H. C., T. W. Hennessy, K. Rudolph, D. Bruden, A. Reasonover, A. Parkinson, R. Sparks, R. M. Donlan, P. Martinez, K. Mongkolrattanothai and J. C. Butler: *Community-onset methicillin-resistant Staphylococcus aureus associated with antibiotic use and the cytotoxin Panton-Valentine leukocidin during a furunculosis outbreak in rural Alaska.* J Infect Dis, 189(9):1565–73, 2004. Journal Article Research Support, U.S. Gov't, P.H.S. United States.

[14] Barabasi, A. L. and R. Albert: *Emergence of scaling in random networks.* Science, 286(5439):509–512, 1999. 245RD Times Cited:5576 Cited References Count:22.

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

[16] Barie, P. S.: *Antibiotic-resistant gram-positive cocci: implications for surgical practice.* World J Surg, 22(2):118–26, 1998. Journal Article Review United states.

[17] Batada, N. N., L. D. Hurst and M. Tyers: *Evolutionary and physiological importance of hub proteins.* PLoS Comput Biol, 2(7):e88, 2006. Journal Article Research Support, Non-U.S. Gov't United States.

[18] Baum, E. Z., S. M. Crespo-Carbone, A. Klinger, B. D. Foleno, I. Turchi, M. Macielag and K. Bush: *A MurF inhibitor that disrupts*

cell wall biosynthesis in Escherichia coli. Antimicrob Agents Chemother, 51(12):4420–6, 2007. Journal Article United States.

[19] Beck, B. J., M. Huelsmeyer, S. Paul and D. M. Downs: *A mutation in the essential gene gmk (encoding guanlyate kinase) generates a requirement for adenine at low temperature in Salmonella enterica.* J Bacteriol, 185(22):6732–5. GM47296/GM/NIGMS NIH HHS/United States Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, P.H.S. United States.

[20] Becker, S. A., A. M. Feist, M. L. Mo, G. Hannum, B. O. Palsson and M. J. Herrgard: *Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox.* Nat Protoc, 2(3):727–38, 2007.

[21] Belunis, C. J., T. Clementz, S. M. Carty and C. R. Raetz: *Inhibition of lipopolysaccharide biosynthesis and cell growth following inactivation of the kdtA gene in Escherichia coli.* J Biol Chem, 270(46):27646–52, 1995.

[22] Bence, A. K. and P. A. Crooks: *The mechanism of L-canavanine cytotoxicity: arginyl tRNA synthetase as a novel target for anticancer drug discovery.* J Enzyme Inhib Med Chem, 18(5):383–94, 2003. Journal Article Review England.

[23] Bonacich, P.: *Factoring and weighting approaches to status scores and clique identification.* Journal of Mathematical Sociology, 2:113–120, 1972.

[24] Bonacich, P.: *Power and Centrality: A Family of Measures.* American Journal of Sociology, 92(5):1170–82, 1987.

[25] Bonday, Z. Q., S. Dhanasekaran, P. N. Rangarajan and G. Padmanaban: *Import of host delta-aminolevulinate dehydratase into the malarial parasite: identification of a new drug target.* Nat Med, 6(8):898–903, 2000.

[26] Bonten, M. J., R. Willems and R. A. Weinstein: *Vancomycin-resistant enterococci: why are they here, and where do they come from?* Lancet Infect Dis, 1(5):314–25, 2001. Journal Article Review United States.

[27] Borer, A., J. Gilad, P. Yagupsky, N. Peled, N. Porat, R. Trefler, H. Shprecher-Levy, K. Riesenberg, M. Shipman and F. Schlaeffer: *Community-acquired methicillin-resistant Staphylococcus aureus in institutionalized adults with developmental disabilities.* Emerg Infect Dis, 8(9):966–70, 2002. Journal Article United States.

[28] Bratkovic, T., M. Lunder, U. Urleb and B. Strukelj: *Peptide inhibitors of MurD and MurE, essential enzymes of bacterial cell wall biosynthesis*. J Basic Microbiol, 48(3):202–6, 2008.

[29] Bunjun, S., C. Stathopoulos, D. Graham, B. Min, M. Kitabatake, A. L. Wang, C. C. Wang, C. P. Vivares, L. M. Weiss and D. Soll: *A dual-specificity aminoacyl-tRNA synthetase in the deep-rooted eukaryote Giardia lamblia*. Proc Natl Acad Sci U S A, 97(24):12997–3002, 2000. R01 AI031788-09/AI/NIAID NIH HHS/United States Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. Research Support, U.S. Gov't, P.H.S. United states.

[30] Burges, C.J.C.: *A tutorial on support vector machines for pattern recognition*. In *Data Mining and Knownledge Discovery*, volume 2, pages 121–167. Kluwer Academic Publishers, Boston, 1998.

[31] Butland, G., M. Babu, J. J. Diaz-Mejia, F. Bohdana, S. Phanse, B. Gold, W. Yang, J. Li, A. G. Gagarinova, O. Pogoutse, H. Mori, B. L. Wanner, H. Lo, J. Wasniewski, C. Christopolous, M. Ali, P. Venn, A. Safavi-Naini, N. Sourour, S. Caron, J. Y. Choi, L. Laigle, A. Nazarians-Armavil, A. Deshpande, S. Joe, K. A. Datsenko, N. Yamamoto, B. J. Andrews, C. Boone, H. Ding, B. Sheikh, G. Moreno-Hagelseib, J. F. Greenblatt and A. Emili: *eSGA: E. coli synthetic genetic array analysis*. Nat Methods, 5(9):789–95, 2008.

[32] Caspi, R., H. Foerster, C. A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Y. Rhee, A. G. Shearer, C. Tissier, T. C. Walk, P. Zhang and P. D. Karp: *The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases*. Nucleic Acids Res, 36(Database issue):D623–31, 2008.

[33] Chang, C.C. and C. C. Lin: *LIBSVM: a library for support vector machines*, 2001.

[34] Chen, Y. and D. Xu: *Understanding protein dispensability through machine-learning analysis of high-throughput data*. Bioinformatics, 21(5):575–81, 2005. Evaluation Studies Journal Article Research Support, U.S. Gov't, Non-P.H.S. England.

[35] Chopra, I.: *Bacterial RNA polymerase: a promising target for the discovery of new antimicrobial agents*. Curr Opin Investig Drugs, 8(8):600–7, 2007.

[36] Chou, H. T., D. H. Kwon, M. Hegazy and C. D. Lu: *Transcriptome analysis of agmatine and putrescine catabolism in Pseudomonas aeruginosa PAO1.* J Bacteriol, 190(6):1966–75, 2008.

[37] Conte, JE.: *Treatment and Prevention.* Manual of Antibiotics and Infectious Diseases. Lippincott Williams & Wilkins, Philadelphia, 9th edition, 2002.

[38] Cornish, R. M., J. R. Roth and C. D. Poulter: *Lethal mutations in the isoprenoid pathway of Salmonella enterica.* J Bacteriol, 188(4):1444–50, 2006.

[39] Covert, M. W., E. M. Knight, J. L. Reed, M. J. Herrgard and B. O. Palsson: *Integrating high-throughput and computational data elucidates bacterial networks.* Nature, 429(6987):92–6, 2004.

[40] Crabb, B. S., T. F. de Koning-Ward and P. R. Gilson: *Toward forward genetic screens in malaria-causing parasites using the piggyBac transposon.* BMC Biol, 9:21, 2011. Editorial Research Support, Non-U.S. Gov't England.

[41] Dandekar, T., F. Moldenhauer, S. Bulik, H. Bertram and S. Schuster: *A method for classifying metabolites in topological pathway analyses based on minimization of pathway number.* Biosystems, 70(3):255–70, 2003.

[42] De Lencastre, H., S. W. Wu, M. G. Pinho, A. M. Ludovice, S. Filipe, S. Gardete, R. Sobral, S. Gill, M. Chung and A. Tomasz: *Antibiotic resistance as a stress response: complete sequencing of a large number of chromosomal loci in Staphylococcus aureus strain COL that impact on the expression of resistance to methicillin.* Microb Drug Resist, 5(3):163–75, 1999. Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, P.H.S. United states.

[43] Delport, R., J. B. Ubbink, H. Bosman, S. Bissbort and W. J. Vermaak: *Altered vitamin B6 homeostasis during aminophylline infusion in the beagle dog.* Int J Vitam Nutr Res, 60(1):35–40, 1990. Journal Article Research Support, Non-U.S. Gov't Switzerland.

[44] Detweiler, C. S., D. M. Monack, I. E. Brodsky, H. Mathew and S. Falkow: *virK, somA and rcsC are important for systemic Salmonella enterica serovar Typhimurium infection and cationic peptide resistance.* Mol Microbiol, 48(2):385–400, 2003.

[45] Dimitriadou, Evgenia, Kurt Hornik, Friedrich Leisch, David Meyer and Andreas Weingessel: *Misc Functions of the Department of Statistic (e1071), TU Wien.* 2006.

[46] Dini, C.: *MraY Inhibitors as Novel Antibacterial Agents.* Curr Top Med Chem, 5(13):1221–36, 2005. Journal Article Review Netherlands.

[47] Domin, M. A.: *Highly virulent pathogens–a post antibiotic era?* Br J Theatre Nurs, 8(2):14–8, 1998. Journal Article Review England NATNews : the official journal of the National Association of Theatre Nurses.

[48] Dzeroski, S., P. Panov and B. Zenko: *Machine Learning, Ensemble Methods in.* Encyclopedia of Complexity and Systems Science, pages 5317–5325, 2009.

[49] Edwards, J. S. and B. O. Palsson: *Metabolic flux balance analysis and the in silico analysis of Escherichia coli K-12 gene deletions.* BMC Bioinformatics, 1:1, 2000.

[50] Erdös, P. and A. Rényi: *On Evolution of Random Graphs.* Publication Math. Inst. of the Hungarian Academy Science, 5:17–61, 1960.

[51] Estrada, E.: *Virtual identification of essential proteins within the protein interaction network of yeast.* Proteomics, 6(1):35–40, 2006.

[52] Fan, F. and D. McDevitt: *Microbial Genomics for Antibiotic Target Discovery.* In *METHODS IN MICROBIOLOGY*, volume 33. Elsevier Science Ltd, 2002.

[53] Fatumo, S., K. Plaimas, E. Adebiyi and R. Konig: *Comparing metabolic network models based on genomic and automatically inferred enzyme information from Plasmodium and its human host to define drug targets in silico.* Infect Genet Evol, 11(1):201–8, 2011. Comparative Study Journal Article Research Support, Non-U.S. Gov't Netherlands journal of molecular epidemiology and evolutionary genetics in infectious diseases.

[54] Fatumo, S., K. Plaimas, J. P. Mallm, G. Schramm, E. Adebiyi, M. Oswald, R. Eils and R. Konig: *Estimating novel potential drug targets of Plasmodium falciparum by analysing the metabolic network of knock-out strains in silico.* Infect Genet Evol, 9(3):351–8, 2009.

[55] Feist, A. M., C. S. Henry, J. L. Reed, M. Krummenacker, A. R. Joyce, P. D. Karp, L. J. Broadbelt, V. Hatzimanikatis and B. O. Palsson: *A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information.* Mol Syst Biol, 3:121, 2007.

[56] Feist, A. M., M. J. Herrgard, I. Thiele, J. L. Reed and B. O. Palsson: *Reconstruction of biochemical networks in microorganisms.* Nat Rev Microbiol, 7(2):129–43, 2009.

[57] Fujishima, H., A. Nishimura, M. Wachi, H. Takagi, T. Hirasawa, H. Teraoka, K. Nishimori, T. Kawabata, K. Nishikawa and K. Nagai: *kdsA mutations affect FtsZ-ring formation in Escherichia coli K-12.* Microbiology, 148(Pt 1):103–12, 2002. Journal Article Research Support, Non-U.S. Gov't England.

[58] Garrett, T. A., N. L. Que and C. R. Raetz: *Accumulation of a lipid A precursor lacking the 4'-phosphate following inactivation of the Escherichia coli lpxK gene.* J Biol Chem, 273(20):12457–65, 1998. GM-51310/GM/NIGMS NIH HHS/United States Journal Article Research Support, U.S. Gov't, Non-P.H.S. Research Support, U.S. Gov't, P.H.S. United states.

[59] Gerdes, S., R. Edwards, M. Kubal, M. Fonstein, R. Stevens and A. Osterman: *Essential genes on metabolic maps.* Curr Opin Biotechnol, 17(5):448–56, 2006. Gerdes, Svetlana Edwards, Robert Kubal, Michael Fonstein, Michael Stevens, Rick Osterman, Andrei 1-R01-AI059146-01A2/AI/NIAID NIH HHS/United States HHSN266200400042C/PHS HHS/United States Research Support, N.I.H., Extramural Review England Current opinion in biotechnology Curr Opin Biotechnol. 2006 Oct;17(5):448-56. Epub 2006 Sep 15.

[60] Gerdes, S. Y., M. D. Scholle, J. W. Campbell, G. Balazsi, E. Ravasz, M. D. Daugherty, A. L. Somera, N. C. Kyrpides, I. Anderson, M. S. Gelfand, A. Bhattacharya, V. Kapatral, M. D'Souza, M. V. Baev, Y. Grechkin, F. Mseeh, M. Y. Fonstein, R. Overbeek, A. L. Barabasi, Z. N. Oltvai and A. L. Osterman: *Experimental determination and system level analysis of essential genes in Escherichia coli MG1655.* J Bacteriol, 185(19):5673–84, 2003.

[61] Ginsburg, H.: *Progress in in silico functional genomics: the Malaria Metabolic Pathways database.* Trends Parasitol, 22(6):238–40, 2006. Journal Article Research Support, Non-U.S. Gov't England.

[62] Grocock, R. J. and P. M. Sharp: *Synonymous codon usage in Pseudomonas aeruginosa PA01.* Gene, 289(1-2):131–9, 2002.

[63] Gursoy, A., O. Keskin and R. Nussinov: *Topological properties of protein interaction networks from a structural perspective.* Biochem Soc Trans, 36(Pt 6):1398–403, 2008.

[64] Gustafson, A. M., E. S. Snitkin, S. C. Parker, C. DeLisi and S. Kasif: *Towards the identification of essential genes using targeted genome sequencing and comparative analysis*. BMC Genomics, 7:265, 2006.

[65] Hahn, M. W. and A. D. Kern: *Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks*. Mol Biol Evol, 22(4):803–6, 2005.

[66] Hall, MA. and LA. Smith: *Practical feature subset selection for machine learning*. In *Processings of the 21st Australian Computer Science Conference*, 1998.

[67] Helm, J. S., Y. Hu, L. Chen, B. Gross and S. Walker: *Identification of active-site inhibitors of MurG using a generalizable, high-throughput glycosyltransferase screen*. J Am Chem Soc, 125(37):11168–9, 2003.

[68] Hopkins, A. L. and C. R. Groom: *The druggable genome*. Nat Rev Drug Discov, 1(9):727–30, 2002.

[69] Hosseinzadeh, H., B.S.F. Bazzaz and M.M. Sadati: *In vitro evaluation of methylxanthines and some antibiotics: interaction against Staphylococcus aureus and Pseudomonas aeruginosa*. Iranian Biomed. J., 10:163–167, 2006.

[70] Hsu, C.W., C.C. Chang and C.J. Lin: *A Practical Guide to Support Vector Classification*, 2010.

[71] Hubbard, T. J., B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, J. Herrero, R. Holland, K. Howe, K. Howe, N. Johnson, A. Kahari, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, C. Melsopp, K. Megy, P. Meidl, B. Ouverdin, A. Parker, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, J. Severin, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, M. Wood, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, A. Kasprzyk, G. Proctor, S. Searle, J. Smith, A. Ureta-Vidal and E. Birney: *Ensembl 2007*. Nucleic Acids Res, 35(Database issue):D610–7, 2007.

[72] Huber, W., A. von Heydebreck, H. Sultmann, A. Poustka and M. Vingron: *Variance stabilization applied to microarray data calibration and to the quantification of differential expression*. Bioinformatics, 18 Suppl 1:S96–104, 2002.

[73] Hunter, W. N.: *The non-mevalonate pathway of isoprenoid precursor biosynthesis.* J Biol Chem, 282(30):21573–7, 2007.

[74] Huthmacher, C., C. Gille and H. G. Holzhutter: *A computational analysis of protein interactions in metabolic networks reveals novel enzyme pairs potentially involved in metabolic channeling.* J Theor Biol, 252(3):456–64, 2008. Journal Article Netherlands.

[75] Hwang, Y. C., C. C. Lin, J. Y. Chang, H. Mori, H. F. Juan and H. C. Huang: *Predicting essential genes based on network and sequence analysis.* Mol Biosyst, 2009.

[76] Illarionova, V., J. Kaiser, E. Ostrozhenkova, A. Bacher, M. Fischer, W. Eisenreich and F. Rohdich: *Nonmevalonate terpene biosynthesis enzymes as antiinfective drug targets: substrate synthesis and high-throughput screening methods.* J Org Chem, 71(23):8824–34, 2006.

[77] Isabelle, G., J. Weston, S. Barnhill and V. Vapnik: *Gene selection for cancer classification using Support Vector Machines.* Machine Learning, 46:389–422, 2002.

[78] Jackson, M., D. C. Crick and P. J. Brennan: *Phosphatidylinositol is an essential phospholipid of mycobacteria.* J Biol Chem, 275(39):30092–9, 2000. AI18357/AI/NIAID NIH HHS/United States AI46393/AI/NIAID NIH HHS/United States Journal Article Research Support, U.S. Gov't, P.H.S. United states.

[79] Jacobs, M. A., A. Alwood, I. Thaipisuttikul, D. Spencer, E. Haugen, S. Ernst, O. Will, R. Kaul, C. Raymond, R. Levy, L. Chun-Rong, D. Guenthner, D. Bovee, M. V. Olson and C. Manoil: *Comprehensive transposon mutant library of Pseudomonas aeruginosa.* Proc Natl Acad Sci U S A, 100(24):14339–44, 2003.

[80] Jelakovic, S. and G. E. Schulz: *The structure of CMP:2-keto-3-deoxy-manno-octonic acid synthetase and of its complexes with substrates and substrate analogs.* J Mol Biol, 312(1):143–55, 2001.

[81] Jeong, H., S. P. Mason, A. L. Barabasi and Z. N. Oltvai: *Lethality and centrality in protein networks.* Nature, 411(6833):41–2, 2001.

[82] Ji, Y., B. Zhang, S. F. Van, Horn, P. Warren, G. Woodnutt, M. K. Burnham and M. Rosenberg: *Identification of critical staphylococcal genes using conditional phenotypes generated by antisense RNA.* Science, 293(5538):2266–9, 2001.

[83] Jordan, I. K., I. B. Rogozin, Y. I. Wolf and E. V. Koonin: *Essential genes are more evolutionarily conserved than are nonessential genes in bacteria.* Genome Res, 12(6):962–8, 2002. Journal Article United States.

[84] Jovanovic, M., M. Lilic, R. Janjusevic, G. Jovanovic and D. J. Savic: *tRNA synthetase mutants of Escherichia coli K-12 are resistant to the gyrase inhibitor novobiocin.* J Bacteriol, 181(9):2979–83, 1999. Journal Article Research Support, Non-U.S. Gov't United states.

[85] Joyce, A. R., J. L. Reed, A. White, R. Edwards, A. Osterman, T. Baba, H. Mori, S. A. Lesely, B. O. Palsson and S. Agarwalla: *Experimental and computational assessment of conditionally essential genes in Escherichia coli.* J Bacteriol, 188(23):8259–71, 2006.

[86] Junker, B.H. and Falk Schreiber: *Analysis of Biological Networks.* Wiley Series in Bioinformatics. Wiley-Interscience, 2008.

[87] Kanehisa, M., M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu and Y. Yamanishi: *KEGG for linking genomes to life and the environment.* Nucleic Acids Res, 36(Database issue):D480–4, 2008.

[88] Karita, M., M. L. Etterbeek, M. H. Forsyth, M. K. Tummuru and M. J. Blaser: *Characterization of Helicobacter pylori dapE and construction of a conditionally lethal dapE mutant.* Infect Immun, 65(10):4158–64, 1997. R01DK50837/DK/NIDDK NIH HHS/United States Comparative Study Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. Research Support, U.S. Gov't, P.H.S. United states.

[89] Kedar, G. C., V. Brown-Driver, D. R. Reyes, M. T. Hilgers, M. A. Stidham, K. J. Shaw, J. Finn and R. J. Haselbeck: *Evaluation of the metS and murB loci for antibiotic discovery using targeted antisense RNA expression analysis in Bacillus anthracis.* Antimicrob Agents Chemother, 51(5):1708–18, 2007. R44 AI053009/AI/NIAID NIH HHS/United States Journal Article Research Support, N.I.H., Extramural United States.

[90] Keseler, I. M., C. Bonavides-Martinez, J. Collado-Vides, S. Gama-Castro, R. P. Gunsalus, D. A. Johnson, M. Krummenacker, L. M. Nolan, S. Paley, I. T. Paulsen, M. Peralta-Gil, A. Santos-Zavaleta, A. G. Shearer and P. D. Karp: *EcoCyc: a comprehensive view of Escherichia coli biology.* Nucleic Acids Res, 37(Database issue):D464–70, 2009. GM071962/GM/NIGMS NIH HHS/United States

GM077678/GM/NIGMS NIH HHS/United States GM75742/GM/NIGMS NIH HHS/United States Journal Article Research Support, N.I.H., Extramural England.

[91] Kim, C. C. and S. Falkow: *Significance analysis of lexical bias in microarray data*. BMC Bioinformatics, 4:12, 2003.

[92] Kitagawa, H., K. Kumura, S. Takahata, M. Iida and K. Atsumi: *4-Pyridone derivatives as new inhibitors of bacterial enoyl-ACP reductase FabI*. Bioorg Med Chem, 15(2):1106–16, 2007. Journal Article England.

[93] Knuth, K., H. Niesalla, C. J. Hueck and T. M. Fuchs: *Large-scale identification of essential Salmonella genes by trapping lethal insertions*. Mol Microbiol, 51(6):1729–44, 2004.

[94] König, R. and R. Eils: *Gene expression analysis on biochemical networks using the Potts spin model*. Bioinformatics, 20(10):1500–5, 2004.

[95] Koschützki, Dirk and Falk Schreiber: *Comparison of Centralities for Biological Networks*. In *Proc German Conf Bioinformatics (GCB 2004)*, volume 53, pages 199–206. Springer-Verlag, 2004.

[96] Kotnik, M., P. S. Anderluh and A. Prezelj: *Development of novel inhibitors targeting intracellular steps of peptidoglycan biosynthesis*. Curr Pharm Des, 13(22):2283–309, 2007.

[97] Ku, M. J., H. J. Yoon, H. J. Ahn, H. W. Kim, S. H. Baek and S. W. Suh: *Crystallization and preliminary X-ray crystallographic studies of 3-deoxy-manno-octulosonate cytidylyltransferase from Haemophilus influenzae*. Acta Crystallogr D Biol Crystallogr, 59(Pt 1):180–2, 2003.

[98] Kumar, P., M. Chhibber and A. Surolia: *How pantothenol intervenes in Coenzyme-A biosynthesis of Mycobacterium tuberculosis*. Biochem Biophys Res Commun, 361(4):903–9, 2007.

[99] Kumar, S. and H. S. Banyal: *Purification and characterisation of the hexokinase of Plasmodium berghei, a murine malaria parasite*. Acta Vet Hung, 45(2):119–26, 1997. Journal Article Hungary.

[100] Lemke, N., F. Heredia, C. K. Barcellos, A. N. Dos Reis and J. C. Mombach: *Essentiality and damage in metabolic networks*. Bioinformatics, 20(1):115–9, 2004.

[101] Leon, P. E.: *Inhibition of ribozymes by deoxyribonucleotides and the origin of DNA*. J Mol Evol, 47(2):122–6, 1998.

[102] Leray, P. and P. Gallinari: *Feature selection with neural networks*. Behaviormetrika, 26(1):145–166, 1999.

[103] Liberati, N. T., J. M. Urbach, S. Miyata, D. G. Lee, E. Drenkard, G. Wu, J. Villanueva, T. Wei and F. M. Ausubel: *An ordered, nonredundant library of Pseudomonas aeruginosa strain PA14 transposon insertion mutants*. Proc Natl Acad Sci U S A, 103(8):2833–8, 2006.

[104] Luscombe, N. M., M. M. Babu, H. Yu, M. Snyder, S. A. Teichmann and M. Gerstein: *Genomic analysis of regulatory network dynamics reveals large topological changes*. Nature, 431(7006):308–12, 2004.

[105] Mao, J., H. Eoh, R. He, Y. Wang, B. Wan, S. G. Franzblau, D. C. Crick and A. P. Kozikowski: *Structure-activity relationships of compounds targeting mycobacterium tuberculosis 1-deoxy-D-xylulose 5-phosphate synthase*. Bioorg Med Chem Lett, 18(19):5320–3, 2008. Journal Article England.

[106] Martin, K. L. and T. K. Smith: *Phosphatidylinositol synthesis is essential in bloodstream form Trypanosoma brucei*. Biochem J, 396(2):287–95, 2006. 067441/Wellcome Trust/United Kingdom Journal Article Research Support, Non-U.S. Gov't England.

[107] Mazel, D., S. Pochet and P. Marliere: *Genetic characterization of polypeptide deformylase, a distinctive enzyme of eubacterial translation*. Embo J, 13(4):914–23, 1994. Journal Article Research Support, Non-U.S. Gov't England.

[108] McNeil, L. K., C. Reich, R. K. Aziz, D. Bartels, M. Cohoon, T. Disz, R. A. Edwards, S. Gerdes, K. Hwang, M. Kubal, G. R. Margaryan, F. Meyer, W. Mihalo, G. J. Olsen, R. Olson, A. Osterman, D. Paarmann, T. Paczian, B. Parrello, G. D. Pusch, D. A. Rodionov, X. Shi, O. Vassieva, V. Vonstein, O. Zagnitko, F. Xia, J. Zinner, R. Overbeek and R. Stevens: *The National Microbial Pathogen Database Resource (NMPDR): a genomics platform based on subsystem annotation*. Nucleic Acids Res, 35(Database issue):D347–53, 2007.

[109] Milgram, S.: *Small-World Problem*. Psychology Today, 1(1):61–67, 1967. Zk284 Times Cited:302 Cited References Count:3.

[110] Mitchell, T.: *Machine Learning*. Mcgraw-Hill Higher Education, 1997.

[111] Moritz, E., S. Seidensticker, A. Gottwald, W. Maier, A. Hoerauf, J. T. Njuguna and A. Kaiser: *The efficacy of inhibitors involved in spermidine metabolism in Plasmodium falciparum, Anopheles stephensi and Trypanosoma evansi.* Parasitol Res, 94(1):37–48, 2004. Journal Article Research Support, Non-U.S. Gov't Germany.

[112] Mushegian, A. R. and E. V. Koonin: *A minimal gene set for cellular life derived by comparison of complete bacterial genomes.* Proc Natl Acad Sci U S A, 93(19):10268–73, 1996. Comparative Study Journal Article United states.

[113] Ness, S. A.: *Basic microarray analysis: strategies for successful experiments.* Methods Mol Biol, 316:13–33, 2006.

[114] Newman, M. E. J., D. J. Watts and S. H. Strogatz: *Random graph models of social networks.* Proceedings of the National Academy of Sciences of the United States of America, 99:2566–2572, 2002. Suppl. 1 525RZ Times Cited:171 Cited References Count:29.

[115] Ogata, H., S. Goto, K. Sato, W. Fujibuchi, H. Bono and M. Kanehisa: *KEGG: Kyoto Encyclopedia of Genes and Genomes.* Nucleic Acids Res, 27(1):29–34, 1999.

[116] Okazaki, S., Y. Hattori and K. Saeki: *The Mesorhizobium loti purB gene is involved in infection thread formation and nodule development in Lotus japonicus.* J Bacteriol, 189(22):8347–52, 2007. Journal Article Research Support, Non-U.S. Gov't United States.

[117] Orth, J. D., I. Thiele and B. O. Palsson: *What is flux balance analysis?* Nat Biotechnol, 28(3):245–8, 2010. R01 GM057089/GM/NIGMS NIH HHS/United States Journal Article Research Support, N.I.H., Extramural United States.

[118] Osterman, A.L. and S.Y. Gerdes: *Microbial Gene Essentiality: Protocols and Bioinformatics.* Methods in Molecular Biology. Humana Press, 2008.

[119] Osuna, E. E., R. Freund and F. Girosi: *Support vector machines: Training and applications.* Massachusetts Institute of Technology, 1997.

[120] Padmanabhan, P. K., A. Mukherjee and R. Madhubala: *Characterization of the gene encoding glyoxalase II from Leishmania donovani: a potential target for anti-parasite drugs.* Biochem J, 393(Pt 1):227–34, 2006. Journal Article Research Support, Non-U.S. Gov't England.

[121] Palsson, Bernhard O.: *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, 1st edition, 2006.

[122] Park, T., S. G. Yi, S. H. Kang, S. Lee, Y. S. Lee and R. Simon: *Evaluation of normalization methods for microarray data.* BMC Bioinformatics, 4:33, 2003.

[123] Peukert, S., Y. Sun, R. Zhang, B. Hurley, M. Sabio, X. Shen, C. Gray, J. Dzink-Fox, J. Tao, R. Cebula and S. Wattanasin: *Design and structure-activity relationships of potent and selective inhibitors of undecaprenyl pyrophosphate synthase (UPPS): tetramic, tetronic acids and dihydropyridin-2-ones*. Bioorg Med Chem Lett, 18(6):1840–4, 2008.

[124] Phung, S.L., A. Bouzerdoum and G.H Nguyen: *Learning pattern classification tasks with imbalanced data sets*. pages 193–208, 2009.

[125] Plaimas, K., R. Eils and R. Konig: *Identifying essential genes in bacterial metabolic networks with machine learning methods*. BMC Syst Biol, 4:56, 2010.

[126] Plaimas, K. and R. Knig: *Machine Learning Methods for Identifying Essential Genes and Proteins in Networks*. Applied Statistics for Network Biology: Methods in Systems Biology. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, 2011.

[127] Plaimas, K., J. P. Mallm, M. Oswald, F. Svara, V. Sourjik, R. Eils and R. Konig: *Machine learning based analyses on metabolic networks supports high-throughput knockout screens*. BMC Syst Biol, 2:67, 2008.

[128] Plaimas, K., M. Oswald, R. Eils and R. Knig: *Integrating Genomic and Transcriptomic Data into Graph Based Approaches for Defining Essential Reactions in the Metabolic Network of Escherichia coli*. In *Proceedings of the Knowledge Discovery, Data Mining, and Machine learning*, pages 55–60, Halle, Germany, 2007.

[129] Przulj, N., D. A. Wigle and I. Jurisica: *Functional topology in a network of protein interactions*. Bioinformatics, 20(3):340–8, 2004.

[130] Rahman, S. A., P. Advani, R. Schunk, R. Schrader and D. Schomburg: *Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC)*. Bioinformatics, 21(7):1189–93, 2005. Comparative Study Evaluation Studies Journal Article Research Support, Non-U.S. Gov't England.

[131] Rahman, S. A. and D. Schomburg: *Observing local and global properties of metabolic pathways: 'load points' and 'choke points' in the metabolic networks.* Bioinformatics, 22(14):1767–74, 2006.

[132] Ratnam, K. and J. A. Low: *Current development of clinical inhibitors of poly(ADP-ribose) polymerase in oncology.* Clin Cancer Res, 13(5):1383–8, 2007. Journal Article Review United States an official journal of the American Association for Cancer Research.

[133] Richards, N. G. and M. S. Kilberg: *Asparagine synthetase chemotherapy.* Annu Rev Biochem, 75:629–54, 2006. CA09126/CA/NCI NIH HHS/United States CA107437/CA/NCI NIH HHS/United States DK52064/DK/NIDDK NIH HHS/United States Journal Article Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov't Review United States.

[134] Robertson, D., P. Carroll and T. Parish: *Rapid recombination screening to test gene essentiality demonstrates that pyrH is essential in Mycobacterium tuberculosis.* Tuberculosis (Edinb), 87(5):450–8, 2007.

[135] Roden, DM.: *Principles of clinical pharmacology.* Harrison's Principles of Internal Medicine. McGraw-Hill, New York, 16 edition, 2004.

[136] Rohdich, F., A. Bacher and W. Eisenreich: *Perspectives in anti-infective drug design. The late steps in the biosynthesis of the universal terpenoid precursors, isopentenyl diphosphate and dimethylallyl diphosphate.* Bioorg Chem, 32(5):292–308, 2004. Journal Article Research Support, Non-U.S. Gov't Review United States.

[137] Sacco, E., A. S. Covarrubias, H. M. O'Hare, P. Carroll, N. Eynard, T. A. Jones, T. Parish, M. Daffe, K. Backbro and A. Quemard: *The missing piece of the type II fatty acid synthase system from Mycobacterium tuberculosis.* Proc Natl Acad Sci U S A, 104(37):14628–33, 2007. Journal Article Research Support, Non-U.S. Gov't United States.

[138] Samal, A., S. Singh, V. Giri, S. Krishna, N. Raghuram and S. Jain: *Low degree metabolites explain essential reactions and enhance modularity in biological networks.* BMC Bioinformatics, 7:118, 2006.

[139] Schapire, RE.: *The strength of weak learnability.* Machine Learning, 5(2):197–227, 1990.

[140] Schapire, RE., Y. Freund, P. Bartlett and WS. Lee: *Boosting the margin: a new explanation for the effectiveness of voting methods*. The Annals of Statistics, 26(5):1651 – 1686, 1998.

[141] Schellenberger, J., J. O. Park, T. M. Conrad and B. O. Palsson: *BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions*. BMC Bioinformatics, 11:213. GM00806-06/GM/NIGMS NIH HHS/United States Journal Article Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov't England.

[142] Schramm, G., N. Kannabiran and R. Konig: *Regulation patterns in signaling networks of cancer*. BMC Syst Biol, 4:162, 2010. Journal Article Research Support, Non-U.S. Gov't England.

[143] Schramm, G., M. Zapatka, R. Eils and R. König: *Using gene expression data and network topology to detect substantial pathways, clusters and switches during oxygen deprivation of Escherichia coli*. BMC Bioinformatics, 8(1):149, 2007.

[144] Schuetz, R., L. Kuepfer and U. Sauer: *Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli*. Mol Syst Biol, 3:119, 2007.

[145] Schuster, M. and E. P. Greenberg: *Early activation of quorum sensing in Pseudomonas aeruginosa reveals the architecture of a complex regulon*. BMC Genomics, 8:287, 2007.

[146] Schuster, S., D. A. Fell and T. Dandekar: *A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks*. Nat Biotechnol, 18(3):326–32, 2000.

[147] Seringhaus, M., A. Paccanaro, A. Borneman, M. Snyder and M. Gerstein: *Predicting essential genes in fungal genomes*. Genome Res, 16(9):1126–35, 2006.

[148] Sgraja, T., M. S. Alphey, S. Ghilagaber, R. Marquez, M. N. Robertson, J. L. Hemmings, S. Lauw, F. Rohdich, A. Bacher, W. Eisenreich, V. Illarionova and W. N. Hunter: *Characterization of Aquifex aeolicus 4-diphosphocytidyl-2C-methyl-d-erythritol kinase - ligand recognition in a template for antimicrobial drug discovery*. Febs J, 275(11):2779–94, 2008. Biotechnology and Biological Sciences Research Council/United Kingdom Wellcome Trust/United Kingdom Journal Article Research Support, Non-U.S. Gov't England.

[149] Shi, W., M. Bogdanov, W. Dowhan and D. R. Zusman: *The pss and psd genes are required for motility and chemotaxis in Escherichia coli.* J Bacteriol, 175(23):7711–4, 1993.

[150] Singh, N., G. Cheve, M. A. Avery and C. R. McCurdy: *Targeting the methyl erythritol phosphate (MEP) pathway for novel antimalarial, antibacterial and herbicidal drug discovery: inhibition of 1-deoxy-D-xylulose-5-phosphate reductoisomerase (DXR) enzyme.* Curr Pharm Des, 13(11):1161–77, 2007.

[151] Sohn, M. J., C. J. Zheng and W. G. Kim: *Macrolactin S, a New Antibacterial Agent with FabG-inhibitory Activity from Bacillus sp. AT28.* J Antibiot (Tokyo), 61(11):687–91, 2008.

[152] Spoering, A. L., M. Vulic and K. Lewis: *GlpD and PlsB participate in persister cell formation in Escherichia coli.* J Bacteriol, 188(14):5136–44, 2006. Journal Article United States.

[153] Srivastava, A., P. Mukherjee, P. V. Desai, M. A. Avery and B. L. Tekwani: *Structural analysis of farnesyl pyrophosphate synthase from parasitic protozoa, a potential chemotherapeutic target.* Infect Disord Drug Targets, 8(1):16–30, 2008. U01/CI 000211-01/CI/NCPDCID CDC HHS/United States U50/CCU423310-01/PHS HHS/United States Comparative Study Journal Article Research Support, U.S. Gov't, P.H.S. Review United Arab Emirates.

[154] Stone, M.: *Cross-validatory choice and assessment of statistical predictions.* Journal of the Royal Statistical Society, 36(2):111147, 1974.

[155] Straus, D. S. and B. N. Ames: *Histidyl-transfer ribonucleic acid synthetase mutants requiring a high internal pool of histidine for growth.* J Bacteriol, 115(1):188–97, 1973. Journal Article United states.

[156] Takahashi, H., A. Ohki, M. Kanzaki, A. Tanaka, Y. Sato, B. Matthes, P. Boger and K. Wakabayashi: *Very-long-chain fatty acid biosynthesis is inhibited by cafenstrole, N,N-diethyl-3-mesitylsulfonyl-1H-1,2,4-triazole-1-carboxamide and its analogs.* Z Naturforsch C, 56(9-10):781–6, 2001. Journal Article Germany.

[157] Testa, C. A., R. M. Cornish and C. D. Poulter: *The sorbitol phosphotransferase system is responsible for transport of 2-C-methyl-D-erythritol into Salmonella enterica serovar typhimurium.* J Bacteriol, 186(2):473–80, 2004.

[158] Tomita, M., K. Hashimoto, K. Takahashi, T. S. Shimizu, Y. Matsuzaki, F. Miyoshi, K. Saito, S. Tanida, K. Yugi, J. C. Venter and 3rd Hutchison, C. A.: *E-CELL: software environment for whole-cell simulation.* Bioinformatics, 15(1):72–84, 1999. Journal Article Research Support, Non-U.S. Gov't England.

[159] Tong, L.: *Acetyl-coenzyme A carboxylase: crucial metabolic enzyme and attractive target for drug discovery.* Cell Mol Life Sci, 62(16):1784–803, 2005.

[160] Tong, L. and Jr. Harwood, H. J.: *Acetyl-coenzyme A carboxylases: versatile targets for drug discovery.* J Cell Biochem, 99(6):1476–88, 2006.

[161] Van Daele, I., H. Munier-Lehmann, M. Froeyen, J. Balzarini and S. Van Calenbergh: *Rational design of 5'-thiourea-substituted alpha-thymidine analogues as thymidine monophosphate kinase inhibitors capable of inhibiting mycobacterial growth.* J Med Chem, 50(22):5281–92, 2007. Journal Article Research Support, Non-U.S. Gov't United States.

[162] Vapnik, V.: *Statistical Learning Theory.* John Wiley and Sons, New York, 1998.

[163] Vauterin, M., V. Frankard and M. Jacobs: *Functional rescue of a bacterial dapA auxotroph with a plant cDNA library selects for mutant clones encoding a feedback-insensitive dihydrodipicolinate synthase.* Plant J, 21(3):239–48, 2000. Journal Article Research Support, Non-U.S. Gov't England for cell and molecular biology.

[164] Wagner, A. and D. A. Fell: *The small world inside large metabolic networks.* Proc Biol Sci, 268(1478):1803–10, 2001.

[165] Wishart, D. S., C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang and J. Woolsey: *DrugBank: a comprehensive resource for in silico drug discovery and exploration.* Nucleic Acids Res, 34(Database issue):D668–72, 2006.

[166] Wuchty, S. and P. F. Stadler: *Centers of complex networks.* J Theor Biol, 223(1):45–53, 2003. Journal Article England.

[167] Yeh, I., T. Hanekamp, S. Tsoka, P. D. Karp and R. B. Altman: *Computational analysis of Plasmodium falciparum metabolism: organizing genomic information to facilitate drug discovery.* Genome Res, 14(5):917–24, 2004.

[168] Zawadzke, L. E., M. Norcia, C. R. Desbonnet, H. Wang, K. Freeman-Cook and T. J. Dougherty: *Identification of an inhibitor of the MurC enzyme, which catalyzes an essential step in the peptidoglycan precursor synthesis pathway.* Assay Drug Dev Technol, 6(1):95–103, 2008.

[169] Zhu, X., M. Gerstein and M. Snyder: *Getting connected: analysis and principles of biological networks.* Genes Dev, 21(9):1010–24, 2007. Zhu, Xiaowei Gerstein, Mark Snyder, Michael Research Support, N.I.H., Extramural Review United States Genes & development Genes Dev. 2007 May 1;21(9):1010-24.

Appendix A

Additional results

A.1 Essential reactions found by our machine learning approach but not by FBA

Amino acid metabolism

Reaction BiGG ID	Reaction name	EC-number	ORF
R_ASNTRS	Asparaginyl-tRNAsynthetase	6.1.1.22	b0930
R_ASPO3	L-aspartate oxidase	1.4.3.16	b2574
R_ASPO4	L-aspartate oxidase	1.4.3.16	b2574
R_ASPO5	L-aspartate oxidase	1.4.3.16	b2574
R_ASPO6	L-aspartate oxidase	1.4.3.16	b2574
R_ASPTRS	Aspartyl-tRNA synthetase	6.1.1.12	b1866
R_ARGTRS	Arginyl-tRNAsynthetase	6.1.1.19	b1876
R_PROTRS	Prolyl-tRNAsynthetase	6.1.1.15	b0194
R_HISTRS	Histidyl-tRNAsynthetase	6.1.1.21	b2514
R_PSERT	phosphoserinetransaminase	2.6.1.52	b0907
R_SERTRS	Seryl-tRNA synthetase	6.1.1.11	b0893
R_SERTRS2	Seryl-tRNA synthetase (selenocystein)	6.1.1.11	b0893
R_THRTRS	Threonyl-tRNAsynthetase	6.1.1.3	b1719
R_CYSTRS	Cysteinyl-tRNAsynthetase	6.1.1.16	b0526
R_METTRS	Methionyl-tRNAsynthetas	6.1.1.10	b2114
R_PHETRS	Phenylalanyl-tRNAsynthetase	6.1.1.20	b1713, b1714
R_TYRTRS	Tyrosyl-tRNAsynthetase	6.1.1.1	b1637
R_LEUTRS	Leucyl-tRNAsynthetase	6.1.1.4	b0642
R_VALTRS	Valyl-tRNAsynthetase	6.1.1.9	b4258

Biosynthesis of steroids

Reaction BiGG ID	Reaction name	EC-number	ORF
R_DMPPS	1-hydroxy-2-methyl-2-(E)-butenyl 4-diphosphate reductase (dmpp)	1.17.1.2	b0029

Fatty acid biosynthesis

Reaction BiGG ID	Reaction name	EC-number	ORF
R_3OAR180	3-oxoacyl-[acyl-carrier-protein] reductase (n-C18:0)	1.1.1.100	b1093
R_3OAR181	3-oxoacyl-[acyl-carrier-protein] reductase (n-C18:1)	1.1.1.100	b1093
R_EAR100x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C10:0)	1.3.1.9	b1288
R_EAR100y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C10:0)	1.3.1.10	b1288
R_EAR120x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C12:0)	1.3.1.9	b1288
R_EAR120y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C12:0)	1.3.1.10	b1288
R_EAR121x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C12:1)	1.3.1.9	b1288
R_EAR121y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C12:1)	1.3.1.10	b1288
R_EAR140x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C14:0)	1.3.1.9	b1288
R_EAR140y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C14:0)	1.3.1.10	b1288
R_EAR141x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C14:1)	1.3.1.9	b1288
R_EAR141y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C14:1)	1.3.1.10	b1288
R_EAR160x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C16:0)	1.3.1.9	b1288
R_EAR160y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C16:0)	1.3.1.10	b1288
R_EAR161x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C16:1)	1.3.1.9	b1288
R_EAR161y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C16:1)	1.3.1.10	b1288
R_EAR180x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C18:0)	1.3.1.9	b1288
R_EAR180y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C18:0)	1.3.1.10	b1288
R_EAR181x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C18:1)	1.3.1.9	b1288
R_EAR181y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C18:1)	1.3.1.10	b1288
R_EAR40x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C4:0)	1.3.1.9	b1288
R_EAR40y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C4:0)	1.3.1.10	b1288
R_EAR60x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C6:0)	1.3.1.9	b1288
R_EAR60y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C6:0)	1.3.1.10	b1288
R_EAR80x	enoyl-[acyl-carrier-protein] reductase (NADH) (n-C8:0)	1.3.1.9	b1288
R_EAR80y	enoyl-[acyl-carrier-protein] reductase (NADPH) (n-C8:0)	1.3.1.10	b1288

Glycerophospholipid metabolism

Reaction BiGG ID	Reaction name	EC-number	ORF
R_DASYN120	CDP-diacylglycerol synthetase (n-C12:0)	2.7.7.41	b0175
R_DASYN140	CDP-diacylglycerol synthetase (n-C14:0)	2.7.7.41	b0175
R_DASYN141	CDP-diacylglycerol synthetase (n-C14:1)	2.7.7.41	b0175
R_DASYN180	CDP-diacylglycerol synthetase (n-C18:0)	2.7.7.41	b0175
R_DASYN181	CDP-diacylglycerol synthetase (n-C18:1)	2.7.7.41	b0175
R_PGSA120	Phosphatidylglycerol synthase (n-C12:0)	2.7.8.5	b1912
R_PGSA140	Phosphatidylglycerol synthase (n-C14:0)	2.7.8.5	b1912
R_PGSA141	Phosphatidylglycerol synthase (n-C14:1)	2.7.8.5	b1912
R_PGSA160	Phosphatidylglycerol synthase (n-C16:0)	2.7.8.5	b1912
R_PGSA161	Phosphatidylglycerol synthase (n-C16:1)	2.7.8.5	b1912
R_PGSA180	Phosphatidylglycerol synthase (n-C18:0)	2.7.8.5	b1912
R_PGSA181	Phosphatidylglycerol synthase (n-C18:1)	2.7.8.5	b1912
R_PSD141	Phosphatidylserine decarboxylase (n-C14:1)	4.1.1.65	b4160
R_PSD181	Phosphatidylserine decarboxylase (n-C18:1)	4.1.1.65	b4160
R_PSSA120	Phosphatidylserine syntase (n-C14:0)	2.7.8.8	b2585
R_PSSA140	Phosphatidylserine syntase (n-C14:1)	2.7.8.8	b2585
R_PSSA141	Phosphatidylserine syntase (n-C16:0)	2.7.8.8	b2585
R_PSSA180	Phosphatidylserine syntase (n-C18:0)	2.7.8.8	b2585
R_PSSA181	Phosphatidylserine syntase (n-C18:1)	2.7.8.8	b2585

Glycerolipid metabolism and Glycerophospholipid metabolism

Reaction BiGG ID	Reaction name	EC-number	ORF
R_AGPAT120	1-tetradecanoyl-sn-glycerol 3-phosphate O-acyltransferase (n-C12:0)	2.3.1.51	b3018
R_AGPAT140	1-tetradecanoyl-sn-glycerol 3-phosphate O-acyltransferase (n-C14:0)	2.3.1.51	b3018
R_AGPAT141	1-tetradec-7-enoyl-sn-glycerol 3-phosphate O-acyltransferase (n-C14:1)	2.3.1.51	b3018
R_AGPAT180	1-octadecanoyl-sn-glycerol 3-phosphate O-acyltransferase (n-C18:0)	2.3.1.51	b3018
R_AGPAT181	1-octadec-7-enoyl-sn-glycerol 3-phosphate O-acyltransferase (n-C18:1)	2.3.1.51	b3018
R_G3PAT120	glycerol-3-phosphate acyltransferase (C12:0)	2.3.1.15	b4041
R_G3PAT140	glycerol-3-phosphate acyltransferase (C14:0)	2.3.1.15	b4041
R_G3PAT141	glycerol-3-phosphate acyltransferase (C14:1)	2.3.1.15	b4041
R_G3PAT180	glycerol-3-phosphate acyltransferase (C18:0)	2.3.1.15	b4041
R_G3PAT181	glycerol-3-phosphate acyltransferase (C18:1)	2.3.1.15	b4041

A.2 Correlation between gene essentiality and all of the features

The feature values of each gene were correlated with the essentiality of the gene (1 = essential, 0 = non-essential). High values indicate that the feature was positively correlated to essentiality. These values were obtained for all gold standards (ecoB, ecoG for *E. coli* and paeJ, paeL for *P. aeruginosa*) and for the combined gold standards (all data).

Table A.1: Correlation coefficients ($R(f)$) of for the correlation between essentiality and all of the features

	ecoB		ecoG		paeL		paeJ	
	$R(f)$	p-value	$R(f)$	p-value	$R(f)$	p-value	$R(f)$	p-value
RUP	-0.145	7E-05	-0.121	0.002	-0.107	0.004	-0.113	0.002
PUP	0.169	4E-06	0.124	0.001	0.056	0.134	0.101	0.006
ND	0.013	0.726	-0.016	0.685	-0.060	0.113	-0.051	0.168
APL	-0.074	0.042	-0.092	0.016	-0.060	0.109	-0.048	0.193
LSP	-0.068	0.065	-0.090	0.018	-0.039	0.300	-0.016	0.669

	ecoB		ecoG		paeL		paeJ	
	$R(f)$	p-value	$R(f)$	p-value	$R(f)$	p-value	$R(f)$	p-value
NS	0.116	0.002	0.082	0.033	0.056	0.137	0.092	0.013
NP	0.054	0.142	0.009	0.811	0.047	0.208	0.070	0.057
NNR	-0.039	0.283	-0.046	0.235	-0.012	0.759	-0.015	0.676
NNNR	-0.050	0.169	-0.043	0.263	-0.027	0.469	-0.009	0.798
CCV	-0.032	0.379	-0.073	0.058	-0.044	0.238	-0.029	0.427
DIR	0.080	0.028	0.099	0.009	0.047	0.216	0.060	0.103
CP	0.109	0.003	0.036	0.350	0.046	0.227	0.086	0.021
LS	0.073	0.045	0.044	0.246	0.060	0.109	0.072	0.052
NDR	-0.108	0.003	-0.093	0.015	-0.080	0.034	-0.012	0.752
NDC	-0.109	0.003	-0.094	0.0142	-0.080	0.033	-0.012	0.738
NDRD	-0.107	0.004	-0.092	0.016	-0.079	0.035	-0.011	0.763
NDCD	-0.108	0.003	-0.093	0.015	-0.080	0.034	-0.012	0.754
NDCR	0.007	0.850	-0.041	0.287	-0.066	0.081	0.041	0.271
NDCC	0.005	0.882	-0.042	0.276	-0.066	0.078	0.040	0.282
NDCRD	0.008	0.821	-0.040	0.301	-0.065	0.085	0.042	0.263
NDCCD	0.007	0.850	-0.041	0.287	-0.066	0.082	0.041	0.269
BW	0.164	1E-05	0.152	7E-05	0.128	0.001	0.127	0.001
CN	0.040	0.274	0.053	0.167	0.047	0.214	0.085	0.021
EC	0.071	0.052	0.075	0.049	0.105	0.005	0.144	1E-04
EV	-0.040	0.272	-0.050	0.189	-0.057	0.127	-0.083	0.025
NAR	-0.100	0.007	-0.045	0.241	-0.093	0.013	-0.142	1E-04
H30	-0.060	0.102	-0.055	0.155	-0.059	0.115	-0.092	0.013
H20	-0.043	0.236	-0.045	0.244	-0.075	0.047	-0.095	0.011
H10	-0.057	0.119	-0.089	0.019	-0.094	0.012	-0.128	5E-04
H7	-0.072	0.048	-0.108	0.005	-0.111	0.003	-0.146	8E-05
H5	-0.099	0.007	-0.131	0.001	-0.124	0.001	-0.150	5E-05
H3	-0.073	0.045	-0.104	0.006	-0.158	2E-05	-0.169	4E-06
NGSE	-0.141	1E-04	-0.110	0.004	-0.037	0.329	-0.031	0.398
MCC	-0.041	0.265	-0.065	0.091	-0.040	0.291	-0.053	0.150
PR	0.126	0.001	0.044	0.250	0.043	0.258	0.049	0.190

	ecoB		ecoG		paeL		paeJ	
	$R(f)$	p-value	$R(f)$	p-value	$R(f)$	p-value	$R(f)$	p-value
Nc	-0.039	0.291	-0.163	2E-05	-0.108	0.004	-0.153	3E-05
T3s	0.059	0.104	0.131	6E-04	0.257	4E-12	0.201	4E-08
C3s	-0.009	0.816	-0.082	0.032	-0.128	6E-04	-0.072	0.053
A3s	-0.028	0.446	0.053	0.169	0.049	0.195	0.058	0.115
G3s	-0.068	0.064	-0.100	0.009	-0.009	0.816	-0.065	0.079
phe	-0.054	0.138	-0.065	0.090	-0.013	0.738	-0.059	0.113
ser	-0.086	0.018	-0.100	0.009	0.016	0.675	0.025	0.492
tyr	-0.082	0.025	-0.118	0.002	-0.030	0.431	-0.050	0.174
cys	-0.081	0.027	-0.089	0.020	-0.075	0.047	-0.100	0.007
leu	0.041	0.263	0.019	0.615	0.041	0.281	0.011	0.776
trp	-0.024	0.509	-0.096	0.012	-0.033	0.381	-0.058	0.115
pro	-0.005	0.883	-0.043	0.259	-0.016	0.675	-0.031	0.406
his	-0.017	0.643	0.005	0.899	-0.078	0.039	-0.063	0.087
arg	0.109	0.003	0.076	0.047	-0.027	0.469	-0.036	0.332
gln	-0.057	0.121	-0.036	0.352	-0.028	0.454	-0.026	0.483
ile	0.027	0.465	0.008	0.832	0.074	0.047	0.110	0.003
met	0.006	0.880	0.035	0.358	0.082	0.029	0.058	0.117
thr	0.023	0.536	-0.005	0.892	0.000	0.996	0.006	0.881
asn	-0.045	0.224	-0.094	0.014	-0.025	0.514	-0.031	0.402
lys	-0.001	0.975	0.045	0.245	0.090	0.017	0.107	0.004
val	0.057	0.122	0.118	0.002	0.099	0.009	0.122	0.001
ala	0.025	0.495	0.040	0.294	-0.045	0.229	-0.063	0.087
asp	0.033	0.374	0.071	0.064	-0.010	0.781	0.042	0.253
glu	0.001	0.977	0.044	0.248	-0.015	0.683	-0.008	0.823
gly	-0.014	0.692	-0.023	0.553	-0.088	0.018	-0.047	0.206

Table A.2: Results of AUC (area under curve of the receiver operator characteristics) for each feature

Feature	AUC				Feature	AUC			
	paeL	paeJ	ecoB	ecoG		paeL	paeJ	ecoB	ecoG
RUP	0.569	0.561	0.597	0.568	H3	0.660	0.625	0.537	0.547
PUP	0.555	0.579	0.637	0.587	H5	0.612	0.602	0.542	0.547
NS	0.555	0.568	0.587	0.541	H7	0.588	0.589	0.525	0.533
NP	0.546	0.558	0.557	0.512	H10	0.565	0.570	0.518	0.524
NNR	0.491	0.479	0.521	0.524	H20	0.539	0.538	0.510	0.508
NNNR	0.512	0.483	0.541	0.542	H30	0.527	0.533	0.513	0.510
CCV	0.523	0.513	0.522	0.549	phe	0.530	0.571	0.551	0.541
DIR	0.535	0.537	0.558	0.560	ser	0.517	0.522	0.557	0.558
ND	0.542	0.547	0.414	0.561	tyr	0.530	0.541	0.572	0.584
APL	0.541	0.546	0.587	0.564	cys	0.576	0.579	0.555	0.551
LSP	0.535	0.541	0.584	0.561	leu	0.534	0.506	0.534	0.516
NDR	0.580	0.521	0.556	0.558	trp	0.580	0.580	0.536	0.587
NDC	0.590	0.534	0.568	0.554	pro	0.508	0.527	0.523	0.523
NDRD	0.586	0.525	0.546	0.547	his	0.565	0.548	0.521	0.499
NDCD	0.587	0.525	0.569	0.560	arg	0.542	0.536	0.597	0.566
NDCR	0.541	0.532	0.542	0.509	gln	0.516	0.513	0.550	0.526
NDCC	0.543	0.529	0.538	0.508	ile	0.565	0.576	0.532	0.511
NDCRD	0.539	0.534	0.546	0.505	met	0.567	0.542	0.496	0.511
NDCCD	0.539	0.535	0.538	0.509	thr	0.495	0.505	0.513	0.521
CP	0.534	0.553	0.578	0.522	asn	0.518	0.525	0.53	0.570
LS	0.545	0.562	0.566	0.519	lys	0.581	0.576	0.497	0.531
bw	0.644	0.616	0.656	0.629	val	0.587	0.595	0.552	0.588
cn	0.516	0.540	0.500	0.513	ala	0.542	0.554	0.514	0.523
ec	0.561	0.584	0.511	0.509	asp	0.491	0.548	0.518	0.529
ev	0.604	0.571	0.589	0.559	glu	0.503	0.511	0.506	0.541
NAR	0.528	0.572	0.550	0.524	gly	0.586	0.546	0.521	0.516
T3s	0.683	0.638	0.579	0.613	codons	0.622	0.629	0.541	0.645
C3s	0.599	0.545	0.516	0.560	COX	0.541	0.536	0.616	0.571
A3s	0.544	0.554	0.516	0.538	MCO	0.528	0.535	0.520	0.541
G3s	0.510	0.554	0.561	0.575	ORT	0.530	0.531	0.588	0.497