RNA Interference Data:

from a Statistical Analysis
to Network Inference

INAUGURAL-DISSERTATION

zur
Erlangung der Doktorwürde
der Naturwissenschafltich-Mathematischen Gesamtfafakultät
der Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Diplom-Informatikerin Bettina Knapp
aus Freudenstadt/ Baden-Württemberg

Erstgutachter: Prof. Dr. Lars Kaderali
Zweitgutachter: Prof. Dr. Roland Eils

Tag der mündlichen Püfung: 23.04.2012

Abstract

Viruses are the cause of many severe human diseases such as Hepatitis C, Dengue fever, AIDS, Influenza and even cancer. In consequence of viral diseases several millions of people die every year all over the world. Due to the rapid evolution of viruses their drug development and treatment are especially difficult. The present work aims at getting a better understanding of the ongoing signaling processes of certain diseases. To do this, methods for the analysis and network inference of RNA interference (RNAi) data are presented.

Recent biological and technological advances in the field of RNAi enable the knockdown of individual genes in a high-content high-throughput manner. Thereby, a detailed quantification of perturbation effects on specific phenotypes can be assessed using multiparametric imaging. This in turn allows the identification of genes which are involved in certain biological processes such as virus-host factors used in the viral life-cycles. However, hit lists of already published RNAi screens show only a small overlap, even for studies of the same virus. This may be due to insufficient data analysis where the potential of microscopic screening data is not fully tapped since individual cell measurements are not taken into account for data normalization and hit scoring. This thesis shows that for RNAi data studying Hepatitis C and Dengue virus the phenotypic effect after a perturbation is highly influenced by each cell's population context. Therefore, novel methodologies are proposed which use the individual cell measurements for the data analysis and statistical scoring. This results in an increased sensitivity and specificity in comparison to already existing methods where these factors are disregarded. The method proposed here allows the identification of already existing as well as new hit genes which are significantly involved in the respective viral life-cycles.

The spatial and temporal placement of these hits, however, still remains unknown, and the ongoing signaling processes are only poorly understood. To understand the underlying biology from a system wide view it is necessary to infer the signaling cascade of involved factors in detail. One of the challenges of network inference is the exponentially increasing dimensionality with an increasing number of nodes. The method proposed in this thesis is formulated as a linear optimization problem which can be solved efficiently even for large data sets. The model can incorporate data of single or multiple perturbations at the same time. The aim is to find the network topology which best represents the given data. Based on simulated data for an small artificial five-node example the robustness of the model against noisy or incomplete data is demonstrated. Furthermore, for this small as well as for larger networks with 10 to 52 nodes it is shown that the model achieves superior results than random guessing. In addition, the performance and the computation time of large networks are better than another approach which has been recently published. Moreover, the network inference method presented here has been applied to data measuring the signaling of ErbB proteins. These proteins are associated with the development of many human cancers. The results of the network inference show that already known signaling cascades can be successfully reconstructed from the data. Additionally, newly learned protein-protein interactions indicate that there are several still unknown feedback and feedforward loops. The proteins of these loops may serve as potential targets to control ErbB signaling. The knowledge about these factors is an important step towards the development of new drugs and therefore,this helps to fight ErbB related diseases.

Zusammenfassung

Viren sind die Ursache von vielen schweren Krankheiten, wie zum Beispiel Hepatitis C, Dengue Fieber, AIDS, Influenza und auch Krebs. Mehrere Millionen Menschen sterben durch die Folgen von viralen Krankheiten jedes Jahr auf der ganzen Welt. Aufgrund der schnelle Weiterentwicklung von Viren ist deren Behandlung und die Entwicklung von Medikamenten besonders schwierig. Durch die vorliegende Arbeit sollen ablaufende Prozesse bei bestimmten Krankheiten besser verstanden werden. Dafür werden Methoden zur Analyse und Netzwerkinferenz von RNA Interferenz (RNAi) Daten vorgestellt.

Neueste biologische und technologische Fortschritte auf dem Gebiet der RNA Interferenz ermöglichen das Herrunterregulieren von einzelnen Genen in einem hochaufgelösten Hochdurchsatzverfahren. Dadurch kann mit Hilfe von multiparametrischen Bildgebungsverfahren eine detailierte Quantifizierung von Perturbationseffekte auf bestimmte Phänotypen durchgeführt werden. Dies erlaubt wiederum die Identifizierung von Genen, die in bestimmte biologische Prozesse involviert sind, wie zum Beispiel Virus-Wirts-Faktoren, die im viralen Lebenszyklus genutzt werden. Hitlisten von bereits publizierten RNAi Studien zeigen jedoch nur eine geringe Übereinstimmung, sogar für Studien die den gleichen Virus untersuchen. Der Grund hierfür kann eine unzureichende Datenanalyse sein, bei der das Potential von Mikroskopie-Daten nicht voll ausgeschöpft wird, da Einzelzellmessungen bei der Normalisierung und beim Hitscoring nicht berücksichtigt werden. Diese Arbeit zeigt, dass für RNAi Daten, die sich mit dem Hepatitis C und Dengue Virus befassen, der phänotypische Effekt nach einer Perturbation stark von dem Populationskontext jeder einzelnen Zelle beeinflusst wird. Deshalb werden neue Methoden vorgestellt, die die Messungen auf einzelnen Zellen für die Analyse und statistische Auswertung berücksichtigen. Dadurch wird eine erhöhte Sensitivität und Spezifizität im Vergleich zu bereits veröffentlichten Methoden, welche diese Faktoren unbeachtet lassen, erreicht. Die hier präsentierte Methode erlaubt die Identifizierung von bereits existierenden sowie neuen Hit-Genen, welche in den jeweiligen viralen Lebenszyklen signifikant involviert sind.

Die räumliche und zeitliche Anordnung dieser Hits bleibt dabei jedoch ungeklärt und die laufenden Signalprozesse sind bislang nur wenig verstanden. Um die zu Grunde liegende Biologie systemübergreifend zu erfassen, ist es notwendig, die Signalkaskaden von involvierten Faktoren im Detail zu rekonstruieren. Eine der Herausforderungen beim Lernen von Netzwerken ist die exponentiell anwachsende Dimensionalität für eine steigende Anzahl an Knoten. Die Methode, die in dieser Arbeit vorgestellt wird, ist als lineares Optimierungsproblem formuliert, das sogar für große Datensätze effizient lösbar ist. Das Modell kann Daten mit einzelnen oder mehreren Perturbationen gleichzeitig berücksichtigen. Ziel ist es, eine Netzwerktopologie zu finden, welche die Daten am Besten repräsentiert. Mit Hilfe von simulierten Daten für ein kleines künstliches Fünf-Knoten Beispiel wird die Robustheit des Modells gengenüber verrauschten und unvollständigen Daten aufgezeigt. Desweiteren wird für dieses kleine, sowie für größere Netzwerke mit 10 bis 52 Knoten gezeigt, dass das Modell bessere Ergebnisse als Raten liefert. Darüber hinaus sind die Resultate und die Rechenzeit bei großen Netzen besser als bei einem anderen Verfahren, das kürzlich publiziert wurde. Überdies wurde die hier vorgestellte Netzwerkinferenzmethode auf Daten, die die Signal-Prozessierung von ErbB-Proteinen untersuchen, angewandt. Diese Proteine werden mit der Entstehung von vielen humanen Krebsarten assoziiert. Die Ergebnisse der Netzwerkinferenz zeigen, dass bereits bekannte Signal-Kaskaden erfolgreich aus den Daten rekonstruiert werden können. Zusätzlich deuten neu gelernte Protein-Protein Interaktionen darauf hin, dass es noch einige bisher unbekannte "Feedforward"- und "Feedback"- Schleifen gibt. Die gelernten Faktoren in diesen Schleifen können als Ziele dienen um die ErbB-Signalgebung zu kontrollieren. Das Wissen über diese Proteine ist ein wichtiger Schritt zur Entwicklung von Medikamenten und dies trägt somit zur Bekämpfung von Krankheiten, die mit ErbB in Zusammenhang stehen, bei.

Acknowledgements

First of all, I want to thank my supervisor Prof. Dr. Lars Kaderali who offered me the possibility to do this work. I am thankful for the great support from Lars and that I was always welcomed to consult him and to discuss my work with him. I am fortunate for his good advices and help and for the various stimulating meetings and discussions we had.

In addition, I acknowledge the supportive help, advices and critical comments from all my colleagues and all the people in the Viroquant Research Group Modeling which shared working hours with me during my PhD time. We had many interesting group seminars where I learned many different things irrespective from my own topic. Special thanks go to Johanna Mazur, who was willing to answer all my mathematical questions, to listen to the difficulties I had and with whom I could discuss all kinds of problems. Furthermore, many thanks go to Prabhav Kalaghatgi, Nora Rieber and Joanne Barry, who were all very supportive in finding alternative formulations and synonyms while I wrote my thesis.

Moreover, I am grateful that Lars introduced me to various co-operation partners and different joint projects which enabled me to enlarge my expertise and to develop the methods presented here. The co-operations with the groups of Prof. Dr. Ralf Bartenschlager, Prof. Dr. Hans-Georg Kräusslich and Dr. Vytaute Starkuviene facilitated the evaluation of my projects on real biological data. Among the people in these groups, my thanks go particularly to Dr. Christoph Claas, Andrius Serva, Dr. Ilka Rebhan, Dr. Anil Kumar and Kathleen Börner. We had many fruitful discussions which helped me a lot to see things from different perspectives. Additionally, I want to thank Lars, Christoph and Dr. Narsis Kiani for proofreading parts of this thesis and for their feedback.

Apart from the people directly or partly involved in Viroquant, I would like to thank Katharina Buck for being a very good friend throughout my studies and my PhD time. She was really very helpful, supportive and motivating and her advices and suggestions made it easier for me to overcome all obstacles during this thesis and my life.

I am deeply grateful to my parents and my whole family. They were always supporting and encouraging me and this helped me more than anything else.

Finally, I would like to thank Sebastian Bieniek for his support especially during the last period of this thesis. I am thankful for all the time we spend together as well as for all his kindness and patience.

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur mit erlaubten Hilfsmitteln angefertigt habe.

Heidelberg, den

Vorname Nachname des Bearbeiters

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the recent years, technical developments enabled a facilitated measurement of biological high-throughput data. This results in both, a qualitative and quantitative improvement of the generated data and offers the potential to get a better understanding of complex biological systems. RNA interference (RNAi), for example, is used to study the role of various genes in different biological processes like virus-host interactions during virus entry, replication and release.
The general aim of this type of experiments is to gain new insights in the spatial and temporal placement of individual genes in the cell. While RNA interference is an easy and fast way to screen genes in a high-content, high-throughput manner, the inference of signal transduction networks from this data is a challenging task [MS06].

In this thesis, we are particularly interested in the inference of gene signaling networks from high-throughput RNAi screening data. Since in biology, experimental measurements are never completely exact, we want to diminish noise in the data to extract the real biological signal optimally. Whereas there exist various methodologies for the analysis of Microarray data, there are only some basic techniques for RNAi data [RKEK09a]. The recent advances in the technical field of image recognition of RNAi screens offer the possibility to perform measurements which are very detailed. Microscopy-based read-outs for example allow the quantification of several phenotypic effects of each individual cell. However, the state of the art method up to now is to use only a summarization of the numerous cell measurements and many of the phenotypic effects such as morphological changes or each cell's population context are in most cases ignored [RRB$^+$11, BHS$^+$10, WRBB08, BBH06]. Since the overlap of hit genes identified by already published RNAi screens is very low even for the same viruses [Gof08], we will show that this effect is at least partially due to the fact that individual cell measurements are not taken into account for the analysis up to now. This is supported by a recent publication from Snijder and colleagues, who showed that the cell context is largely influencing viral infection [SSR$^+$09]. In this thesis, we develop an analysis method which decreases the false positive and false negative rate of controls in two given data sets to a minimum. The true biological signal can be extracted and this allows the identification of hit genes significantly involved in the phenotypes of interest.

After the identification of genes which are involved in certain processes, the next step is to study how they interact with each other to understand the underlying biology in a much broader context. Getting a better insight of the ongoing processes, may enable the identification, for example, of key factors involved in individual diseases. Thereby, the development of new drug targets which influence different disease pathways such as the life cycle of viral infections can be enhanced.
Although, RNAi screening data have a high potential to give new insights in signaling pathways, the network inference is a challenging task. The number of possible network topologies, for instance, is increasing exponentially with the number of genes and therefore, a complete enumeration of the solution space is computationally not tractable for high dimension problems [KDZ$^+$09, MBS05]. However, in biology signaling process are most often mediated via dozens or hundreds of genes. We present a method which is able to solve even large-scale problems efficiently and thus, this enhances the use of network inference in real biological problems significantly.

Organization of the Thesis

This thesis is structured into two parts. In the first part, the whole process of RNA interference and the analysis of RNAi screening data is discussed in detail. We start with explaining the biological background of RNAi data and the experimental setup. Furthermore, we introduce RNAi data analysis and hit identification methods by giving an overview of already existing approaches. Thereafter, we describe new normalization and statistical scoring approaches which have been developed in this thesis. Based on the measurements of each individual cell, we show that the phenotypic effect after single perturbations is highly influenced by each cell's context. Our approach takes, beside some technical parameters, the cell context for normalization and for the calculation of significance scores. Thereby, we achieve much higher sensitivity and specificity values in comparison to existing methods on RNAi screening data of two data sets studying Hepatitis C and Dengue virus infection.

The second part describes the challenge of inferring networks from perturbation data. First, we discuss already existing methods, and then give the mathematical background of linear programming which is a basis of the model presented here. The proposed model assumes that the signal transduction within a network is given as an information flow. The silencing effect of a gene on the measured output is therefore propagated down the path in the underlying graph. Preceding genes influence the effect of genes which are further down in the network. We present a linear program to find the network topology and the corresponding parameters which are best reflecting the data.

By formulating the inference task as a linear optimization problem it can be solved efficiently using the simplex algorithm [Sch99]. This allows to reconstruct networks of 10 nodes using simulated data in less than ten minutes which is significantly outperforming an existing method called DEPN published by Froehlich *et al.* [FSA$^+$09] which takes almost an hour. In addition, our results show a much higher performance in terms of sensitivity, specificity and precision than the DEPN approach. Furthermore, we show based on simulated data of a small five-node network that our approach is able to deal with noisy and missing data as well as with single and double perturbations at the same time.
Finally, we use a real data set studying ErbB signaling to reconstruct the interactions of the involved genes. We learned already known interactions as well as new ones which indicate for example, that there are feedback loops which have not been discovered yet.

Structure of the Thesis

The different Chapters within this thesis are organized as follows: This Chapter gives a general introduction of the topic. The Chapters 2 and 3 belong to the first part of the thesis. Chapter 2 introduces the biological background of RNA interference and viruses. Then, RNAi experiments and data generation are reported. Furthermore, existing strategies for the normalization and statistical analysis of RNAi data are explained.
In Chapter 3 we describe the newly developed strategy of processing RNAi screening data based on individual cell measurements. First, we describe the data on Hepatitis C and Dengue virus which is used to evaluate our methods and then, we discuss two already existing analysis approaches. Thereafter, we explain our approach and present results of the two data sets. Finally, we compare the results and discuss them in detail.

The next four Chapters (Chapter 4 to 7) form the second part of this thesis. In Chapter 4 an introduction on graphs and networks in biology is given. Then, the problem of inferring networks from perturbations data is

discussed and previously published methods are presented.
Chapter 5 explains the theory of optimization problems to give a background knowledge of linear problems and solvers. This is important for the model developed in this thesis and which is explained in Chapter 6. Here, we formulate the network inference task as a linear problem which can be solved efficiently using the simplex algorithm. At the end of Chapter 6, we describe a strategy to assess the performance of network inference methods.
Finally, in Chapter 7 we present results on simulated data as well as on real data using our model for the network inference. We use simulated data of a small five-node toy example and larger problems with networks of 10 to 52 nodes to assess and compare the performance of our approach with a recently published method and random guessing. Furthermore, using real data, we infer a network of 16 nodes involved in ErbB signaling.

Finally, Chapter 8 summarizes and concludes the whole thesis and gives an outlook.

Part I

RNAi Data

Chapter 2

Biological Background

This Chapter gives a background of the underlying biology and technology. It is separated into three different Sections. First, the concept of knockdowns using short interfering RNAs (siRNAs) is explained and viruses are introduced in general. The two virus types Dengue and Hepatitis C, which both belong to the family *Flaviviridae*, we discuss in more detail.

The second Section describes the methodology of RNA interference experiments using RNAi-based cell arrays and microwell plates. Furthermore the process of image-acquisition and -recognition is introduced. Finally, in the third Section, we explain methods commonly used for the RNAi data analysis, including approaches for the quantification of the data quality, normalization strategies and methods for the hit calling.

2.1 Biological Background

Systems biology aims at understanding biological processes when looking at the system as a whole. The goal is to describe complex interactions and to gain a systemic view of biological systems. During the last years, various technologies have been developed and enhanced to model cell function with the help of global cell measurements [CHI10]. For example RNA interference offers the determination of genes which are important for specific phenotypes by looking at the cell response after silencing them [PCL$^+$09]. A multitude of different possible applications exist, but especially for treating several diseases, such as virus infections, the identification of target genes offers new possibilities for drug development [AHP$^+$10] and thus, this is of great importance.

2.1.1 RNA Interference

RNAi is a process mediated by double-stranded RNA (dsRNA) molecules which enable sequence-specific, post-transcriptional gene silencing in a high-

throughput fashion. RNAi was first discovered in *Caenorhabditis elegans* (*C. elegans*) when dsRNA was injected into worm and the silencing of specific genes could be observed [FXM+98].
Figure 2.1 shows the basic steps of RNAi: After the introduction of dsRNA into the cytoplasm of the cell, the enzyme dicer (an endoribonuclease of the RNase III family) cuts it into siRNAs of about 21 to 28 nt in length [MT04]. These duplexes consist of a guide and a passenger strand, represented in red and yellow, respectively, in Figure 2.1. While the passenger strand is ejected and degraded, the guide strand is incorporated into the RNA-induced silencing complex (RISC) where it allows, since it is unwound, binding to the target mRNA of perfectly complementary sequence [CS09, MS02]. This leads to the RISC cleavage of the mRNA and thus, its translation is avoided and the production of the protein is blocked.

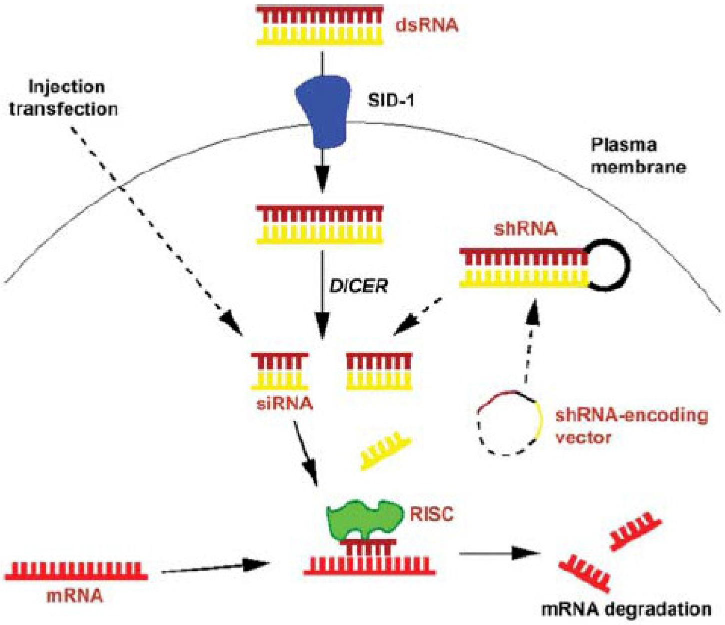


Figure 2.1: The RNAi mechanism [BEWS04]. First dsRNA is transfered to the cell which then is cutted by a DICER into duplexes. These are directed to the RISC complex which leads to the degradation of the homologous mRNA. (For more details see text.)

Using long dsRNA is historically the most common method for gene silencing in worms, flies and plants. In the case of *C. elegans* dsRNA is possibly transported into the cell by SID-1 (Systemic RNA Interference Defective) which is an RNA transporter [FH03]. Other possibilities are the direct microinjection into the animal or soaking the animals in a medium

which contains the dsRNA [BEWS04].
In mammals an interferon anti-viral response is induced when using dsRNAs longer than 30 nt. This results in a non-specific degradation of mRNA sequences. Elbashir and co-authors found in 2001 [EHL+01] that this can be avoided using chemically synthesized siRNA duplexes of 20 to 23 nt length, which are in their structure similar to dicer-processed dsRNA.
The siRNAs are directly transfered into the cytoplasm either by normal transfection or through vectors which are expressing the siRNAs. Thereby, the successful targeting of specific mRNAs in mammalian cells is enabled and the non-specific inhibition of protein synthesis is circumvented. The used vectors are DNA strands encoding short hairpin RNAs (shRNAs) which are cleaved by dicer. The vectors are often used for transfection since siRNA-mediated RNAi is of transient nature and they allow the control of the shRNAs by a promoter. This is particularly useful if genes involved in cell growth, differentiation or apoptosis pathways are targeted. See [KM06] for more details.

The RNAi technology offers many advantages in contrast to gene knockdowns: it is post-transcriptional and therefore reduces the risk of compensatory gene regulation. Furthermore, it is fast and less expensive and because of its temporary effect, the experimenter can control the time frame of the knockdown to an amount of 24 to 72 hours after siRNA delivery.
Thus, the discovery of RNAi offers a great potential, not only for the systematic elucidation of gene function and gene involvement in biological processes, but also for drug target identification in different diseases [BEWS04, DS09, WM09]. The RNAi technique is nowadays routinely used in biological experiments and allows high-throughput screens, for example focusing on genes involved in mitosis [NWH+10], immune response [MKG+05] or viral infection [BDB+08, LBN+09]. A genome-wide targeting is enabled and thereby, all candidate genes of a specific phenotype can be identified.
Nevertheless, there are some disadvantages. For example off-target effects which are due to sequence homology within a protein family or common domain. The choice of the target mRNA sequence is an important step in the design of an RNAi experiment. Choosing three to four regions within the target mRNA is recommended, and several siRNAs should be used to target the same gene [MS06, TB10, KCT+07]. In addition, regions which have at least 11 nt in common should be avoided, since they can affect off-target mRNAs [JBS+03].
The undesired effects can be uncovered by performing validation experiments such as gel-based real-time PCR or RNAi experiments with a high-content low-throughput screening.

2.1.2 Viruses

A virus is a subcellular infectious agent which can survive only if it interacts with factors of the replication mechanisms of a living cells organism. They do not have an own metabolism or a cellular structure and are not able to reproduce outside of a host cell.
The cell morphological structure of different viruses show high diversity. The virions (virus particles) are very small, between 10 and 300 nm, and have a very simple structure, consisting of only two parts: the genetic material and a protein coat (capsid). The capsid consists of identical protein subunits (capsomers) which are encoded by the viral genome and serves as a physical protection for the genetic material. Some viruses are additionally enveloped by lipids which are surrounding the capsid when it is outside of a cell. No matter whether they are enveloped or not, standard light microscopes cannot visualize the virions, so transmission electron microscopes are used.

Different viral species show a broad spectrum of different genomic structures. The genomic material is either RNA, DNA or both (retrovirus). Whether a retroviral genome is DNA or RNA depends on the different stages in the viral life cycle.
The viral genome is shaped circular, linear or segmented (genome is divided into separate parts), irrespective of the type of the nucleic acid. Single- (unpaired nucleic acid) or double-strands (two complementary paired nucleic acids) are possible. For the formation of the strands it does not matter whether it is RNA or DNA and some viruses even have a combination of single- and double-stranded genomes.

Although there are millions of virus types which differ broadly in their viral life cycle, five basic stages can be summarized for all of them:

1. Attachment:
   The viral capsid interacts with specific receptors located on the cell surface of the host cell.

2. Penetration:
   Depending on the virus type, the virions can enter the cell either using receptor-mediated endocytosis, or the viral and cellular membrane fuse and thereby the viral entry is enabled.

3. Uncoating:
   The viral capsid is removed and viral genomic material is released into the cytoplasm of the cell.

4. Replication:

The viral genome is replicated using both host and viral proteins which have been synthesized in advance using cellular processes.

5. Release:
   The new viral genome is coated by newly synthesized capsomers which self-assemble into capsids. Then the virus is released from the cell and a new replication cycle can start and new cells can be infected.

All organisms - animals, plants and bacteria - can be infected by viruses and numerous diseases (e.g Influenza, AIDS, Dengue fever, Hepatitis) are caused by viral infection. In the following, a more detailed description of Dengue virus and Hepatitis C virus is given.

Dengue Virus

The Dengue virus causes the Dengue fever, which is also known as breakbone fever. It is transmitted by the mosquito *Aedes aegypti* and affects around 50 million people per year in tropical and sub-tropical areas [Wor09]. Typical flu-like symptoms include fever, headache, patechial rashes and muscle pain. A severe course of the disease can lead to Dengue hemorrhagic fever or Dengue shock syndrome which are both life-threatening. There exists no vaccine against it up to now. The the transmission can be limited by reducing the number of mosquitoes.

The Dengue fever virus (DENV) belongs to the family *Flaviviridae* and there exist four serotypes, DENV-1 to DENV-4. The viral genome is composed of a single, positive strand RNA molecule with about 11 kb length. It is packaged by dimeric virus capsid proteins in a host-derived lipid-bilayer and enveloped by glycoproteins. After receptor-mediated endocytosis, the viral RNA is released into the cytoplasm and uncoated [MKR05].
Then, the genome is translated into a single poly-protein using viral and host factors and replication takes place in membrane-bound vesicles on the cell's endoplasmic reticulum (ER). Once the newly synthesized RNA and the structural proteins bud together in the lumen of the ER, virus assembly takes place. The resulting particles are still non-infectious and have to be transported through the *trans*-Golgi network (TGN) where they are cleaved by the host protease furin. Only the mature and infectious virions are released by exocytosis [UIPT$^+$10, MKR05].

Hepatitis C Virus

Hepatitis C is one of five known Hepatitis viruses (labeled from A to E). Symptoms of the acute phase (first six months after infection) are similar to flu symptoms. Up to 70% of infected people do not have symptoms at all. In the chronic phase (Hepatitis C virus infection exists for more than six months), Hepatitis C is affecting the liver. Although it is mostly still asymptomatic, a once established chronic infection can cause severe liver failures. Fibrosis, cirrhosis or liver cancer are possible consequences. It is transmitted by blood-to-blood contact and according to the World Health Organization [Wor11] about three percent of the world's population are infected and five to ten million people in Europe. Currently, no vaccine is known but a persistent infection can be medicated in some patients.

Like the Dengue virus, the Hepatitis C virus (HCV) belongs to the family *Flaviviridae* and it is an enveloped, positive sense, single-stranded RNA (9.6 kb long) virus. The life cycle of HCV looks as follows: the HCV particles attach to the cell surface and this triggers the release into the cell cytoplasm (Figure 2.2). Among all possible HCV receptors such as CD81, scavenger receptor B type I (SR-BI), low-density lipoprotein receptor (LDL-R) and asialoglycoprotein receptor (ASGP-R) [Tan06], CD81 is studied most. It has been shown that CD81 mediates binding of HCV via its envelope glycoprotein [PUC$^+$98]. After uncoating of the the viral genome from its capsid it is translated at the rough ER [BFP04]. Thereafter, genome assembly and HCV protein expression takes place and the progeny virions are assembled. Little is known about these steps in detail [Tan06] and the release of the newly produced virus particles from the host cell is not yet completely understood.

Whereas available broad-spectrum antibiotics are effectively used against bacterial pathogens, broad-spectrum antiviral therapeutics are hard to develop. Although viruses harbour a small genome they can reprogram host cells to promote their own replication. Due to the rapid evolution of viruses it is particularly difficult to treat infections. A starting point is to identify essential viral proteins, which can serve as drug targets. However, due to the high mutation rate of viruses this is a challenging task. Recent advances in RNAi and high-content screening microscopy facilitate the identification of new host factors [Che09]. This might provide further insights into the viral life cycle and aid in the development of new drugs.

2.2 RNAi Experiments

To find genes which are important for the viral life cycle, the technology of RNAi experiments is used. The viral response can be easily quantified after gene knockdowns.

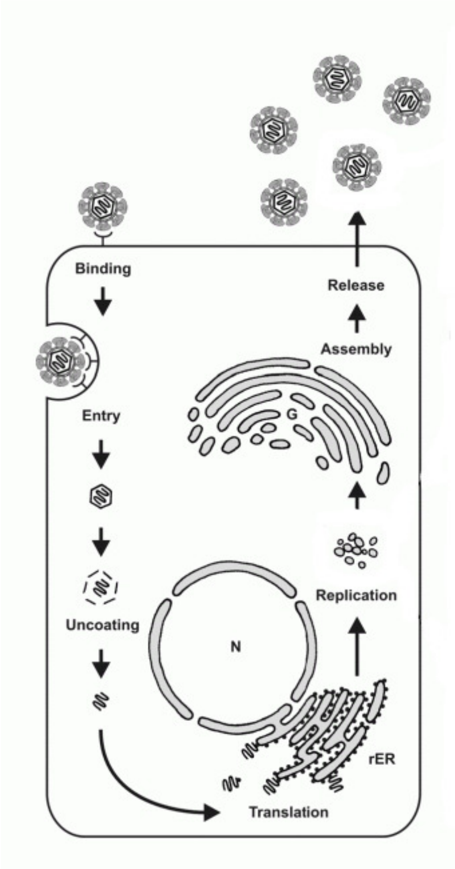


Figure 2.2: The Hepatitis C virus life cycle (modified from [BFP04]). The viral particle attaches to the cell surface and enters the cell. After uncoating the viral genome is translated at the rough ER (rER). After replication and assembly of the progeny virions, the newly produced virus is released from the cell. For details see text.

Basically there are two different possibilities for high-throughput RNAi experiments: RNAi-based cell microarrays on LabTek chambered coverglass slides and microwell plate based arrayed screening [MS06]. Both methods are explained in detail in the following two Sections.

2.2.1 RNAi-based Cell Microarrays

The cell microarray is a glass slide (in most cases a LabTek chambered coverglass slide) which allows transfection with hundreds to thousands of RNAi reagents at the same time. A typical experiment on RNAi-based cell arrays (see Figure 2.3) starts with the preparation of the transfection solutions containing the siRNA samples (synthesized siRNAs, enzymatically derived siRNAs (esiRNAs), plasmid-based shRNAs or virus-based shRNAs) which are derived from a certain siRNA library. Next, the samples are spotted on the LabTeks using a microarraying robot.

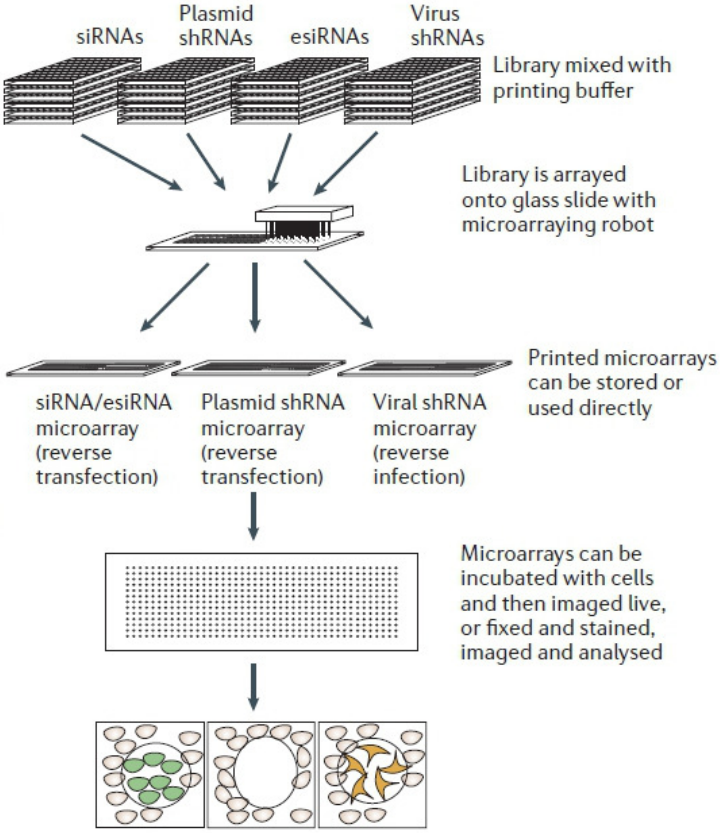

siRNAs
Plasmid shRNAs
esiRNAs
Virus shRNAs
Library mixed with printing buffer
Library is arrayed onto glass slide with microarraying robot
Printed microarrays can be stored or used directly
siRNA/esiRNA microarray (reverse transfection)
Plasmid shRNA microarray (reverse transfection)
Viral shRNA microarray (reverse infection)
Microarrays can be incubated with cells and then imaged live, or fixed and stained, imaged and analysed

Figure 2.3: Workflow for RNAi-based cell microarrays [MS06]. See text for details.

After drying and possibly storage of the slides, mammalian cells are seeded on the LabTeks. Finally, fluorescence microscopes or plate readers are used for generating the read-out via life cell imaging or imaging using immunostaining of transfected cell arrays [ENL$^+$07]. Life cell imaging can be done after a certain incubation time (mostly around 20 hours) without further handling. The other approach uses fixed and immunostained cells after an incubation with siRNA for about 48-60 hours depending on the used protocol.

In each spot positioned on the LabTek one siRNA is transfected into the overlying cells, which results in the knockdown of the respective gene. This so-called “reverse transfection” often leads to superior efficiency when compared to conventional formed transfection experiments [CLH$^+$08].

Using cell microarrays allows a high sample density and long-term storage after the spotting process. This reduces costs and correlations between different assays are increased since they can be performed on the same production day. Nevertheless, there are some disadvantages. The most important one is the risk of cross-contamination. The individual reagents are physically not separated and still the total number of spots per array should be maximized for a high-speed and a parallel data acquisition. However, Reymann and colleagues presented in 2009 [RBB$^+$09] a 9216-microwell cell array which does not show cross-contamination. They introduce physically separated cavities on cell arrays using a titanium coating. Three reference markers at the borders of the glass slide ensure the precise spot location. Accurate spotting is of great importance, because it allows the correct detection of the spot matrix during the scanning process.
A further disadvantage is given by the small number (in comparison to microwell plate based arrays) of cells per spot. This reduces the statistical relevance of a single experiment. In addition, no reverse transcriptase-polymerase chain reaction (RT-PCR) can be performed for validation after the data analysis and hit selection. This is due to the fact that individual spots are not physically separated and thus, the individual knockdown experiments cannot be processed further. All of these problems are solved in the microwell plate based screening process which is described in Section 2.2.2.

For measuring a phenotypic effect under the influence of a certain virus, cell seeding has to be followed by infection. After an additional incubation time, fixation and imaging is possible like in a non-virus based setting. To quantify the viral signal intensity during the image recognition process, the viral genome is tagged by a fluorescence protein, for instance the Green Fluorescence Protein (GFP).

2.2.2 Microwell Plate based Screening

Microwell plates exist of different sizes (12, 96 or 384 well-plates) of physically separated wells. In each well different reagents (siRNAs, plasmid shRNAs or viruses) can be applied.
During the experimental setup, a library of gene-targeting reagents has to be chosen first. Second, the reagents are arranged on one or several multi-well plates. Then, infection (if viral shRNAs are used), transfection or reverse transfection (if cells are added after the reagents) takes place. This results in a gene-specific silencing of the corresponding targets. For plates with more than 96 wells reverse transfection [KPH$^+$07] is recommended since it is easier to handle and it introduces less technical variability [TB10].
In well-plate based screens it is possible to pool reagents by grouping multiple siRNAs targeting the same gene together in one well [LVHWN10, BH10]. This decreases the number of necessary wells and thereby, this saves costs and time. However, for these screens it is technically challenging to get uniform pools of transfected cells. Furthermore, the efficacy of each individual sample might be negatively affected since highly potent sequences become diluted [MS06].

After the experimental process, a plate reader or a fluorescence microscope (see Section 2.2.3) is used to extract the phenotypic effect after the perturbations. A microscopic analysis allows the recording of several images per well. Thereby, much more cells can be screened in in comparison to screens performed on LabTeks (around 400-10.000 cells/well for well-plates and 100-400 cells/spot for LabTek). This enhances the statistical relevance of the data [ENR$^+$08].
In contrast to the microarray format, wells are physically separated and therefore cross-hybridization cannot occur. Moreover, the read-out is simplified since no spotting-matrix detection is necessary.

A disadvantage of microwell plates is that they cannot be stored as long as LabTeks. Erfle and co-authors presented in 2008 [ENR$^+$08] a protocol, which allows solid-phase reverse transfection in multi-well plates. In this method well plates are dried after adding the siRNA transfection solution into the wells. This allows the storage of the ready-to-transfect arrays for at least 12 weeks after production without losing transfection efficiency. This is not as good as for RNAi microarrays, but it offers a valid alternative if the storage is not the most important criterion for choosing an RNAi screening platform.

2.2.3 Image Acquisition

There are basically two different technologies for the image acquisition: plate readers, which are exclusively used to detect signals in microtiter plates and fluorescence microscopes, which are used for both microarray and multiwell plates. Like for DNA microarrays, the plate reader detects for each spot a single fluorescence read-out representing the phenotype.
Fluorescence microscopes are more sophisticated. They allow the acquisition of a multi-parametric read-out and therefore a much more detailed phenotypic classification [EP07]. For high-throughput screens this requires automated screening systems, offered for example by Olympus. An automatic Olympus fluorescence microscope quantifies data with its integrated proprietary image analysis software ScanR. The algorithms of ScanR are undisclosed.
Alternatively, customized software as for instance developed by Petr Matula [MKW$^+$09] can be used. It takes two-channel images for input: one channel represents the cell nucleus stained with the DNA-binding fluorescent agent DAPI (or HOECHST) and the other channel measures the viral protein production level using GFP intensities. In short, the software of Dr. Matula works as follows: first, each siRNA spot is localized by dividing the LabTek images into rectangles if LabTeks are used. For well-plates this step is not necessary. Then, in each spot the DAPI channel is used to segment the cell nuclei and the GFP channel to measure the mean virus signal intensity of each cell. The individual cells are then classified with certain criteria such as size, circularity or position on the array. See [MKW$^+$09] or Section 3.2.2 for more details.

2.3 RNAi Data Analysis

Measurements in biological experiments are never completely exact. During the whole experimental setup of RNAi screening and imaging several problems may occur where noise and errors are introduced. These errors can be purely due to technical problems such as pipetting, robot, or scanning failures as well as to an inhomogeneous staining, an inhomogeneous cell growing and a different transfection or infection efficacy. Moreover, different concentrations of the solvent solution in the edges of plates or different cell numbers in the individual positions are frequent. The silencing effect of individual siRNAs can lead to cell death or cell clumping which might cause background or saturation effects. Apart from problems occurring during transfection, viral infection may introduce even more data variation since it can lead to a different cellular behavior.

Systematic noise which affects entire plates or certain positions can be modeled and normalized. If the variation of the data is due to stochastic

errors, replicate measurements are required to find significant hits with the help of hypothesis testing [MHC+06, GCD+05].
After performing RNAi experiments, the generated data has to be analyzed to find possible sources of noise and errors, to control and quantify the quality of the data, to reduce systematic errors and to extract the biological signal.

During my time as a PhD student in the group of Prof. Dr. Lars Kaderali, I supervised Nora Rieber during her master thesis. She developed a pipeline for the analysis of RNAi data. The pipeline has been implemented in the statistical language R [R D09] as the Bioconductor [Bio11b, GCB+04] package RNAither [RKEK09b, RKEK09a]. The work includes the assessment of the data quality, data normalization and hit calling. In the following the most important methods of this pipeline are discussed to give an introduction of the state-of-the-art analysis methods for RNAi screening data.

2.3.1 Quality Control

In biological experiments the use of controls - positive and negative - helps to evaluate the quality of the data. Whereas negative controls should not show phenotypic effects, positive controls are supposed to achieve maximal degree in difference. For RNAi screens performed using an optimal design, positive and negative controls are used in a reasonable number on each individual plate. Controls allows the calculation of several different quality metrics such as the dynamic range or the Z' factor.

Dynamic Range (DR)

The ratio between the geometric means of the positive and the negative controls is defined as DR [MHC+06]:

$$DR = \frac{\mu_{neg}}{\mu_{pos}}. \tag{2.1}$$

It quantifies the separability of controls and thus, whether they worked as expected. For RNAi experiments which assume that the mean intensity of the positive controls is higher than the mean intensity of the negative controls, a small (near zero) DR shows a good performance of the data.

Z' factor

Z' factor [ZCO99] is comparable to DR, yet it is more sophisticated. It is defined as the ratio of the separation of the distributions of the negative and the positive controls with:

$$Z' = 1 - 3\frac{\sigma_{pos} + \sigma_{neg}}{|\mu_{pos} - \mu_{neg}|}, \tag{2.2}$$

where μ_{pos} and μ_{neg} are mean intensity values of the positive and negative controls, respectively, and σ_{pos} and σ_{neg} are their standard deviations. According to Zhang et al. [ZCO99], $Z' = 1$ demonstrates an optimal, $0.5 \leq Z' < 1$ a good and $Z' \leq 0.5$ a bad separation of controls.

Coefficient of Variation (CV)

Although, controls are essential for quality assessment, they are often not available or they did not work properly in some screens or at least on individual plates. The coefficient of variation [TOR+01] does not depend on controls and measures the data quality based on the reproducibility of results. It is defined as the ratio of the standard deviation of an siRNA (σ_{siRNA}) and its mean (μ_{siRNA}):

$$CV = \frac{\sigma_{siRNA}}{\mu_{siRNA}}. \tag{2.3}$$

The data is better the smaller (near zero) the CV value.

Correlation between Replicates

The correlation coefficient between pairs of replicates measures their relationship. Thus, it gives information about the reproducibility and reliability of the data. There are several different types to quantify the degree of correlation, i.e. Pearson's , Spearman's rank or Kendall's rank correlation coefficients (for details see [SH06]). The correlation coefficient is $+1$ for perfect correlation and -1 for perfect anti-correlation. Everything in-between indicates the degree of the linear relationship with being closer to zero, being less correlated (or less anti-correlated).

Data Visualization

Apart from assessing the data quality via metrics, data visualization and a visual evaluation is a good strategy. Most often used plots include:

- Histograms, density plots or boxplots of controls, individual plates or the whole screen to show the data distribution.
- Plate plots (sometimes also called heatmaps) to visualize the color-coded distribution of the intensity values on individual plates.

- Scatterplots of signal intensities versus cell counts or versus position number on the individual plate to illustrate whether the signal depends on other factors.
- Scatterplots between pairs of replicates to represent their correlation.

For more details about the individual plots and how to interpret them see for example [GCD$^+$05].

2.3.2 Normalization

Whereas a huge variety of different methods exist for DNA microarray data analysis, only some standard normalization techniques are available for the processing of RNAi data [BBH06, PGB10, RKEK09a]. Basically, normalization techniques can be separated into two groups: intra- and between-plate normalization. The first one aims in reducing systematic errors on individual plates. Examples are effects due to different cell counts or due to different positions of siRNAs on the plate (i.e. edge-effects).
The between-plate normalization removes systematic bias which occurs on different plates. Plates which are, for instance, spotted on distinct days or measured with a diverse microscopical setting, may have different overall signal intensities. Since data is varying across specific experimental setups, a standard normalization strategy which is suitable to all of them cannot be stated. Often, there is not only one possible solution, and also a combination of different methods which aim at different features of the data are thinkable [WRBB08].

Normalization on Controls

The most easiest way to normalize between different plates is to scale the intensity values based on the controls. Thus, assuming an additive error model, the mean or median of the controls of a plate is subtracted from each intensity value x of the same plate and the result is divided by the controls standard deviation:

$$norm(x) = \frac{x - \mu_{ctr}}{\sigma_{ctr}}, \tag{2.4}$$

where μ_{ctr} and σ_{ctr} are the mean (or median) and standard deviation (or median absolute deviation (MAD), see [SH06]) over all spot intensities of the controls on a plate. Whether for the normalization the negative or the positive controls shall be chosen, depends on the type of experiment. For RNAi data, negative controls are used in most cases.
Although this method is easy in computation and interpretability, it is sensitive to outliers [BSF$^+$09] and as mentioned in Section 2.3.1 controls are not

always available or show large variations within a screen. In this case, more sophisticated normalization strategies are recommended.

Normalization with Z-scores

A further data scaling strategy is called z-score normalization [MHC+06]. For each spot x, the z-score is defined as:

$$Z(x) = \frac{x - \mu_x}{\sigma_x}, \tag{2.5}$$

where μ_x and σ_x are the mean and standard deviation (or median and MAD, for a more robust analysis) over all spot intensities of one plate. This allows the quantification of the signal of each siRNA relative to the rest and thus, makes different plates comparable. However, this method can be used only if the assumptions hold, that most siRNAs in a screen do not have an effect and that they are equally distributed on the individual plates [BSF+09].

B-score Normalization

The B-score method [BSF+09] is a more robust analog to the z-score normalization. It allows the comparison of different plates since it scales the data according to the overall plate median. Additionally, it normalizes within a single plate by removing row and column effects.
It is calculated by subtracting the overall plate, the column and the row median from each siRNA:

$$Bscore(x_{ijp}) = \frac{x_{ijp} - (\mu_p + \hat{C}_{jp} + \hat{R}_{ip})}{MAD_p} = \frac{x_{ijp} - \hat{x}_{ijp}}{MAD_p} = \frac{r_{ijp}}{MAD_p}, \tag{2.6}$$

where x_{ijp} is the siRNA x in row i and column j on plate p. $\hat{x}_{ijp}$ is the fitted value computed by a two-way median polish [MT77] that estimates systematic measurement offsets for each row i ($\hat{R}_{ip}$) and column j ($\hat{C}_{jp}$). Furthermore, μ_p is the median of plate p and $MAD_p = median|r_{ijp} - median(r_{ijp})|$ for all i, j on plate p.

Lowess Normalization

Lowess (locally weighted polynomial regression [Cle79, Cle81]) normalization performs within-plate corrections. If RNAi data is multi-parametric different read-outs may depend on each other and these can introduce a systematic bias. For example positions on plates which show higher cell count may have higher signal intensities.
In order to remove this systematic error, a polynomial regression function is

fitted to the data for a given number of intervals. Since the polynomial approximation of data is better the smaller the intervals, a sliding window approach is used. Data points which are nearer to the estimated fit are weighted higher than more distant points. Combining the polynomials of each interval results in a smooth curve which models the data. The normalized signal intensities are the difference of the signal intensity values and the corresponding point on this curve.

2.3.3 Hit Calling

After data normalization the hit selection takes place. A commonly used practice is to use z-score normalized intensities which deviate from the bulk, so which are greater than or smaller than a certain threshold, as hit criteria. Often, this is done if there are not enough replicate measurements and thus, no significance level can be computed. The choices for an optimal threshold depend on the research aspect. If maximum saturation is of interest, so e.g. finding all genes involved in a certain pathway, the threshold should be chosen rather small. By contrast, if the goal is to find most significant hits, the threshold should be high, to reduce costs and time needed in the validation screen [WRBB08].

Apart from hits which show a clear difference to the rest, siRNAs with only small, but at the same time very robust effects are of interest as well. Therefore, hypothesis testing is used if enough replicated are available. This allows the distinction between significant phenotypic effects and those due to mere chance. If hypothesis testing is used for each siRNA, p-values are calculated and hits can be defined using both criteria: absolute z-scores higher than a given threshold and p-value smaller than a predefined significance level.

T-test

The t-test is an example of a parametric (assuming normally distributed data) testing method. It tells if the mean of a normally distributed data set differs significantly from the mean of a Gaussian distribution (one-sample t-test) or from the mean of another distribution derived from a normally distributed data set (two-sample t-test).
Assume that we have $n \in \mathbb{N}$ samples of a normally distributed data set with sample mean μ and sample standard deviation σ. If the null hypothesis (H_0) assumes that $\mu = \mu_0$ with μ_0 being a specified value (i.e. zero), then the t-test calculates:

$$t = \sqrt{n}\frac{\mu - \mu_0}{\sigma}. \tag{2.7}$$

H_0 is rejected with significance level α if:

$$|t| > t(1 - \frac{\alpha}{2}, n - 1), \tag{2.8}$$

with $t(1 - \frac{\alpha}{2}, n - 1)$ being the $(1 - \frac{\alpha}{2})$-quantile of the t-distribution with n-1 degrees of freedom.

The two-sample t-test, for two data sets having the same sample size, is calculated as:

$$t = \frac{\mu_1 - \mu_2}{\sigma_{1,2} \cdot \sqrt{2/n}}, \tag{2.9}$$

with μ_1 and μ_2 being the mean of the two sample data sets and $\sigma_{1,2} = \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}}$ with σ_1, σ_2 their standard deviations. The degrees of freedom is $2n - 2$ with n being the number of samples in each set of data.
Furthermore, there are two possible scenarios regarding the variances of the distributions: the standard t-test assumes equal variances and the Welch's t-test different ones. For more details see [SH06].

Mann-Whitney Test

The Mann-Whitney test is a commonly used non-parametric test which does not assume normal distribution. Originally published in 1945 by Wilcoxon [Wil46] it has been enhanced by Mann and Whitney in 1947 [MW47].

Chapter 3

RNAi Data Analysis

3.1 Introduction

The combination of RNAi screens and fluorescence microscopy results in high-dimensional image-based readouts where for each siRNA the viral fluorescence intensity signal of several hundreds of cells is quantified. Figure 3.1 shows an assembly of the raw microscopy images of an example RNAi LabTek array with 384 spots. In each spot, located in one of the 384 rectangles, one siRNA has been used to silence the respective gene in the overlying cells. One image has been taken per spot. The Figure shows overlaying fluorescence signal intensities of the DAPI (cell nucleus) and GFP (viral signal) channels. Hundreds of cells can be identified and characterized for each knockdown using image recognition software like described in Section 2.2.3.

This allows in contrast to siRNA screens using bulk measurements a more detailed view since for each knockdown a multi-parametric phenotypic readout can be quantified. The image of the LabTek given in Figure 3.1 clearly shows, that there are spots which have smaller intensities in the middle of the array. This is most probably due to noise for example based on an inhomogeneous staining or an inhomogeneous cell growing in some areas of the plate. To remove these effects an adequate normalization strategy is necessary. However, analysis methods for this type of data are still lagging significantly behind experimental developments. If the morphology or the cell-cycle state of a cell is not of primary interest as for example in [JCL$^+$09, WKIKG09, NWH$^+$10, FPK$^+$10], only some basic strategies as discussed in Section 2.3 are used for data analysis on an averaged phenotype. This results in an information loss of hundreds of individual cell measurements and thus, in the identification of false hits or a small reproducibility of results. Moreover, each cell's population context is not taken into account. This is in contrast to a recent study by Snijder and co-authors [SSR$^+$09]. The authors showed that the population context of a cell greatly influences the

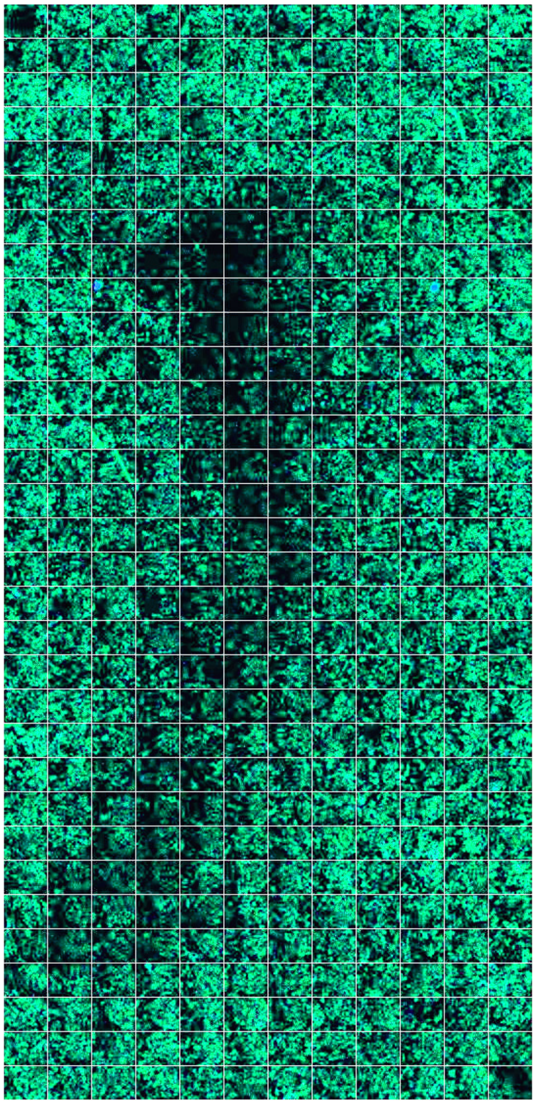


Figure 3.1: Raw microscopy image of an example RNAi LabTek array. On the array are 384 rectangles arranged in 12 columns and 32 rows. In each rectangle one siRNA has been spotted to knockdown the corresponding gene in the overlying cells. Shown are the fluorescence signal intensities of the DAPI (cell nucleus) and GFP (viral signal) channels together (see Section 2.2.3). In the middle of the array the rectangles have smaller intensities. This effect is most probably due to noise artifacts which should be normalized in a subsequent analysis.

variation in virus infection, endocytosis and membrane lipid composition. Therefore, the authors claim that determining each cell's context like cell size or cell shape is of great importance for the interpretation of phenotypic effects.

Although, the study by Snijder shows the great need of an analysis based on individual cell information, only few methods are accounting for this so far. Suratanee and co-authors [SRM$^+$10] proposed a spatial clustering approach (see Section 3.4). Fuchs and colleagues used multiparametric phenotypic profiles of RNAi screening data [FPK$^+$10] to cluster genes. The approach is based on morphological changes of individual cells within a cell population and this is used to identify new gene functions. Using multi-dimensional phenotypic similarity, the authors identified DONSON as a new component in the DNA damage. However, none of these methods propose strategies for normalization of cell signal intensities against the effects due to each cell's context.

Designing suitable statistical methods for quality control and hit selection is one of the most fundamental challenges in high-throughput experiments [Eis06]. The use of analytic metrics to assess and rank the effects of individual siRNAs in combination with hypothesis testing to control false positive and false negative rates are main approaches [CZK$^+$08]. However, data analysis based on individual cell measurements poses new methodological challenges and no publication addresses this problem so far in its full range. The methods presented by Fuchs *et al.* and Suratanee *et al.* provide a first step towards the analysis of RNAi screening data based on individual cell measurements, but they do not use population context for normalization. Thus, they do not offer a full analysis pipeline including normalization and statistical testing methods. Therefore, we developed a method where for the first time single cell measurements are used to normalize data and to define hits.

We present results on two high-throughput, high-content viral screens of Hepatitis C virus (HCV) and Dengue virus (DENV). We normalize the measurements of each individual cell based on the population context, and thus, we are correcting for a population bias present in the data. Furthermore, within-plate and between-plate normalization methods allow to remove spatial effects. Borrowing ideas from functional enrichment analysis, we calculate p-values based on individual cell data. This enables the statistical identification of significant siRNAs with a higher statistical power than when using bulk measurements.

This Chapter is structured as follows: First,we explain the experimental setup and details about the image analysis performed of both screens (Section 3.2). Then, we show how the state-of-the-art analysis method (AVERAGE) is applied to analyze the HCV data in the Section 3.3. Then, we discuss in Section 3.4 the strategy introduced by Suratanee *et al.* (RIPLEY) which has also been applied on the HCV screen [SRM+10]. In Section 3.5, we explain the methods of our newly developed approach (CELL-BASED) and present results of the HCV and the DENV screen. The Section 3.6 quantifies the performance of the AVERAGE, RIPLEY and CELL-BASED methods based on results of the HCV screen. Then, we provide a pathway analysis and a functional annotation of host dependency factors using our method on the HCV and DENV data in Section 3.7. Furthermore, we show in Section 3.8 how the population context influences the phenotype in a non-virus RNAi screen. Finally, we conclude and discuss the whole Chapter in detail in Section 3.9.

We note, that the data, the methods, some of the Figures as well as parts of the results presented in this Chapter have been submitted for publication in BMC Bioinformatics recently [KRK+11].

3.2 Data

In the laboratory of Prof. Dr. Bartenschlager, two different high-throughput, high-content primary RNAi screens have been performed. Both screens are targeting the same 719 human kinases. The first screen has been done by Ilka Rebhan and aims at the identification of host cell factors involved in Hepatitis C virus replication. Anil Kumar has carried out the second screen where the effect of each kinase is studied after Dengue virus infection.

3.2.1 Experimental setup

RNAi screening has been performed on LabTek chambered coverglass slides like described in [RRB+11], respectively [Kum10] for the HCV, respectively the DENV screen. In short, the experimental setup is as follows: seven different plates are used to target the 719 different human kinases. Each plate is repeated twelve (HCV) or six (DENV) times. Furthermore, three different siRNAs are used to target each kinase using the siRNA library from Ambion (*Silencer*® Human Kinase siRNA Library V3 (AM80010V3)). The siRNAs have been reversely transfected into Huh7.5 cells as described in [ENL+07]. Incubation time of the LabTeks after cell seeding was 36 hours for the HCV and 48 hours for the DENV screen. Then, cells of the HCV screen were infected with an HCV GFP reporter virus and 36 hours later immunostained with an GFP-specific antibody. For the DENV screen a wild type Dengue virus (New Guinea C strain) was used for infection and an immunostaining for envelope protein of the virus after 24 hours, to allow the quantification of the viral infection. Images of each siRNA spot are analyzed using a scanning microscope (Scanˆ R, Olympus Biosystems) with a 10x objective (Olympus, cat. no. UPSLAPO 10x).

All siRNA spots with less than 125 cells are omitted from the analysis in both screens to avoid inaccurate measurements due to too little cell numbers. Analogously, spots with more than 500 cells are excluded since a too dense cell population can lead to cell clumping and thus, to a different cell behavior or saturation effects. Additionally, the quality of the images is controlled for staining artifacts like out-of-focus images, intensity over-saturation or dirt particles by eye inspection, resulting in an overall exclusion of 15% of the images.

3.2.2 Image analysis and quality assessment

Beside the inspection by eye, the automatic image analysis system developed by Matula *et al.* [MKW$^+$09] has been used to analyze the image data of both screens. Here, the DAPI and GFP signal can be quantified for each LabTek. The first channel (DAPI) is representing the cell nuclei and the second channel (GFP) the virus signal intensity.
In brief, the algorithm for image analysis designed by Matula and colleagues contains four main steps:

1. Spot localization of knockdown (for RNAi microarrays)
2. Cell identification using the DAPI channel
3. Calculation of cell specific features
4. Measurement of viral fluorescence signal intensity

In the first step (required only if RNAi microarrays have been used) a rectangle is manually entered into the first image, which had been marked in advance. Then, this rectangle is copied to all other images according to a specific grid matrix. In each rectangle, the software positions a spot according to maximal difference of virus intensities between inside and outside of the spot. This represents the knockdown are where the siRNA has been spotted.

In the second step, the DAPI channel is used to segment single cell nuclei by applying an edge-based approach where a binary image f is calculated from the input image g. For this, the results of the gradient magnitude and the Laplacian operator are combined:

$$f(x,y) = \begin{cases} 1 & \text{if } |\nabla g| > T \text{ and } |\nabla^2 g| < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

with ∇ being the Nabla operator, $|\nabla g| = \sqrt{g_x^2 + g_y^2}$, $\nabla^2 g = g_{xx} + g_{yy}$, where g_x, g_y, g_{xx}, g_{yy} corresponds to first and second order partial derivatives of g using an approximation based on Gaussian derivative filters [YGvV+95]. Furthermore, T is a threshold automatically determined using the unimodal background symmetry method [vGvK11]. All the connected pixels with the negative Laplacian which contain at least one pixel of the image f are selected. After removing small connected components, the remaining components are morphologically closed and holes are filled. Finally, for all segmented objects several morphological features like size and circularity are calculated to identify cells [MKW+09].

The objects are then specified further in the third step using the following parameters:

- Position in spot (X- and Y-coordinates)
- Size of cell
- Size of nucleus
- Percentage of overexposed (saturated) pixels in cytoplasm
- Circularity of cell shape

In step four, the GFP channel is used to compute the virus signal of each cell by taking the mean intensity inside the nucleus neighborhood. We assume a direct proportionality between infection / replication efficiency and GFP signal intensity. The neighborhood is defined as non-overlapping rings around segmented nuclei (see [MKW+09]).

To quantify the data further, several features for each well or spot are computed:

- Number of objects
- Number of cells
- Size statistics of cells
- Mean virus signal intensity
- Percentage of overexposed (saturated) pixels in cytoplasm
- Percentage of infected cells

3.3 Data Analysis using AVERAGE Approach

The commonly used method at the moment, even for detailed microscopic read-outs, is to calculate the mean or median of virus signal intensities of all cells given in one well or spot [RRB$^+$11, BHS$^+$10, WRBB08, BBH06]. After summarizing the measurements the quality of individual spots or plates is assessed using methodologies like described in Section 2.3.1. Thereafter, standard normalization strategies like b-score, z-score or Lowess are applied (see Section 2.3.2). Then standard hypothesis testing procedures like t-test for normally distributed data or the Mann-Whitney Test for non-parametric testing are used (compare Section 2.3.3) [PGB10, BSF$^+$09, MHC$^+$06, BBH06].

3.3.1 Quality Control

The data of the HCV screen has been analyzed (and published in [RRB$^+$11]) using the AVERAGE method by taking the mean of the GFP signal intensities of all cells within one spot. Quality analysis on raw HCV data reveals that positive and negative controls are not perfectly separated within each replicate. An average DR of 2.8 ($\sigma = 0.41$) and an average Z' factor of -1.23 ($\sigma = 0.71$) is observed. Doing the same analysis for the DENV data results in dynamic range values which are slightly better (DR with $\mu = 2.15$, $\sigma = 1.67$) but mean Z' factors are even worse with -6.87 ($\sigma = 2.9$).
The average coefficient of variation of raw signal intensities results in 0.1 ($\sigma = 0.03$) for the HCV and in 0.26 ($\sigma = 0.1$) for the DENV screen, showing that the overall variance is higher for DENV data.

Furthermore, we calculated the mean and standard error of Pearson correlation coefficient of replicate measurements resulting in 0.2 ± 0.008 for HCV and 0.15 ± 0.01 for DENV of raw signal intensities. The large variability of results between the replicate measurements is not surprising since we have high number of replicates (twelve (HCV), six (DENV)) versus typically only two to three for well-plate assays.

3.3.2 Normalization

Exemplary, in Figure 3.2 a scatterplot of mean raw signal intensities of all cells per spot versus the number of cells in the corresponding spots of a randomly selected plate of the HCV screen is given (the remaining plates look similar). This indicates that the number of cells (measured in the DAPI channel) and

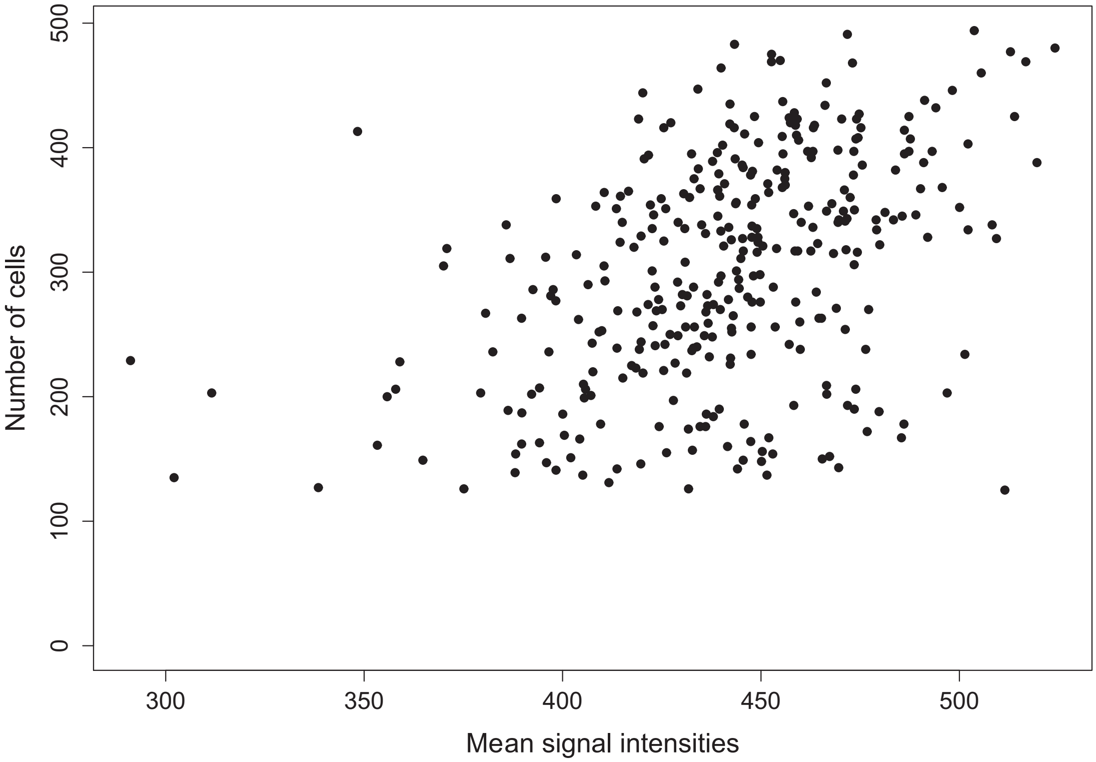


Figure 3.2: Mean raw signal intensities of all cells per spot are plotted against the number of cells in the corresponding spots of a randomly selected plate of the HCV screen.

the viral signal intensities (measured in the GFP channel) are correlated. To quantify this, we calculated the Pearson correlation coefficient of the mean virus signal intensities and the number of cells within each spot for all plates of the HCV and the DENV screen. Results are shown in Figure 3.3. Plates are enumerated based on the date where experiments have been done. In one experiment each of the seven LabTeks of the two screens have been performed once. For the HCV screen all but two of the plates are showing significant correlations between the two channels (p-value smaller than significance level of $\alpha = 0.05$ using two-sided Fisher's Z transformation test [SH06] on the null hypothesis that Pearson's correlation coefficient is equal to zero). In the DENV screen eight plates are significantly correlated and 16 significantly anti-correlated (p-value smaller than significance level of $\alpha = 0.05$ using two-sided Fisher's Z transformation). There are several possible reasons for the converse behavior of some plates in the DENV screen. It may be due, for example, to different environmental conditions like different temperatures when performing experiments on different dates. Independently from the source of the dependency of the two channels, they have to be de-correlated to unbias the data. For this, we applied the Lowess normalization strategy (see Section 2.3.2) on the data, to minimize the Pearson correlation coefficient values of the virus signal intensities and the cell counts in each spot (see Figure 3.4). We did this for all plates of both screens to ensure a unique treatment of the data, even if some plates did not show significant correlations before.

After normalization, none of the plates of the HCV screen show significant Pearson correlation coefficients with a p-value greater or equal to a significance level of $\alpha = 0.05$ using two-sided Fisher's Z transformation and only two plates of the DENV screen (p-value of 0.0016 and 0.0168). To conclude, in general the Lowess normalization successfully de-correlates the two channels in both screens.

Since the calculations of the DR and Z' factor showed that controls are not clearly separated and reliable on all plates, a normalization based on controls cannot be used for between-plate corrections. As B-score normalization is more robust than Z-score and it additionally corrects for spatial effects [MHC$^+$06, BSF$^+$09], we use B-score to normalize all the plates of the DENV and the HCV screen. Thereby, the high between-plate variations, quantified with the low Pearson correlation coefficient values between replicate plates, can be improved. Whereas the Pearson correlation coefficient (PCC) values averaged to 0.2 ± 0.008 for HCV and 0.15 ± 0.01 for DENV of raw signal intensities, after normalization they average to 0.28 ± 0.004 (HCV) and 0.23 ± 0.008 (DENV).

The twelve, respectively six replicate measurements of the HCV, respec-

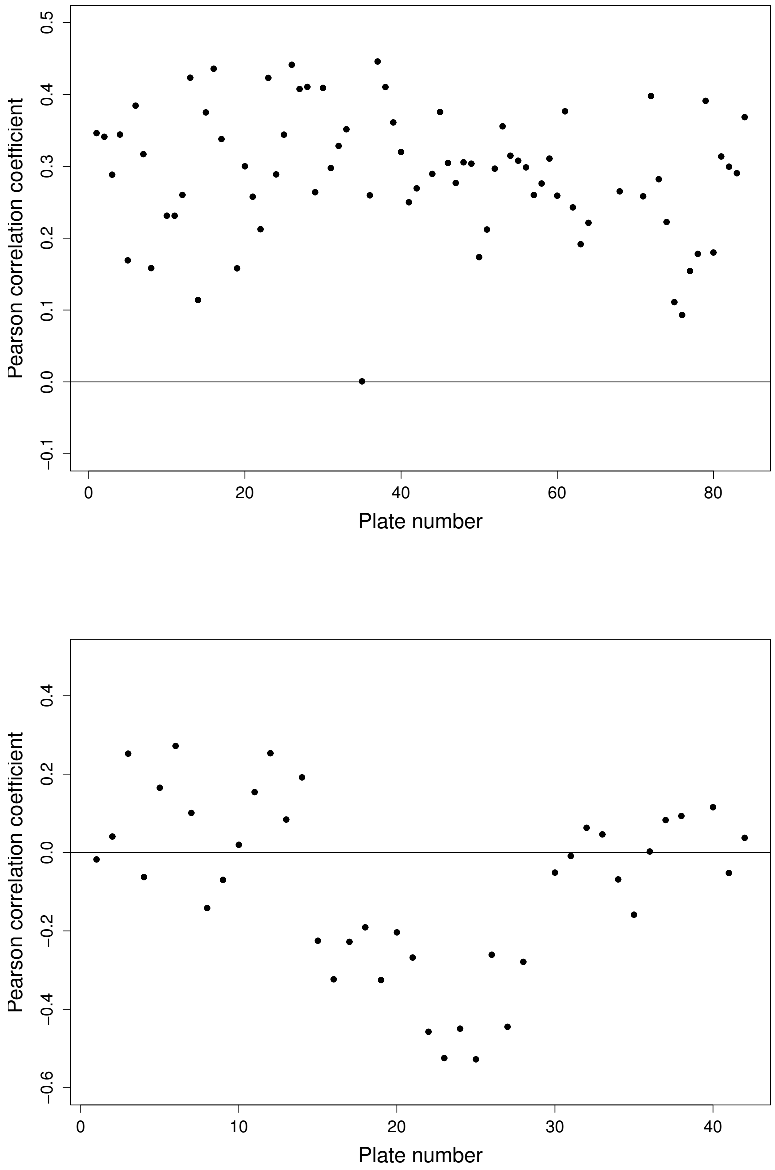
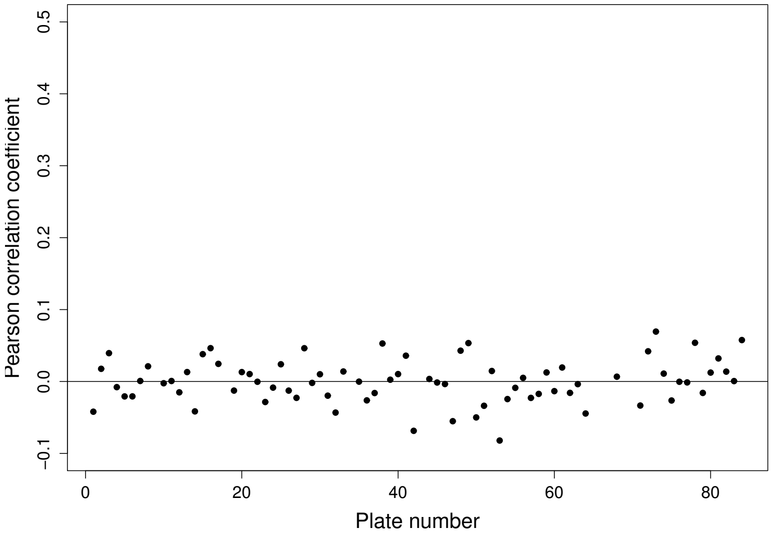


Figure 3.3: Pearson correlation coefficients of raw mean signal intensities of all cells per spot and the number of cells in the corresponding spots of all plates of the HCV screen (left) and the DENV screen (right). Plates are enumerated based on the date where experiments have been performed (in one experiment each of the seven LabTeks of the HCV and DENV screen have been performed once).



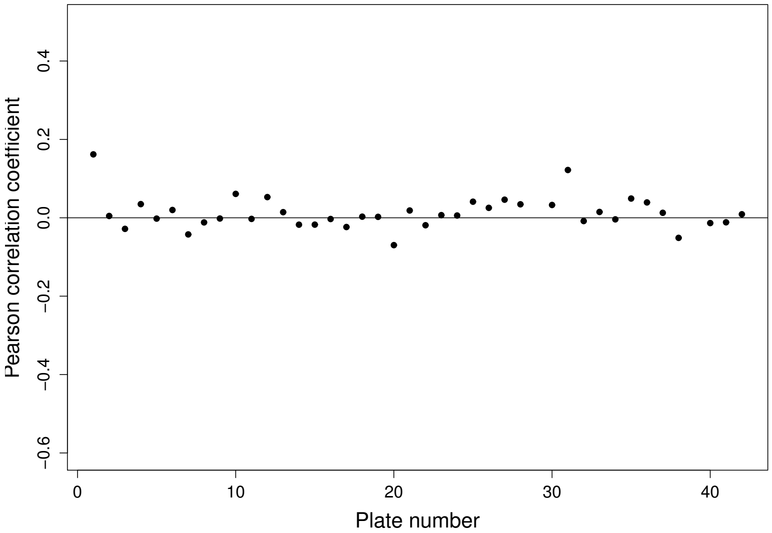


Figure 3.4: Pearson correlation coefficients of lowess normalized mean signal intensities of all cells per spot and the number of cells in the corresponding spots of all plates of the HCV screen (left) and the DENV screen (right). Plates are enumerated based on the date where experiments have been performed (in one experiment each of the seven LabTeks of the HCV and DENV screen have been performed once).

tively the DENV screen were then summarized by taking their mean. Since data is normally distributed (p-value$< 2.2 \cdot 10^{-16}$ for both data sets using a Kolmogorov-Smirnov test against normal distribution, see [SH06] for details) the one-sample Welch's t-test on the null-hypothesis of differential GFP signal intensity per well has been used to calculate the significance of siRNAs. Then, a combined thresholding for z-scores > 1.5 and for p-values ≤ 0.05 are applied for hit identification.

Data analysis was performed using the statistical program R [R D09] and the RNAither [RKEK09a] package from Bioconductor [Bio11b, GCB$^+$04].

3.4 Data Analysis using the RIPLEY Approach

The HCV screening data has additionally been analyzed by Suratanee *et al.* in the year 2010 [SRM$^+$10]. The authors assume that viral infection is mainly spread by cell-to-cell contacts and as a consequence of this, infected cells form cell clusters. By systematically analyzing clustering patterns of individual knockdowns, the RIPLEY approach aims at detecting knockdowns where infected cells do not form such clusters and thus, the viral infection efficiency is diminished.

Suratanee and colleagues use microscopy images and the point pattern analysis method called Ripley's K-function to define the degree of spatial clustering of infected and and non-infected cells for each individual siRNA. For this, they classify cells based on their virus signal into infected and non-infected using a threshold defined by maximizing the difference in infection rates between positive and negative controls. Then, the clustering behavior is studied using the inhomogeneous K-function as described by Baddeley and co-workers [BMW00]:

$$K_{inhom}(r) = \frac{1}{|A|} \sum_{i=1}^{N} \sum_{j=1, i \neq j}^{N} \frac{e_{ij} I_r(d_{ij})}{\lambda(y_i), \lambda(y_j)}, \tag{3.2}$$

with $r > 0$ being a pre-given radius and N the number of cells observed in the given area A (whole image). Furthermore, d_{ij} is the Euclidean distance between cell i and j, and $I_r(d_{ij})$ equals to one if $d_{ij} < r$ and zero otherwise. The edge-correction factor e_{ij} is calculated by the border method [Rip81]. Moreover, $\lambda(y_i)$ and $\lambda(y_j)$ are estimated intensities at spots y_i and y_j using the Gaussian kernel smoother [BMW00]. The radius range of r has been chosen with 35% of the shorter side of the image. For the computation of the clustering score the area between the curves of the inhomogeneous K-function and a simulated random distribution is calculated. If the score is positive the curve of the inhomogeneous K-function is above the curve of simulated random distribution and this indicates a tendency for clustering. If it is negative no clustering is given. The authors computed clustering scores for infected and non-infected cells. The final clustering score is calculated as the difference of the score of infected and non-infected cells (see [SRM+10] for details).

The RIPLEY approach has been applied on the HCV screen and clustering scores have been calculated for each knockdown on raw images without performing any normalization [SRM+10].

3.5 Data Analysis using the CELL-BASED Approach

In the AVERAGE approach all signal intensities for one spot are summarized and individual cell measurements are completely neglected. Although, the RIPLEY method uses a spatial approach which takes the spatial information for infected and non-infected cells into account, other features defining each cell (like cell morphology) are ignored. Thus, neither of the two methods use individual cell information for normalization and for the identification of knockdowns.
Since Snijder and colleagues [SSR+09] reported that each cell's context highly influences its phenotypic effect, our CELL-BASED approach aims at analyzing the data based on several technical as well as cell context features. After normalizing against the effects due to these features, we use an approach based on the idea of gene set enrichment analysis to define hit siRNAs which are significantly reducing the viral replication efficiency. For this, we assume that for each knockdown there are two populations of cells coming from two normal distributions of infected and non-infected cells.

3.5.1 Gaussian Mixture Model to Define the two Subpopulations of Infected and Non-Infected cells

Figure 3.5 shows example microscopy images of the fluorescence signals of the nuclei (DAPI) and the virus (GFP) signal for one negative (scrambled) and one positive control (CD81, targeting the viral receptor of HCV) of the HCV screen. We note, that the data in the DENV screen behaves similar and therefore, the images are not shown explicitly. The Figure shows that cells have an inhomogeneous virus signal even within the same spot (see GFP channel of the CD81 control). In addition to the within spot heterogeneity,

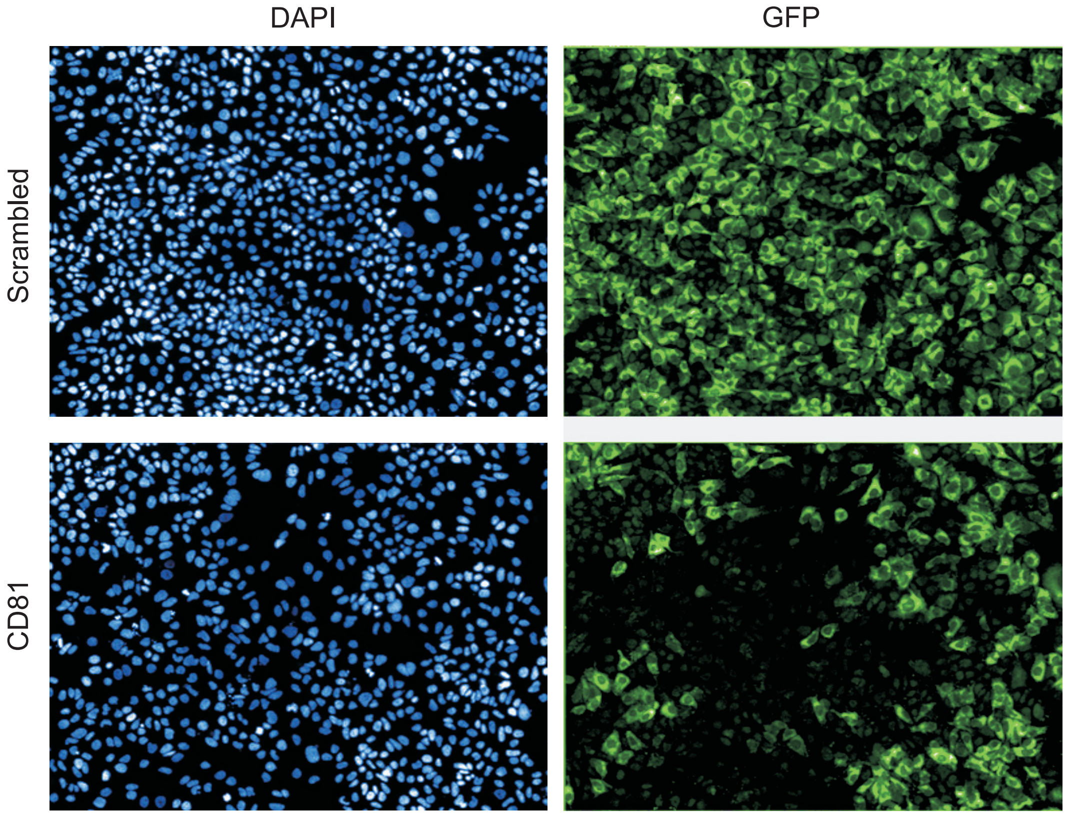


Figure 3.5: Microscopy images of one negative (scrambled) and one positive (CD81) control in the HCV screen. The left panel shows the nuclei (DAPI) channel and the right panel the virus signal (GFP).

both controls clearly show a difference in the expressed GFP signal. As discussed in Section 3.3.2 the overall viral signal intensity of individual spots is dependent on the number of cells in that spot. However, this does not explain the different viral intensity signals in the case of controls, where the number of cells are almost the same. In numbers, there are on average 326.82 (standard error of 5.44) cells per spot for positive controls which directly target the viral genome (HCV321 and HCV138) and 338.32 (standard error of 3.43) cells per spot for negative controls (scrambled). Using a two-sided, two-sample Welch's t-test on the cell-counts in the positive and negative controls, this results in a p-value of 0.11. Therefore,this shows no significant apoptotic effect of the corresponding knockdowns. The same observation holds for the DENV screen with 279.81 cells (standard error of 7.64) and 287.0 cells (standard error of 5.3) per positive (DV-NS5 and DV-NS2) and per negative controls (scrambled), respectively (p-value= 0.46).

However, although in the DAPI channel no significant effect is measured for controls, Figure 3.5 suggests that the knockdown of CD81 has a clear effect on the viral replication since the GFP signal is clearly reduced in comparison to the GFP signal of the negative control. To amplify this, we show in Figure 3.6 the distribution of log-GFP signal intensity values of each cell of the negative and positive controls, respectively, for the HCV screening data (DENV screening data looks similar). The Figure shows that for both types of controls the distribution of the signal intensities is bimodal and we assume that the two populations are reflecting infected and non-infected cells. The brown curve corresponds to the signal distribution of all cells in the entire screen and it perfectly matches the behavior of negative controls. The positive controls instead show pronounced differences concerning the sizes of the two

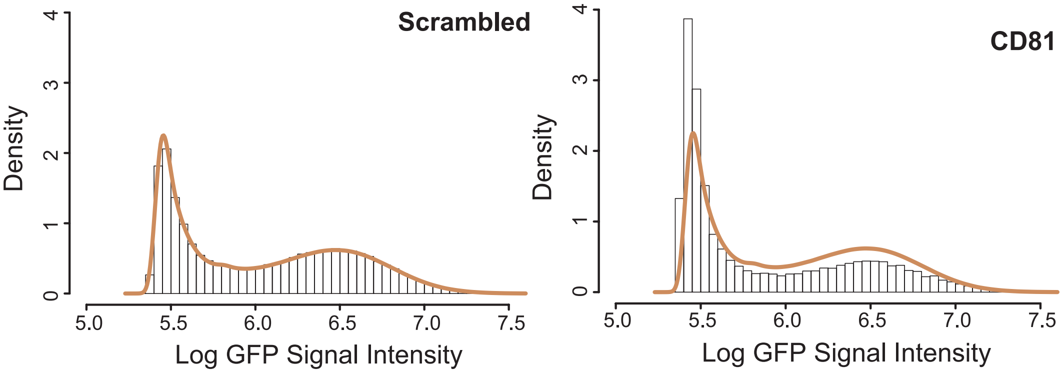


Figure 3.6: Histogram of single cell log-transformed raw GFP intensity signals for controls (negative control: scrambled; positive control: CD81) in HCV screen. The brown curve indicates the signal distribution of all cells in the entire screen.

subpopulations. This indicates that the knockdown influences the bimodal distribution where under optimal conditions (perfect transfection, knockdown and infection efficiency) positive controls would show only background GFP signal intensities, and negative controls maximal GFP signal intensities. Thus, we assume that the size of the two distributions (mean and standard deviation of the two Gaussians) is influenced by viral infection. To quantify this, we use the Gaussian Mixture Model methodology.

Methodology

Since we aim at distinguishing between infected and non-infected cells of the given signal intensities measured in the GFP channel of both screens, we divide the data into two clusters. For this, we perform a model-based clustering using a Gaussian mixture model with two components of the log-transformed GFP cell signal intensities x as follows:

$$f(x) = \alpha_1 \mathcal{N}(x|\mu_1, \sigma_1^2) + \alpha_2 \mathcal{N}(x|\mu_2, \sigma_2^2), \tag{3.3}$$

where α_k with $k \in \{1, 2\}$ is the probability that an observation comes from the k^{th} mixture component and $\alpha_k \in [0, 1]$ with $\alpha_1 + \alpha_2 = 1$. Furthermore, μ_k and σ_k^2 are the mean and variance of each Gaussian.
The likelihood function for data coming from two univariate mixture components is

$$p(x|\alpha, \mu, \sigma) = \prod_{i=1}^{n} \left(\alpha_1 \mathcal{N}(x_i|\mu_1, \sigma_1) + \alpha_2 \mathcal{N}(x_i|\mu_2, \sigma_2) \right), \tag{3.4}$$

with $\mathcal{N}(x_i|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp -\frac{(x_i-\mu_k)^2}{2\sigma_k^2}$ being the Gaussian density function and x_i with $i \in \{1, \ldots, n\}$ being the signal intensity of the ith cell. For more details see [FR07, Bis07].

In order to find the best model, model parameters α_k, μ_k and σ_k are estimated using Expectation-Maximization (EM) to maximize the likelihood. EM consists of an iteration of two steps:

1. 'E'-step: estimates the conditional probability that an observation x_i belongs to group k:

$$p(group = k|x_i) = \frac{\alpha_k \mathcal{N}(x_i|\mu_k, \sigma_k^2)}{\alpha_1 \mathcal{N}(x_i|\mu_1, \sigma_1^2) + \alpha_2 \mathcal{N}(x_i|\mu_2, \sigma_k^2)}. \tag{3.5}$$

2. 'M'-step: computes the maximum likelihood parameter estimates given the probabilities from the 'E'-step:

Raw data	infected	non-infected
HCV	$5.5 \pm 2.4 \cdot 10^{-5}$	$6.32 \pm 1.4 \cdot 10^{-4}$
DENV	$6.71 \pm 2.5 \cdot 10^{-4}$	$7.14 \pm 2.6 \cdot 10^{-4}$

Table 3.1: Mean and standard error of bimodal Gaussian mixture components for infected and non-infected cells of the raw data of the HCV and DENV screen.

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{n} p(group = k|x_i) x_i \tag{3.6}$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_{i=1}^{n} p(group = k|x_i)(x_i - \mu_k)^2 \tag{3.7}$$

$$\alpha_k = \frac{N_k}{n}, \tag{3.8}$$

where $N_k = \sum_{i=1}^{n} p(group = k|x_i)$ and $i \in \{1, \dots, n\}$.

Initial estimates are obtained by splitting data into 0.5-Quantiles. So data has been sorted according to increasing signal intensities and then, 50% of the cells with smallest intensities are used to define group $k = 1$, and 50% of the cells with highest intensities form group $k = 2$. Now, the initial μ_k, σ_k^2 can be computed directly by calculating the mean and standard deviation of the cell intensities given in each group and $\alpha_k = 0.5$ in the initial estimation. Using these values, the EM begins with its first 'E'-step. It has been shown that this yields good results for maximum likelihood estimations of mixture models in practice [FR02]. Each update of the parameters increases the likelihood function until the algorithm converges (see [Bis07] for details).
When using different initial estimates for the EM this can result in different mixture components of the Gaussian distributions. However, since this clustering is only used to support the idea of having two different groups of cells (infected versus non-infected) and since we are not using the clustering for the later data analysis, we are not further optimizing initial estimates.

Results

The results of these calculations are summarized in Table 3.1 for the log-transformed raw virus signal intensities of the HCV and DENV screening data. The Table shows that the two Gaussian distributions of infected and non-infected cells have well separated means.

In Figure 3.7 the distribution of the raw signal intensities of all cells of the HCV screen is demonstrated. In addition, it illustrates the distribution of simulated data. For the data simulation, we generated data points with two ($k \in \{1, 2\}$) Gaussian distributions with μ_k, σ_k^2 and N_k as computed with the Gaussian mixture model. The Figure demonstrates that the mixture of the two Gaussians fits the data well. The same holds for the DENV data (not shown explicitly). The discrepancies between the two distributions may be due to noise of the data.

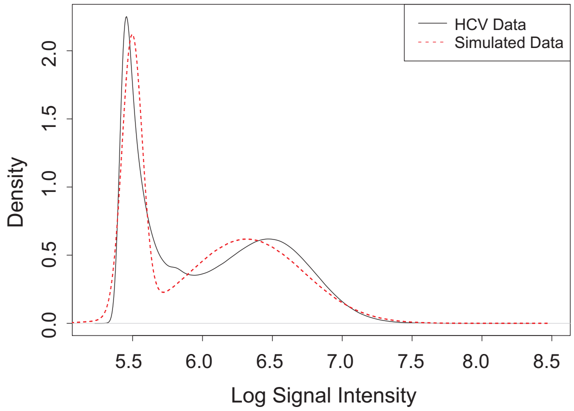


Figure 3.7: The black (straight) line shows the signal distribution of raw log GFP signal intensities of all cells in the HCV screen. The red (dotted) line represents the distribution of simulated data generated with the parameters as computed with the Gaussian mixture model.

3.5.2 Normalization

Cell Context

According to Snijder and colleagues [SSR+09], each cell's population context greatly influence the phenotypic response. We assume that different viral signal intensity distributions between different spots are due to different amounts of infected versus non-infected cells of individual spots. However, since cells within individual spots of our data are not showing homogeneous signal intensities (compare Figure 3.5 in Section 3.5.1), these effects are supposed to be mainly due to population context parameters like described by Snijder *et al.*. Therefore, we use a similar procedure like Snijder and colleagues to study the population effect on virus replication in RNAi data of the HCV and the DENV screen.

For this, each cell is described with six different population features. They are calculated based on the DAPI and GFP channels extracted with the image recognition software implemented by Petr Matula [MKW$^+$09] (compare also Section 3.2.2). The six features are:

1. *Nucleus Size*: Size of the nucleus.
2. *Cell Size*: Size of the cell.
3. *Cell shape*: Circularity of the cell nucleus.
4. *Cell number*: Number of cells per spot.
5. *Cell Density*: Density of surrounding cells.
6. *Population Border*: Location (center or border) of the cell in a local cell population.

The feature "nucleus size" is equal to the number of pixels after nuclei segmentation given by the image-analysis software like discussed in 3.2.2 for each cell. There is no stain for the cell cytoplasm and therefore, the size of the nucleus is used to approximate feature "cell size" by dilating the cell nucleus mask. For this, a ring around the nuclei of fixed size (six pixels to and six pixels out of the nuclei center) is used. The "cell size" is given by the number of pixels of the ring. According to this, the features "nucleus size" and "cell size" are highly redundant with the "nucleus size" being the one having higher precision since it is directly based on the DAPI channel. That is why the feature "cell size" is several times replaced by the feature "nucleus size" in the following analysis.
The third feature is a descriptor of the shape of a cell, namely the inverse of the cell circularity $C = \frac{4\pi A}{P^2}$ where P is the perimeter and A the area of the cell nuclei after segmentation. Thus, the feature is defined with:

$$\textit{cell shape} = \frac{P^2}{4\pi A}. \tag{3.9}$$

The inverse is used to transform the values into a range between zero and one with being equal to one for perfectly circular objects and bigger than one for more complex shapes.
Although there are no significant differences in cell counts for control spots (compare Section 3.5.1) we showed in Section 3.3.2 that there is a dependency of signal intensities and cell numbers per spot in both data sets. Therefore, feature four -"cell number"- is given by the number of objects identified as cells after cell nuclei segmentation within the spot area. It accounts for different signal intensity levels due to different cell counts.
A Gaussian kernel density estimator [BA97] based on nuclei centers (X- and Y- coordinates) is used to approximate the local cell density in feature number five.
Finally, using the coordinate information given for each cells nuclei center

within a spot, the sixth feature defines whether a cell is located in a local population or not. Since there are on average 261.33 ($\sigma = 159.68$), respectively 252.55 ($\sigma = 124.19$) cells per spot in the HCV, respectively the DENV screen, we split each spot into a 15x15 grid. In each rectangle at least one cell is located in an average spot. We count how many cells are positioned in each single rectangle. Based on this, we define cells located in a rectangle neighboring to at least one empty rectangle to be at the border of a local population. This feature is binary with each cell being either at a population border or not. To measure the influence of the grid size on the results, we use additionally a 10x10 and a 20x20 grid in every spot. Both different grid sizes do not change final results of the HCV and the DENV host dependency factors.

There are several possible features which can be additionally quantified using microscopic images. Example are the cell-cycle status, apoptotic cells and cell elongation [FPK$^+$10, NWH$^+$10, WKIKG09, JCL$^+$09, SSR$^+$09]. However, for high-throughput data as presented here we have several hundreds of cells per spot, 384 spots per plate, and seven different plates repeated on twelve (HCV) and six (DENV) replicates. This results in millions of individual cell measurements for each feature and thus in a high dimensionality. Therefore, we focus in this thesis on the features named above to make the analysis also useful for large-scale screens such as whole-genome arrays. Nevertheless, the inclusion of many more features for the data analysis is generally possible and may be advantageous.

Technical Features

Apart from the population context features, we use four features to account for technical (spatial or between-plate) artifacts:

1. *Spot Border*: Location of the cell in spot (center or border of spot).

2. *Row*: Row signal intensity median.

3. *Column*: Column signal intensity median.

4. *Plate*: Overall plate signal intensity median.

For the calculation of the first technical feature, the grid of rectangles like for the calculation of feature “population border” is used. If a cell is within a rectangle which is located at the border of the spot, the cell is defined to be at the spot border and centered otherwise. The location of each cell is calculated by the coordinates of its nuclei. We use this feature to show whether there are phenotypic effects based on the position of cells within a spot. It is a binary feature like the “population border” feature.

The next two features estimate the individual row and column effects for each plate and therefore, they address the spatial variation within plates. They are calculated by taking the median of the signal intensities of all cells given in the spot of the corresponding row or column per plate. For this, we are not doing any correction for different cell numbers of spots within a single row or column. This factor is addressed already using the cell context feature “cell number”.
The last technical feature takes the median of all signal intensities of all cells for each plate. This is an estimator for overall plate effects and accounts for between-plate variation.

Estimation of Within-Bin and Between-Bin Variability

To estimate the effects on viral infection of the features named above, we calculate a *variability-ratio*. For this purpose, we first reduce the huge dataset size (similar to the procedure given in Snijder *et al.*), for a better data handling by defining groups of cells with similar properties. The data in each feature is divided into 5%-quantiles. Clearly, we use only two bins for the two binary features which account for the effect whether a cell is at the border of a bunch of cells or at the border of a spot.

We calculate the standard deviation of the viral cell signal intensities within each bin $i \in \{1, \ldots, n\}$. The average of these standard deviations is used to estimate the *within-bin variability* (σ_{within}) for each feature:

$$\sigma_{within} = \frac{1}{n} \sum_{i=1}^{n} \sigma_i, \tag{3.10}$$

where σ_i denotes the standard deviation in the ith bin.

To calculate the *between-bin variability* ($\sigma_{between}$) we compute the standard deviation between the means of different bins for each feature:

$$\sigma_{between} = \sigma(\mu_1, \ldots, \mu_n), \tag{3.11}$$

where μ_i with $i \in \{1, \ldots, n\}$ denotes the mean of the signal intensities in the ith bin.

The variability-ratio ($Ratio_{var}$) is now calculated for each feature by dividing the between-bin variability through the within-bin variability:

$$Ratio_{var} = \frac{\sigma_{between}}{\sigma_{within}}. \tag{3.12}$$

Feature	HCV screen	DENV screen
Nucleus Size	12.96	16.24
Cell Size	12.42	16.3
Cell shape	1.81	5.62
Cell number	7.58	13.89
Cell Density	2.32	3.22
Population Border	6.57	9.72
Spot Border	4.9	7.84
Row	4.45	10.86
Column	4.29	11.91
Plate	14.35	38.9

Table 3.2: Percental variability-ratios of each of the cell features in the HCV and the DENV screen.

We used, additionally to the 5%-quantiles, which results in 20 bins, different bin sizes of 10, 50 and 100 corresponding to 10%-, 2%- and 1%-quantiles, respectively, to measure whether this significantly influences the variability-ratio. We calculated for each feature the CV (see Section 2.3.1) of the variability-ratios of all bin sizes and observed CVs much smaller than 1 every time ($CV_{max} = 0.24$, $CV_{\mu} = 0.05$, $CV_{\sigma} = 0.08$ for the HCV data; $CV_{max} = 0.32$, $CV_{\mu} = 0.05$, $CV_{\sigma} = 0.1$ for the DENV data). Thus, the variations of the results are not significant.

To compare how largely the individual features influence viral infection in comparison to the total variation observed in a population, we calculate percental variability-ratios ($Ratio_{var} * 100$ with $Ratio_{var}$ like given in Equation 3.12) of raw data. The results are listed in Table 3.2 for HCV and DENV data, showing considerable variability for individual features in both screens. For a better comparison of the percental variability-ratios of individual features, we visualized the results for both screens additionally in Figure 3.8 (feature “cell size” has been replaced by “nuclei size” due to the reasons explained above). The Figure clearly illustrates, that the highest influence on viral infection in both screens is due to a technical feature, namely feature “plate effects”, with a percental variability-ratio of 14.35% for HCV and 38.9% for DENV (compare Table 3.2).
This is not surprising, since technical variability occurs in most screens and has been reported several times before [BSF^{+}09, MHC^{+}06]. However, the second most important parameter is the population context feature describing the cell size. This confirms findings by Snijder and co-authors [SSR^{+}09] who showed that the cell context highly influences viral infection.

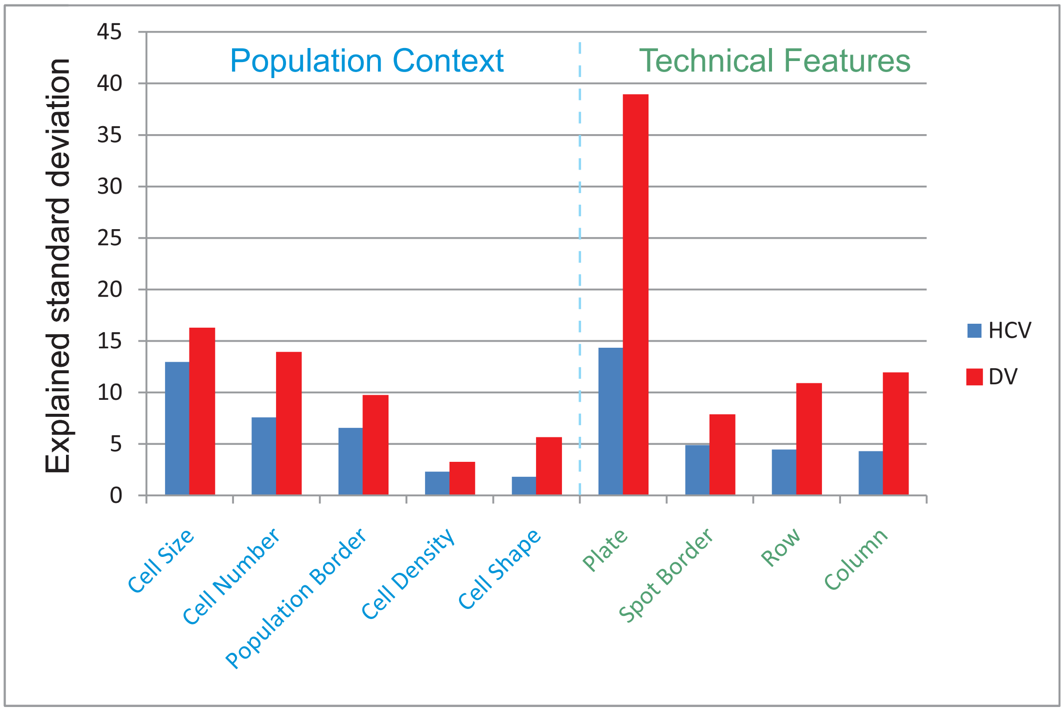


Figure 3.8: Percental variability-ratio of cell context and technical features of the HCV and the DENV data. Shown are the ratios of the between-bin standard deviation to the average within-bin standard deviation, as a measure of the fraction of the variation explained by the cell population or technical feature under consideration. This quantifies the explained standard deviation of the features with respect to single-cell viral infection efficiency. Blue bars: HCV screen, red bars: DENV screen.

Nevertheless, we observed considerable differences concerning the main contributing factors of population features for DENV in comparison to Snijder and colleagues. They measured six features for each cell: the size of the population to which it belongs, its local cell density, its position on a cell cluster edge, its cell size, and its mitotic and apoptotic state. They showed that the location of a cell at the edge or in the middle of a cell cluster is the most important feature, followed by the local cell density. The cell size, which is the most important cell context feature in our analysis, is the least important factor in the study performed by Snijder *et al.* (see [SSR$^+$09] for details).
This difference may be due to varying experimental conditions: First, both studies use different cell lines for the experiments (Huh7.5 (screen presented here) versus HeLa (Snijder *et la.*)). Second, Snijder and co-authors used 96 well-plates whereas we use chambered coverglass slides (LabTeks). As already reported in Section 2.2, this results in quantitative differences of seeded cells (around 400-10.000 cells/well on well-plates in comparison to 100-400 cells/spot on LabTeks). Furthermore, individual wells are physically separated using well-plates but not when using LabTeks. Third, the data measured by Snijder and colleagues did not consider any knockdowns.
We suppose that due to at least one of these differences the cellular behavior and thus, the viral signal is influenced in a different way within the two studies. As a consequence of this, Snijder *et al.* identified not the same factor having the biggest influence on virus infection like we do.

Figure 3.9 is illustrating the relative importance of the five population context features (again “cell size” was replaced by “nuclei size”) for the HCV and DENV screen in comparison to the total variability of the cell context features. We note that the percentage of each population context is calculated when assuming that all population features sum up to 100. However, this is most probably not the case since not all features are supposed to be independent from each other. The “cell size” for instance can be influenced by the “cell number” or the “cell density”. Nevertheless we use this notation to indicate once more that “cell size” is the most important feature (42% HCV, 33% DENV), followed by “cell number” (24% HCV, 28% DENV) and then by the position of a cell in a local population (21% HCV, 20% DENV). Local “cell density” and “cell shape” are the least contributing factors, however, they nevertheless influence viral infection with 7% (both viruses) and up to 12% (DENV), respectively.

In Figure 3.10 the mean and standard deviation of the two most important population context features “cell number” and “cell size” for the 20 individual bins are shown for the HCV and DENV data to show how these features influence the viral signal intensities. The feature “cell size”, for example, shows a small decrease of the log signal intensities for low bin numbers, so for small cells. Then, it increases and finally decreases again. This behavior is

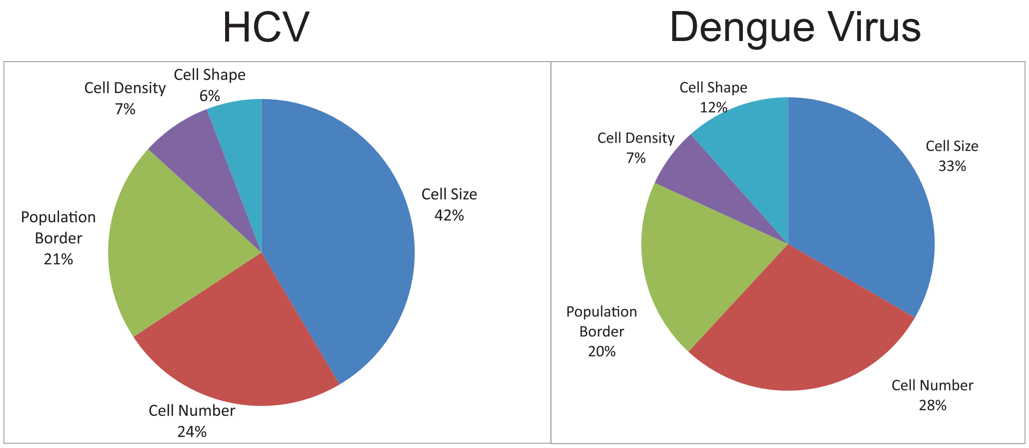
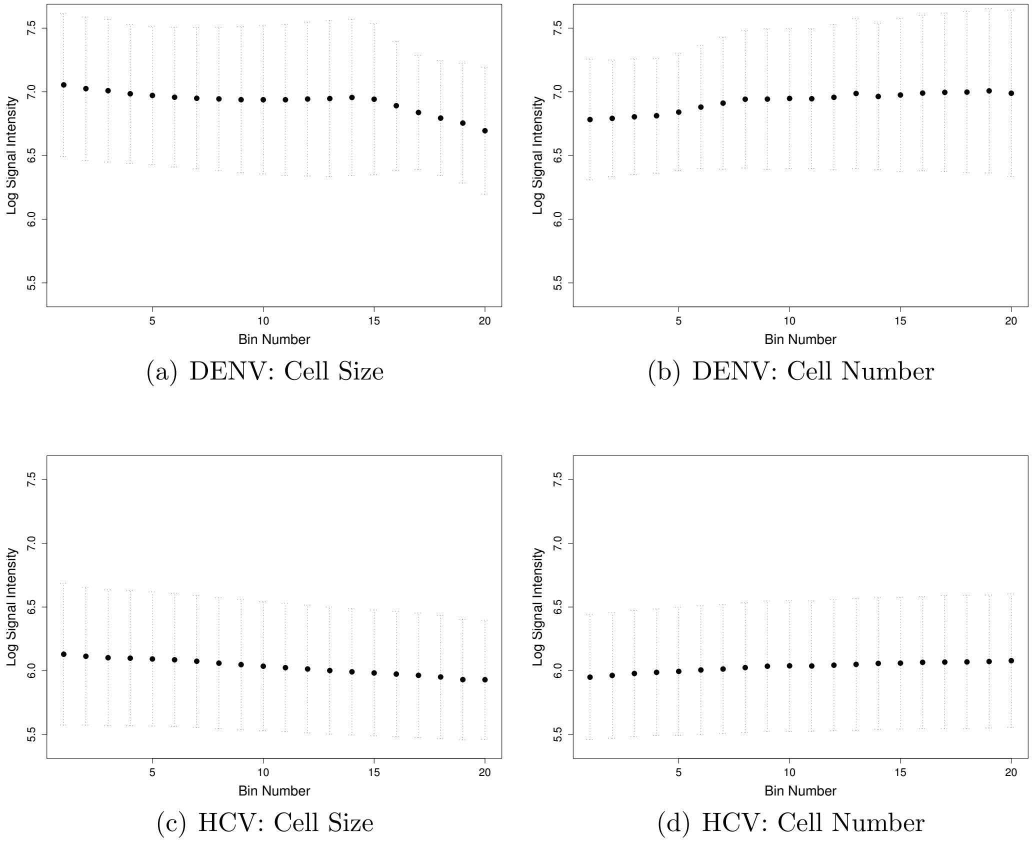


Figure 3.9: Relative explained standard deviation for cell context features for HCV and DENV in comparison to the total variability due to cell context features.



(a) DENV: Cell Size

(b) DENV: Cell Number

(c) HCV: Cell Size

(d) HCV: Cell Number

Figure 3.10: Mean and standard deviation of two selected features of the two screens for the 20 individual bins. Low bins correspond to low cell sizes/cell numbers.

similar also for the different bin sizes of $n = \{10, 50, 100\}$ (see Appendix A).

To test whether the data is linear or not we used the Harvey-Collier test for linearity. This test performs a t-test on the recursive residuals [HC77] computed with a linear regression model on the log signal intensities and the raw features (without binning). We did not use binned features since the binning has been used to allow a better data visualization and to identify most important features by the calculation of the variability-ratio. For the later data normalization we use the full range of individual measurements to ensure the most sophisticated analysis as possible.
The null hypothesis of the Harvey-Collier test states that the true relationship is linear which means that the mean of the recursive residuals equals zero. Based on a significance level of $\alpha = 0.05$ the null hypothesis is rejected for both the HCV and DENV screen (p-values$\leq 2.2 \cdot 10^{-16}$) for all features of the DENV and HCV screen, except for the "spot border" feature of HCV (p-value $\leq 2.987{\cdot}10^{-7}$) and the "column feature" of DENV (p-value $\leq 1.22{\cdot}10^{-4}$) which demonstrates the non-linearity. Due to these non-linear effects of the population context and technical features, we use use a non-linear normalization strategy.

Non-linear Normalization Strategy: MARS

For normalization of the single cell intensity values against the ten features we use multivariate adaptive regression splines (MARS) developed by Jerome H. Friedman in 1991 [Fri91]. MARS is a non-parametric regression technique producing continuous models in a greedy heuristic approach.
The method assumes that a response variable y is dependent on predictor variables $\mathbf{x} = (x_1, \dots, x_n)$, given N measurements for all variables $i \in \{1, \dots, n\}$, so $y = (y_1, \dots, y_N)$ and $x_i = (x_{i1}, \dots, x_{iN})$. Furthermore, the system that has generated the data is described by

$$y = f(x_1, \dots, x_n) + \varepsilon. \tag{3.13}$$

The parameter ε is the error term and models the dependency of y to factors apart from the given predictor variables, and which are not observed or measured. We assume ε to be zero. Function f captures the joint predictive relationship of y on $\mathbf{x}$.

The aim of MARS is to build a function $\hat{f}(\mathbf{x})$ that approximates $f(\mathbf{x})$ based on the given data. For this purpose MARS uses a weighted sum of multivariate spline *basis functions* $B_m(\mathbf{x})$:

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m B_m(\mathbf{x}), \tag{3.14}$$

where a_0 is the coefficient of the constant basis function B_1 and each a_m for $m \in \{1, \ldots, M\}$ is the coefficient of the basis function B_m given by the best fit of the data. The number M of basis functions and the parameters associated with them are directly determined by the data using residual sum of squares (RSS) (for details see [SH06]) and a modification of generalized cross-validation (GCV) [Fri91]. To ensure that the resulting function $\hat{f}$ is continuous at all points, each basis function B_m consists of the product of one or several truncated cubic functions characterized by three "knots". For more details see [FR95, Fri91].

In the case of RNAi data the response variable is defined by the single cell intensity measurements and the predictor variables by the ten features.
To normalize against the features, we subtract from each intensity measurement y_j for $j \in \{1, \ldots, N\}$ its fitted value after having calculated the best fit $\hat{f}(\mathbf{x})$ of the viral signal intensities:

$$y'_j = y_j - \hat{f}(x_{1j}, \ldots, x_{nj}), \tag{3.15}$$

where y'_j is the jth corrected intensity value which estimates the residuals accounting for population and technical artifacts.

To show that the normalization minimizes the effects of the individual features on viral infection, we calculated the percental variability-ratios (Equation 3.12) of the normalized data and compared it with those of the raw data for HCV and DENV. Results are shown in the Tables 3.3 and 3.4, respectively. The Tables demonstrate that MARS decreases the variability-ratios considerably. In general, the variability-ratios are up to 14.39% (HCV) and 38.9% (DENV) before we normalized the data and they are decreased to a minimum of 4.89% (HCV) and 9.84% (DENV) after normalization.
There are two features ("spot border" and "cell density") in the DENV data which show slightly higher variability-ratios after than before normalization (after: 9.84%, before: 7.84% for "spot border" and after: 3.23%, before: 3.22% for "cell density"). We assume that this can be explained by the general high variability (compare Section 3.3) of the DENV data and the very high feature space of the huge data set. Due to this, MARS is not able to perfectly fit the data since it also aims at avoiding overfitting by using a generalized cross-validation [FR95].

To study whether the assumption stated above that a non-linear regression will result in a better fit than a linear one, we calculated the goodness-of-fit (R^2) of the MARS and of a linear regression. The results listed in Table 3.5 show that MARS has higher R^2-values at least for the HCV data. For the DENV data both values are equal.

Feature	before normalization	after normalization
Nucleus Size	12.96	1.21
Cell Size	12.42	1.31
Cell shape	1.81	1.27
Cell number	7.58	0.56
Cell Density	2.32	2.28
Population Border	6.57	1.52
Spot Border	4.9	4.89
Row	4.45	0.42
Column	4.29	0.47
Plate	14.35	0.94

Table 3.3: Percental variability-ratios for each feature in the HCV screen before and after normalization.

Feature	before normalization	after normalization
Nucleus Size	16.24	3.38
Cell Size	16.3	1.82
Cell shape	5.62	1.12
Cell number	13.89	1.24
Cell Density	3.22	3.23
Population Border	9.72	0.81
Spot Border	7.84	9.84
Row	10.86	0.89
Column	11.91	0.8
Plate	38.9	1.7

Table 3.4: Percental variability-ratios for each feature in the DENV screen before and after normalization.

	linear	MARS
HCV	0.0344	0.0354
DENV	0.1464	0.1464

Table 3.5: R^2 values of the linear and the MARS regression for the HCV and DENV data.

3.5.3 Gene Set Enrichment Analysis Approach

In Section 3.5.1 we showed that the raw log virus signal intensities of the HCV and the DENV data can be clustered into two Gaussian distributions of

Normalized data	infected	non-infected
HCV	$-0.5 \pm 3.63 \cdot 10^{-5}$	$0.34 \pm 1.24 \cdot 10^{-4}$
DENV	$-0.72 \pm 1.71 \cdot 10^{-4}$	$0.21 \pm 1.71 \cdot 10^{-4}$

Table 3.6: Mean and standard error of bimodal Gaussian mixture components for infected and non-infected cells of the normalized data of the HCV and DENV screen.

	positive control	negative control
HCV	0.61 ± 0.002	0.4 ± 0.001
DENV	0.42 ± 0.002	0.21 ± 0.001

Table 3.7: Mean and standard error of mixture coefficients of the uninfection cell component probability for positive and negative controls in the HCV and DENV screens.

clearly separated means. This holds also for the normalized data. To verify this we applied the model-based clustering using a Gaussian mixture model of two components on the normalized HCV and DENV signal intensities. The average and standard error of the resulting two Gaussian distributions are given in Table 3.6 for both screens. The two population means of infected and non-infected cells (-0.5 versus 0.34 for HCV and -0.72 versus 0.21 for DENV) are clearly distant from each other.

Furthermore, Figure 3.6 in Section 3.5.1 indicates that the raw signal intensities of cells within one spot have a bimodal distribution (infected versus non-infected cells) with different mixture components (mean and standard variation) of the two Gaussian distributions for positive and negative controls. Table 3.7 lists the mean mixture coefficients of the uninfection cell component probability $\alpha_{non-infected}$ for the positive and negative controls in both screens. Since, the probability that an observation is coming from the population of the uninfected cells is higher (HCV: 0.61, DENV: 0.42) for positive than for negative controls (HCV: 0.4, DENV: 0.21) in both screens, we conclude that knockdowns of genes required for viral infection or replication (virus host dependency factors (HDF)) have a higher proportion of cells with weak signal intensities.

To identify virus HDFs we want to find knockdowns which result in a shift of the distribution away from the distribution of negative controls, towards an increased number of non-infected cells showing only background signal. Therefore, we use the distribution of the cells within one spot to assign each

knockdown with a significance value.
For this, we apply the approach of gene set enrichment analysis (GSEA) to test whether cell intensities after normalization given in one spot are enriched to the infected or non-infected population.

GSEA

The idea of GSEA, which is one of the most popular strategies for detecting differentially expressed gene sets, has been first introduced in the field of gene expression analysis by Mootha *et al.* [MLE+03]. The GSEA used in this work is based on the methods proposed by Sweet-Cordero [SCMS+05]. This is the standard Kolmogorov-Smirnov test (e.i. [SH06]) applied on two running sums which denote the number of sorted differentially expressed genes which are, or which are not in the given gene set.
Using GSEA in the context of RNAi screens requires basically one change of the original usage. Unlike in the analysis of gene expression data, the sets are defined not by genes but by cells. These sets can be specified based on different biological concepts, for instance, cells coming from one spot, one siRNA or one gene.

GSEA starts with a list D with N samples and computes a statistical score based on the correlation of the measurements (g_j) to the phenotype of interest for all $j \in \{1, \dots, N\}$. A variety of statistical scores, e.g., the t-score or signal-to-noise-ratio can be used in this step ([AS09]). Based on this score, the list D is sorted and a running sum statistic RS is calculated for each predefined collection of genes $G_1, G_2, \cdots, G_m$. The sorted list is processed from top to bottom and the two running sums RS_{G_k} and $RS_{\bar{G}_k}$ are calculated. RS_{G_k} is increased every time a sample belongs to G_k and $RS_{\bar{G}_k}$ each time a sample belongs to the complementary set $\bar{G}_k$:

$$RS_{G_k}(i) = \sum_{\substack{g_j \in G_k \\ j \leqslant i}} \frac{1}{N_{G_k}} \tag{3.16}$$

$$RS_{\bar{G}_k}(i) = \sum_{\substack{g_j \notin G_k \\ j \leqslant i}} \frac{1}{N - N_{G_k}}, \tag{3.17}$$

where N_{G_k} is the number of $g_j \in G_k$.
Finally, an enrichment score ES_{G_k} for each G_k is defined as the maximal deviation from zero of DIF, where DIF is the difference of the running sum G_k

and its complementary set $\bar{G}_k$:

$$\mathrm{DIF}_i(G_k, \bar{G}_k) = RS_{G_k}(i) - RS_{\bar{G}_k}(i) \tag{3.18}$$

$$ES_{G_k} = \mathrm{DIF}_j(G_k, \bar{G}_k) \text{ where } j = \arg\max_i |\,\mathrm{DIF}_i(G_k, \bar{G}_k)|. \tag{3.19}$$

GSEA and RNAi Data

In the case of RNAi experiments, cells within each single spot k are considered as predefined sets G_k and the accumulation on top or bottom of the sorted list of all cell intensities in the total screen is evaluated. The running sums are calculated based on the ranked normalized viral signal intensities measured in the GFP channel. Virus host dependency factors which decrease the viral signal are characterized by a positive ES (enriched to the left and thus to the population of infected cells).
To assess the significance of the obtained ES we use permutation testing. We randomly permute the assignment of the cells with a specific spot, in each plate. Then we calculate an ES of the permuted cells (ES_{perm}) for each spot and each plate. Thereafter, we take the average of the computed ES_{perm} of all plates. The distribution of the resulting averaged ES_{perm} values is used to calculate significance levels for the observed (unpermuted) data. The bonferroni method accounts for multiple testing. We use a significance level of $\alpha = 0.05$ on the corrected p-values and a positive $ES \geq 1.5$ times the standard deviation of the combined ES_{perm} to define hits.

In Figure 3.11 the DIF of the sorted cell intensity values for negative and positive controls of a randomly chosen plate of the HCV screen is shown. The remaining plates, and those of the DENV screen behave similar. Importantly, the Figure visualizes that the DIF is different for the positive and the negative controls. The DIF value is increasing until a maximum of about 0.2 is reached and then decreasing again for positive controls. In contrast, the DIF of the negative controls is fluctuating around zero. High positive ES values are an indicator for virus HDF and values around zero indicate that the respective knockdown is not influencing viral replication or infection.

Figure 3.12 shows the histogram of the median ES for each siRNA in the HCV and DENV screen. Positive and negative controls are marked explicitly. Interestingly, the positive and negative controls are perfectly separated for the HCV data, but not for the DENV data where some positive controls are not working properly. Most probably this is due to the higher variability of the DENV data (see Section 3.3). However, since other quality measures like other controls on the same plates or correlation between replicates were fine, we decided to not remove the plates of the affected controls.

Figure 3.11: Computed DIF values for the four positive and seven negative controls of a randomly selected plate of the HCV screen.

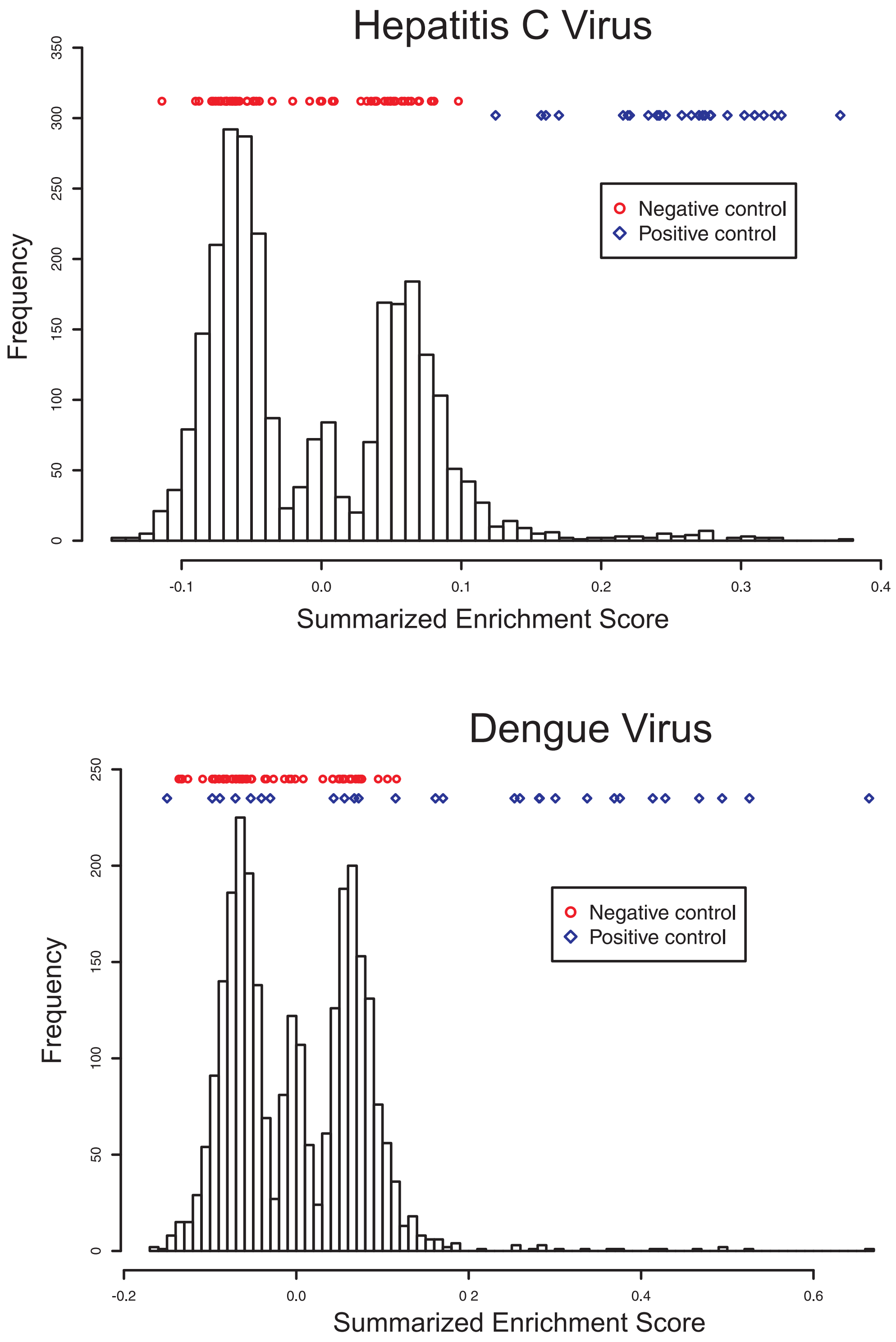


Figure 3.12: Distribution of the median *ES* of all plates of the HCV and DENV screens. The location of positive and negative control scores are indicated by blue diamonds and red circles, respectively.

In Figure 3.12 we observe furthermore, that there are three peaks of the histograms in both screens. This tri-modality effect can also be seen in Figure 3.13, which illustrates each summarized *ES* per siRNA in the whole HCV screen. The red curves visualizes the *ES* sorted by increasing order. The tri-modality is caused by the computation of the median siRNAs of replicated plates. Since we have an even number of replicates (twelve for HCV and six for DENV) the median is around zero if exactly one half of the replicates has positive and the other half has negative *ES* values. This reflects the peak in the middle of both screens which is located around zero and corresponds to siRNAs not having an effect on the phenotype. The siRNAs which have the majority of replicates with positive or negative *ES* values occur in the right or left peak, respectively. This tri-modality effect does not occur when using the mean which, however, is less robust to outlier siRNAs.

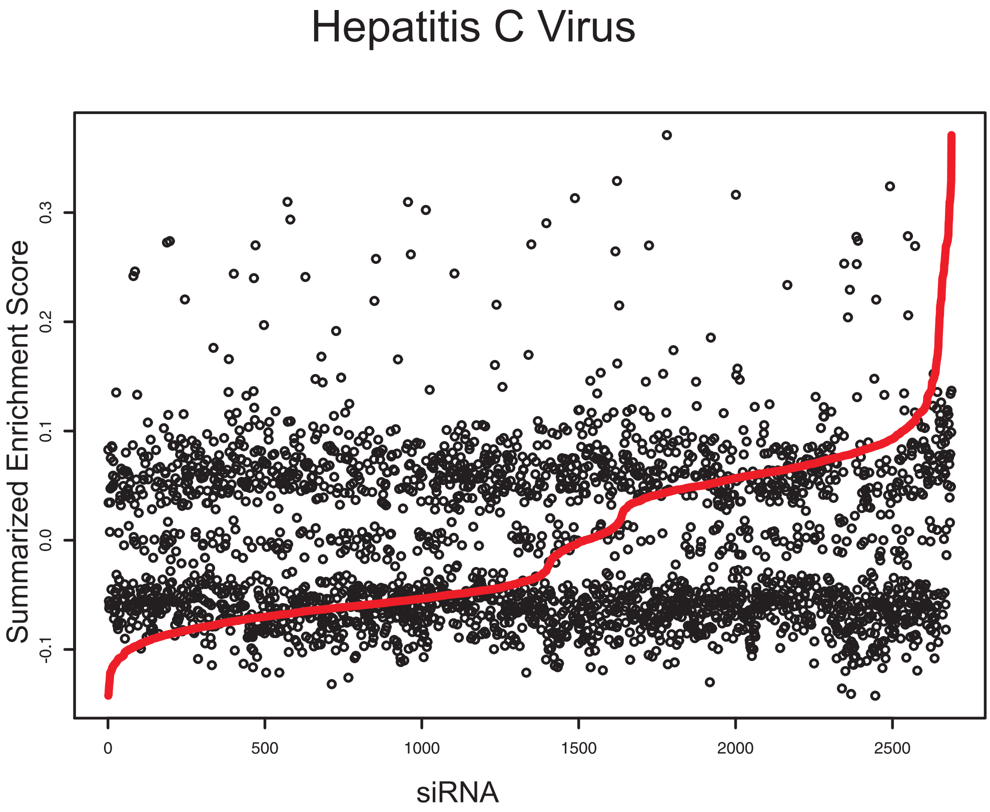


Figure 3.13: Median *ES* over replicates of each siRNA in the HCV screen. The siRNAs are plotted in the sequence as spotted on the plates and the red line shows the siRNAs sorted by increasing *ES*.

3.6 Hepatitis C Virus Host Dependency Factors

Using the CELL-BASED approach like presented above for normalization and hit-scoring results in a hit list of 54 HDF which are significantly reducing HCV replication (see Appendix B.1). We compared our results with results derived with the AVERAGE and RIPLEY methods described in Sections 3.3 and 3.4, respectively. We computed pairwise Pearson correlation coefficients between the z-scores, clustering scores and the enrichment scores calculated with the AVERAGE, RIPLEY and CELL-BASED method, respectively. Whereas the scores of the CELL-BASED and RIPLEY method are anti-correlated (PCC=-0.48), the results of the AVERAGE and RIPLEY as well as the AVERAGE and CELL-BASED method do not show any correlation (PCC=0.02 and PCC=0.002, respectively).

Defining appropriate thresholds is a crucial step in defining relevant hits and different filters may impose biases [Gof08]. Obviously, this is especially true for the three scores compared in this study, which have been computed differently. The z-score has been calculated based on how many standard deviations the mean virus signal intensity per spot is above or below the average of zero. The enrichment score is based on whether the virus signal intensities of cells given for an individual spot are enriched in the set of infected or non-infected cells. And finally, the clustering score indicates whether infected cells show a local cell clustering within a spot.

Thus, especially for the results where individual scores are not correlated, overlaps between individual hit lists may depend on the score thresholds. Therefore, we decided to uniquely use the significance level of 5% for hit identification as well as a score threshold of 1.5 times the standard deviation of the underlying score distribution for all methods. For the analysis using the RIPLEY method however, this would result in a hit list of only 14 genes which is considerably less than for the other two approaches. Therefore, we used all genes which have a p-value smaller than 0.05 and a negative clustering score like proposed in the original publication [SRM+10].

Figure 3.14 shows the overlap of the hit lists, with only six genes being significant HDFs found by all three methods. The overlapping genes are the positive controls HCV321 and HCV138 which directly target the viral RNA genome as well as CD81 which is the main entry receptor of HCV. Moreover, the remaining overlapping genes are *phosphatidylinositol 4-kinase alpha* (PI4KA) and *casein kinase II subunit alpha* (CSNK2A1) which have already been reported to play a role in the HCV replication cycle (see [LBN+09, VPC+09, BTP+09, TBP+09, TLDR+09, BCH+09], respectively [KLC99]) and *fms-related tyrosine kinase 4* (FLT-4) which has been suggested

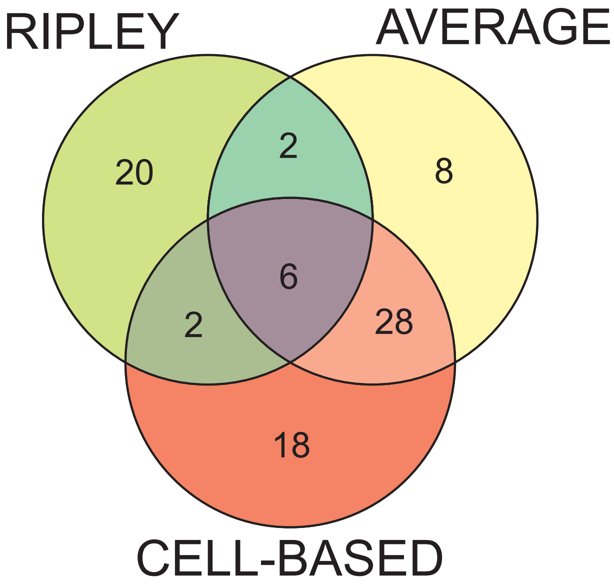


Figure 3.14: Venn-Diagram of the hits at the gene level using CELL-BASED, AVERAGE and RIPLEY analysis methods.

to be related to HCV in earlier publications [AS09, SRM+10].

Furthermore, 44 genes have been identified using the AVERAGE approach. Among them, 34 genes are also found using the CELL-BASED method which results in an overlap of 68% and indicates high agreement between the two methods. Interestingly, this is in contrast to the small Pearson correlation coefficients (PCC=0.002) computed on the scores of both methods. Out of the 30 genes identified with RIPLEY only 10 (33%) could be confirmed by the other two methods, although the Pearson correlation coefficients on the scores of RIPLEY and CELL-BASED was highly anti-correlated (PCC=-0.48). RIPLEY and AVERAGE show an overlap of only eight genes (18%).

The small overlap of the RIPLEY method with the other two approaches may be explained by the fact that the data is not normalized, neither against technical, nor against cell context parameters. However, as shown in Section 3.5.1 theses parameters can highly influence the virus signal intensities. Although, RIPLEY is considering the local cell neighborhood and does not summarize virus signal intensities of individual cells like in the AVERAGE approach, it is just a first step towards the analysis of RNAi data based on single cell measurements. We assume that the method can be immensely enhanced by considering normalization procedures like MARS presented above.
The AVERAGE method, on the other hand, normalizes against technical features or cell numbers. Individual cell measurements, however, are completely

neglected. In the CELL-BASED approach we therefore combined both ideas and normalized against technical and population context parameters by using single-cell measurements.

3.6.1 Sensitivity and Specificity of Controls

We assume to produce more reliable hits when using individual cell data. To evaluate this we calculate whether our approach identifies positive and negative controls. We compute sensitivity and specificity values on controls for different p-values and scores used for hit-calling. Then, receiver operator characteristic (ROC) curves (see for example [SH06, Faw06]) are produced and the area under the curves (AUC) is calculated for the three methods. We note that in the two-class problem (positive and negative controls) random guessing would correspond to a diagonal line in the ROC plot and an AUC of 0.5.

In Figure 3.15 the ROC curves for AVERAGE, CELL-BASED and RIPLEY are shown. Although all ROC curves are much better than random guessing, the CELL-BASED curve lies above the others and thus, outperforms AVERAGE and RIPLEY.
This is furthermore represented in the AUC values which is best for CELL-BASED with 0.99 in comparison to 0.95 and 0.87 for AVERAGE and RIPLEY, respectively. For the AVERAGE method a loess normalization was used to normalize for general trends between the mean viral signal intensities and the number of cells within one spot. Furthermore, b-score normalization was applied to normalize against spatial plate effects. The AUC values show, that the AVERAGE method cannot minimize the introduction of false positive and false negative for controls on a single spot level as good as the CELL-BASED method by purely normalizing against technical artifacts and cell counts. Thus, our approach is superior in terms of both, sensitivity and specificity of identifying positive and negative controls.

Since our analysis approach consists basically of two independent methods (normalization against the features using MARS and the statistical test based on the idea of GSEA) we analyze which of the two methods contributes more to the increased performance of classifying controls. Therefore, we use each of the two methods independently on the HCV data. For the first method (MARS-ONLY) we take raw log virus signal intensity values and normalize them against the features. Then, we use RNAither [RKEK09a] to average the cell intensities of one spot and compute z-scores for each spot. By applying a threshold of 1.5 times the standard deviation of the z-scores of each replicate

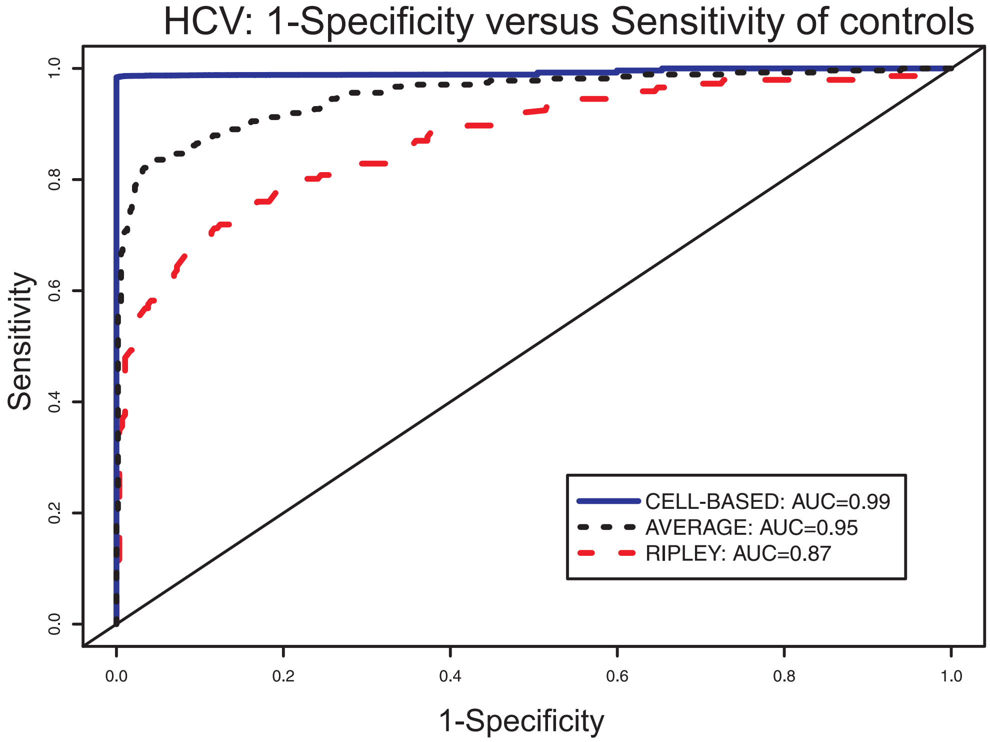


Figure 3.15: Receiver operator characteristic analysis of identified positive and negative controls in the HCV screen. Sensitivity and specificity of controls is computed for different thresholds on computed z-scores, ES values and clustering scores using AVERAGE, CELL-BASED and RIPLEY methods, respectively. For all three methods additionally a significance level of 5% is used.

we define hits on an individual spot level. We note, that the calculation of significance levels for each spot is not possible in this method.
For the second method (GSEA-ONLY), we calculate ES on the raw log virus signal intensities and use the nonparametric statistical test based on permutations to calculate p-values for each spot. We use bonferroni corrected p-values ≤ 0.05 and $ES \geq 1.5$ times the standard deviation of median summarized replicate ES for finding individual spots which significantly decrease viral replication.
We perform for both methods a ROC analysis on individual control spots and compute AUC values. MARS-ONLY results in an AUC of 0.971 and GSEA-ONLY in an AUC value of 0.987. We note, that the GSEA-ONLY method is applied on the raw intensities of the individual cell measurements. In contrast, the single-cell data has been summarized for each spot after the normalization in the MARS-ONLY method. Nevertheless, both individual methods are able to yield better results in comparison to the AVERAGE (AUC=0.95) and RIPLEY (AUC=0.87) approach. Yet, the combination of MARS-ONLY and GSEA-ONLY in our CELL-BASED analysis approach gives the best result when compared to the stand-alone methods.

In addition, we summarize replicate measurements of MARS-ONLY by taking their median and use a two-sample, two-sided Welch's t-test to define significance values for the individual siRNAs. We use an alpha threshold of 0.05 on uncorrected p-values and 1.5 times the standard deviation of z-scores to define significant siRNAs. The same is done for GSEA-ONLY, although p-values for individual siRNAs are calculated based on the *ES* using the nonparametric test. The overlapping genes of the resulting hits for

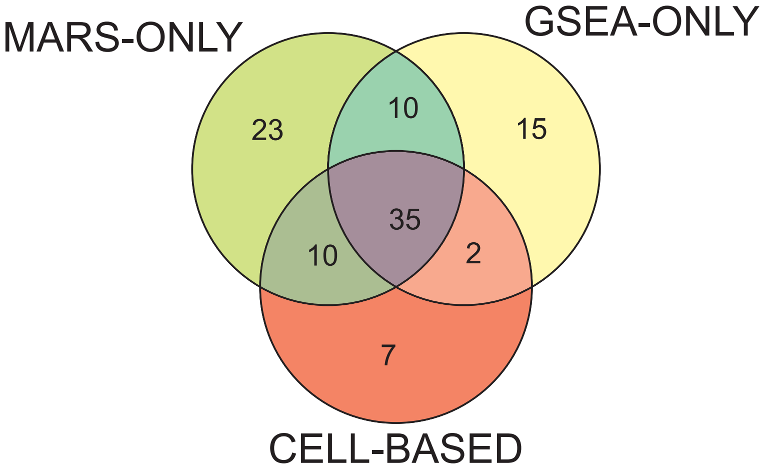


Figure 3.16: Venn-Diagram of the hits at the gene level using CELL-BASED, MARS-ONLY and GSEA-ONLY analysis methods.

GSEA-ONLY, MARS-ONLY and CELL-BASED are 35 (see Figure 3.16).

Among the 54 hits found using the combined CELL-BASED method 47 (87%) are also found with the two independent methods.

To further compare the CELL-BASED method with the other approaches, we compute the robustness of results over individual replicates for all of them. We calculate average pairwise Pearson correlation coefficients of the scores over the twelve individual replicates in the HCV screen. Similar results are obtained for CELL-BASED, MARS-ONLY, GSEA-ONLY, and AVERAGE with 0.26 ± 0.004, 0.26 ± 0.004, 0.29 ± 0.005 and 0.28 ± 0.004 (mean ± standard error), respectively. RIPLEY performs worst with 0.11 ± 0.007 which is even less than Pearson correlation coefficients of raw data (0.2 ± 0.008).

3.6.2 Validation screen

Published by Reiss and co-authors [RRB+11] a secondary validation screen has been done on hit genes identified by the AVERAGE analysis method. We use this validation screen to perform a more detailed evaluation of our method. Here, we compute the sensitivity and specificity on the subset of genes tested in the validation screen for varying z-score thresholds for the validation results and varying ES score thresholds and adjusted p-values ≤ 0.05 for hits of the CELL-BASED and varying p-values and negative clustering scores for the RIPLEY method. We could not evaluate the AVERAGE method on the validation screening data because this approach was used to select genes for validation. Since no negatives were chosen for validation no true and false positives can be calculated and therefore, no sensitivity and specificity values.

Figure 3.17 visualizes the AUC values for CELL-BASED and RIPLEY over different increasing z-score thresholds on the validation data. The Figure clearly demonstrates that for each z-score value in the validation screen the area under the ROC curve value is much better for the CELL-BASED than for the RIPLEY method. Therefore, CELL-BASED method is again showing superior performance concerning sensitivity and specificity in comparison to RIPLEY.

3.7 Pathway Analysis and Functional Annotation of Host Dependency Factors

Using the CELL-BASED analysis approach we identified 54 HDFs for HCV and 57 HDFs for DENV. The DENV hit list is tabulated in Appendix B.2. For both screens we mapped the corresponding hits to KEGG [KG00] (see Section 4.1.3) and Biocarta [Bio11a] using the functional enrichment analysis

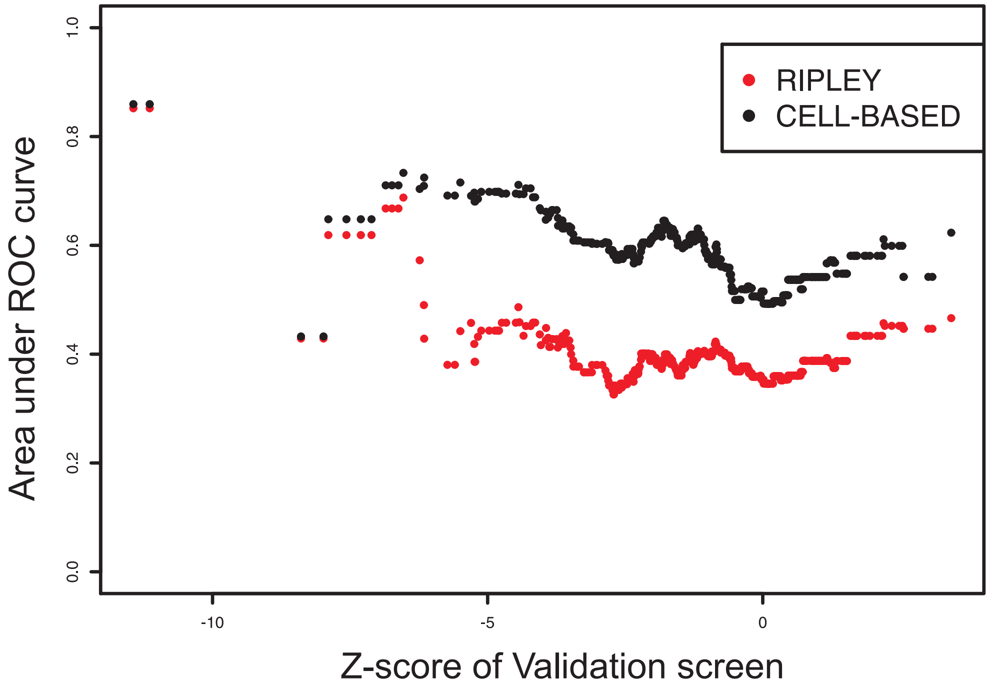


Figure 3.17: Area under the ROC curve values using CELL-BASED and RIPLEY approaches over different z-score thresholds on an HCV validation screen. Here, the intersection of predicted hits on the primary screen using CELL-BASED and RIPLEY approaches with genes screened in the validation screen is used to compute ROC curves and AUC values for varying *ES* values and clustering scores, respectively.

provided by DAVID [6.711] to select significantly enriched pathways. Results of both screens are shown in Appendix B.3.

We identified 29 pathways being involved in HCV using the hits of the CELL-BASED approach. Among them, 20 are significant with a p-value $\leq$ 0.05. In contrast, using the hits identified with the RIPLEY method not a single pathway could be identified. Using the hits of the AVERAGE method only two pathways are found and both of them have p-values $\geq$ 0.05: Axon guidance (p-value = 0.061) and Purine metabolism (p-value = 0.082). Purine metabolism has also been identified with the CELL-BASED method (p-value = 0.04). Axon guidance plays a role in the formation of the neuronal network and thus, it does not seem to be directly linked to HCV signaling. Downstream signaling of this pathway, however, induces changes in cytoskeletal organization which has already been reported to be involved in HCV [RRB$^+$11]. The hit genes found in the Axon guidance pathway are *LIM domain kinase 2* (LIMK2), *Cofilin* (CFL) and *proto-oncogene tyrosine-protein kinase Met* (MET). LIMK2 and CFL play a role in the regulation of the actin cytoskeleton and MET in mechanisms like endocytosis or focal adhesion. All three processes have been identified using the hits derived with the CELL-BASED method.

Beside the already mentioned endocytosis, focal adhesion, and regulation of the actin cytoskeleton, enriched processes for HCV using the CELL-BASED hits include signaling in the immune system, and the ErbB and MAP kinase signaling. All have previously been reported to play a role in HCV signaling by Reiss *et al.* [RRB$^+$11] where they pooled their screen with other screens published earlier. Notably, using the CELL-BASED method presented here we identified the same pathways without the additional information of other screens. This indicates once more the high sensitivity of our approach. A further evidence of our results is, for example, the ErbB and the MAP kinase signaling pathways, which have been reported to be of importance for HCV and flaviviruses in general in 2009 [LBN$^+$09].
Several additional pathways are identified, including regulation purine metabolism, TLR signaling and several cancer-related pathways. To conclude, this clearly shows that our approach results not only in an increased sensitivity and specificity on an individual siRNA level, but also in an increased sensitivity on a pathway level.

Using the hits of the DENV screen, we identified in total 20 enriched pathways and 14 of them have a p-value $\leq$ 0.05. Enriched processes include again focal adhesion, immune signaling, and the ErbB and MAP kinase pathways. Thus, although the overlap of HCV and DENV HDFs is small at the gene level (7 genes) the enriched pathways are showing a significant overlap (13 pathways (65%)), reflecting the close evolutionary relationship of both viruses (compare Section 2.1.2).

3.8 Population Context of Non-Virus Screens

In viral RNAi screens, cells are not only transfected with a certain siRNA but also infected with a virus. To test whether the high influence of population context parameters on the phenotypic read-outs is true also for non-virus RNAi screens, we use an image-based screen of an innate immune signaling pathway using liposomal reagents.
Cells have been reverse transfected with siRNA on spots of LabTek chamber slides. The signaling of the pathway under investigation has been triggered by transfection of the cells with a defined stimulus. A few hours later, the pathway activation in individual cells has been assessed by microscopy of a fluorescent reporter.
Across the whole screen (about 2.4 Mio. cells have been analyzed) a strict correlation between population context and the rate of pathway activation has been detected. This is due to the vastly different susceptibility for liposomal transfection among cells growing in different micro-contexts. Especially the correlation between each cells local density is observable by eye-inspection. To quantify this, we calculate the explained standard deviation of the rate of pathway activation by the population context features for each plate. The mean and standard deviation across replicated plates of the individual population features are for Cell Size: 8.1 ± 2.57, Density: 9.24 ± 3.5, Cell Number: 8.8 ± 2.4, Cell Shape: 8.16 ± 2.75 and for Population Border: 8.9 ± 8.3. Thus, the population context features show high effects on the phenotype and thus, they have to be normalized to perform hit-calling.

Moreover, technical features average to 6.1 ± 1.9 (Row), 4.4 ± 2.1 (Column) and 4.5 ± 2.71 (Spot Border). Analysis has been done on individual plate level and thus, for the feature addressing the plate effect (Overall) no binning has been performed. We calculate AUC values for the positive and negative controls given in the screen after normalizing against the population context and against the technical features and applying our approach for statistical hit scoring. We received increased performance (AUC=0.66) in comparison to an analysis without normalizing against the features (AUC=0.58).

3.9 Discussion

In this Chapter we introduced a new approach for the analysis and statistical processing of high-content, high-throughput microscopic screens. We used individual cell measurements to remove the observed phenotypic effects on viral infection and replication. Individual cell fluorescence intensities have been normalized against population context and technical artifacts, to reveal the true biological signal. We furthermore developed a statistical testing procedure which uses a modified version of functional enrichment analysis to assign each knockdown with a significance value. Based on two large-scale RNAi virus infection screens we showed superior performance concerning sensitivity and specificity of our method in comparison to two other approaches recently published.

An evaluation based on individual cell measurements can exploit the information contained in hundreds to thousands of cells in one spot. Thereby, the biological variability of cells being in different states is taken into account. Obviously, the cells within one spot are treated in the same way and are not technical replicates. Thus, they are not independent from each other. Nevertheless, we identified two clearly separated distributions of cells within one spot. This shows that there are phenotypic differences of individual cells even if they are treated in the same way. Our results show that the integration of multidimensional phenotypes from high-content screens can make data analysis and hit scoring much more specific. Taking the individual cell measurements within each spot into account highly improves sensitivity and specificity values. The number of false positives and false negatives on single spot level is limited to a minimum resulting in an almost perfect classification.

Previous infection screens targeting the same virus showed a very low overlap of identified hits [Che09, Gof08]. Although this is improved when considering overlaps at pathway level, there is still a surprisingly high variability in results. This finding has been also confirmed in our comparative analysis using the AVERAGE, RIPLEY and CELL-BASED approaches, showing significant differences in resulting hit lists of the same HCV screen. Since we reported large influences of the cell population context on viral infection, and this has been also described previously by Snijder and co-authors [SSR$^+$09], we conclude that the population factors contribute at least partially to this problem.
Looking at the enriched pathways identified with the hit lists using the AVERAGE and CELL-BASED approach, our results clearly show an increased sensitivity. We found substantially more pathways than when using the AVERAGE method (with the RIPLEY approach no pathways at all could be

identified). Several of the discovered pathways have already been associated with HCV and some are newly discovered.

Whereas Snijder *et al.* reported the location of a cell at the edge or in the middle of a local cell cluster is the main contributing factor influencing DENV infection, our results indicate that the size of the cell is the most important cell context feature for DENV as well as for HCV. The difference may be due to different experimental conditions like using different cell-lines (Huh7.5 versus HeLa), different platforms (LabTeks versus well-plates) and RNAi data versus cellular data without any siRNA transfections.

Our results on a non-viral screen strongly indicate that the population context not only influences RNAi infection screens. However, virus screens are more complicated since the infection itself can induce virus phenotypic effects. These cytopathic effects may directly influence a population context feature. An infection may lead for example to larger cell sizes of infected cells. Normalization against the cell size would then destroy the effects of the perturbations. Since in the HCV and DENV there are no control spots without infections, we cannot test whether the cells in our screens suffer from cytopathic effects. However, the analysis with GSEA-ONLY, where we do not normalize against cell population effects, results in similar sensitivity and specificity values on controls, with the CELL-BASED method being even slightly better. This, as well as the increased sensitivity of the CELL-BASED analysis on pathway level, indicates that we do not destroy effects when normalizing against cell context features, but allow a more sophisticated and improved analysis.
To conclude, high-content screening offers a powerful tool to further elucidate virus host interactions in the future, with significant advantages over high-throughput screens with low-dimensional, non-microscopy based readouts.

Part II

Network Inference

Chapter 4

Networks in Biology

This Chapter is part of the second part of this thesis where we describe how RNAi data is used to infer signaling networks. RNAi experiments and subsequent data processing allow the identification of genes related to a specific phenotype. However, their spatial and temporal ordering within the biological processes remain unknown. To understand the behavior of biological systems in more detail, it is necessary to elucidate how individual genes interact with each other. Graphs are an often used tool for the analysis and modeling of such interactions. Therefore, we start this Chapter with giving a general background on graphs in Section 4.1. Then, Section 4.2 discusses the challenge of inferring signaling networks from high-throughput gene perturbation data and already published network inference strategies are explained. Furthermore, we outline the advantages and drawbacks of each method.

4.1 Background

4.1.1 Graphs

Graphs provide an easy way to visualize and interpret the structure of related entities. A graph G consists of a set of *nodes* V and a set of *edges* E that connect the nodes, so *graph* $G = (V, E)$. The edges can be *directed* (with a *parent* and a *child* node) or *undirected* . Additionally, they can be allocated with certain attributes, for instance numbers indicating the strength of the relationship between the two nodes they connect. This is called edge weight w_{ij} of the edge between node i and j with $i, j \in E$. If the graph is binary, edges are either present ($w_{ij} = 1$) or absent ($w_{ij} = 0$).

A *self-loop* is an edge which connects a node with itself and a *cycle* allows a *walk* from node v_0 via an alternating sequence of edges and nodes back to node v_0. The *degree* of node v is equal the number of edges connected to it. For directed graphs the *in-degree*, respectively the *out-degree* is defined

to distinguish between the number of incoming, respectively, the number of outgoing edges of a node.
A graph is *connected* if there is a walk between each pair of nodes, otherwise it is *disconnected*. Furthermore, directed graphs can have *root* nodes (nodes which have only outgoing edges) and *leaves* (nodes which have only incoming edges).
There are several special types of graphs for example *directed acyclic graphs* (DAGs) which are directed and without cycles or *transitively closed* graphs where for each pair of nodes which are connected by a directed path a directed edge exists between them. So if there is a path from node i to node j then there is also an edge from i to j for all $i, j \in E$ of graph $G = (V, E)$. For more details about graphs see for example [GCD$^+$05].

4.1.2 Bayesian Networks

Bayesian networks are often used to infer signaling networks [MBS05, Mar06, FSA$^+$09, KDZ$^+$09], see also Section 4.2.3.
Bayesian networks are DAGs whose nodes represent random variables and whose edges the conditional dependencies. Unconnected nodes represent variables which are conditionally not dependent on each other. A probability function for each node defines the probability of its variable. The function takes as input the values of the variables of the parent nodes.

Formally speaking, let $G = (V, E)$ be a DAG with $n \in \mathbb{N}$ nodes, and X be a set of random variables assigned to each of them. Then, X is a Bayesian network with respect to G if its joint probability density function is given as the product of the individual density functions, which are conditional to their parent variables:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i | pa(X_i)), \tag{4.1}$$

where $pa(X_i)$ is the set of parents of node $i \in \{1, ..., n\}$. If a node has no parent nodes, its probability distribution is an independent distribution. For details about Bayesian network learning see for instance [NBBW07, Hec96].

4.1.3 Graphs in Biology

In biology, graphs provide an easy tool to understand, represent and model data. There are various different applications of graphs in biology [GCD$^+$05, ZZC08], however, we focus here only on two of them, namely Gene Ontology and biomolecular pathways.

Gene Ontology Graphs

The Gene Ontology (GO) [Ash00] project is a bioinformatics initiative which addresses the need of a standard representation of genes and their products across different species and databases. The GO Consortium includes gene product annotation data and enables its fast access and easy processing.
Furthermore, it provides a structured vocabulary of terms (which are identified with an unique GO label) for describing gene products according to three different ontologies:

- Molecular function:
  Defines the elemental activities of a gene product at the molecular level.

- Biological processes:
  Molecular events with a defined beginning and ending.

- Cellular component:
  Part of the cell or its extracellular environment where a gene is acting.

The vocabulary terms are represented as nodes arranged in a DAG and the relationships between them are defined as directed edges. The nodes are arranged hierarchically, which means child terms being more specific than its parents. Moreover, each edge is categorized by one of different types of relations such as *"is a"*, *"part of"* or *"regulates"*.

Biomolecular Pathways

Various different models of molecular networks have been constructed so far. Examples are protein-protein interaction (PPI) networks, gene co-expression networks or signaling networks [Alb05].
In PPI networks nodes represent proteins and their interactions are modeled by undirected edges. Most often, edges are further specified for instance by confidence scores about the reliability of the interaction. Large PPI networks offer an expanded insight into the organizational principles and topological properties. An example for this is network centrality, which is used to find hub-genes. These genes are highly connected and essential for certain processes [SWL$^+$05, WWW$^+$07].
Gene co-expression networks model highly correlated genes [BTS$^+$00, CBGB04, ZH05]. Nodes correspond to genes, and edges are significant pairwise correlations of gene expressions measured for instance with DNA microarrays. It is assumed that correlated genes are co-expressed and that co-expression is a result of co-regulation and co-functioning [AKB04]. Therefore, co-expression networks allow to draw hypotheses of the function of yet unknown genes. Furthermore, they offer the possibility to compare processes across different species

or different cell lines. For this appropriate tools to query the networks to find cluster or sub-networks of conserved genes are given [ZZC08].
Signaling networks describe the interplay of different molecules to coordinate biological activities within a cell. These activities can describe for example gene regulations, responses to external stimuli or enzymatically catalyzed reactions. To store, publish and furthermore work with the different types of networks in a formal, ontology-based manner, they are integrated into databases. KEGG (Kyoto Encyclopedia of Genes and Genomes) [KG00] is one example of a public database resource. It consists of 16 main databases such as the "KEGG PATHWAY" database. Biological systems within KEGG are represented via graphs where the set of nodes are KEGG objects (database entries) and the edges are biological relationships. Similar to GO, each object is specified using an unique identifier. KEGG PATHWAY is a collection of manually drawn molecular interaction/reaction graphs of metabolism and other cellular processes arranged hierarchically. Various signaling networks are listed for a wide range of biological phenomena, such as cell growth, apoptosis or differentiation.

4.2 Networks and RNAi Data

While RNAi is a powerful tool to identify genes involved in a specific biological process, the network inference out of RNAi data is a challenging task [MS06]. One of the problems is that the dimensionality of the network inference is increasing exponentially with the number of nodes. For a directed graph of $n \in \mathbb{N}$ nodes there are $2^{n(n-1)}$ possible undirected network topologies without self-activating edges [HP73]. Thus, a complete enumeration of the solution space is not possible for large problems.

The combination of the RNAi technology with mRNA or protein expression measurements is an often used strategy to study the effect of individual gene knockdowns on other genes. Thereby, it is learned how genes interact with each other [FFS$^+$07, FSA$^+$09, MBS05].
Several authors use perturbation data in combination with database and literature knowledge to infer the network topology. Ourfali et al. [OSI$^+$07] formulated an integer programming approach to infer an integrated network of protein-protein and protein-DNA interactions. For this, they use effects of knockout data on gene expression levels and database information. A similar approach has been published by Lan and co-authors [LSR$^+$11]. They construct signaling and regulatory networks by linking genetic and transcriptomic screening data with data of known molecular interactions. However, both approaches are only possible if information of the respective proteins or genes are already given. In contrast to these methods, several approaches do not need a

pre-given interaction network and the signaling network is inferred purely from the data. In the following, we discuss some of these approaches in detail and explain whether they can deal with large-scale problems of high dimensionality or not.

4.2.1 Nested Effects Model

Markowetz and co-authors [MBS05] developed a method called Nested Effects Model (NEM) to infer networks from perturbation data. It is a computational framework based on the nested structure of affected downstream genes which scores network hypotheses in a Bayesian manner. Within their model they distinguish two kinds of genes: *S-genes* and *E-genes*. S-genes ("S" stands for "silenced" or "signaling") are the candidate pathway genes silenced by RNAi while the pathway is stimulated. The effects of the perturbed genes are measured with the E-genes ("E" stands for "effect") using DNA microarray gene expression data. E-genes are downstream genes which are not part of the model and considered only as reporters of the signal flow in the pathway. Genes which show a high expression change are classified to be E-genes.
Assuming that the E-genes are transcriptional phenotypes regulated by the S-genes on the second level, interventions to the S-genes interrupt the signal flow throughout the pathway and can be directly measured using the E-genes. Thus, given a candidate network topology of S- and E-genes and the position of interventions, its agreement to the downstream response of experimental data can be distinguished.

The work flow of NEMs starts with an adaptive data discretization:
The continuous microarray data is transformed to binary values using the mean of positive controls for distinguishing between signal is interrupted or not (corresponding to 1 or 0, respectively). Therefore, state 0 is naturally given for all genes in a stimulated, unperturbed pathway. If the signal flow cannot reach a node due to an intervention at some node upstream in the pathway, its state is assumed to become 1.
Let μ_i^+ and μ_i^- be the mean of the positive and the negative controls for E-gene E_i with $i \in \{1, ..., |E|\}$, then the binary data E_{ik} is defined as:

$$E_{ik} = \begin{cases} 1 & \text{if } C_{ik} < \kappa * \mu_i^+ + (1 - \kappa) * \mu_i^-, \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

where C_{ik} is the continuous expression level of E_i in experiment k ($k \in \{1, ..., n\}$, with ideally $n \geq 5$) and $\kappa \in [0, 1]$ a parameter to control the false negative rate. This results in a binary matrix $D = (e_{ik})$ with $e_{ik} = 1$ if E_i is showing an effect in experiment k.
The S-genes take the value 1 or 0 depending on whether the signaling has been

interrupted or not. The S-genes which are in state 1 after the perturbation of the S-gene S form the influence region of S. Furthermore, the set of all influence regions are summarized in a silencing scheme Φ. Assuming each E-gene has only one parent in S, its state is 1 if the parent gene is 1, and 0 otherwise.

The parameters $\Theta = \{\theta_i\}_{i=1}^m$ with $\theta_i \in \{1, \ldots, n\}$ and $m = |E|$ are used to show if E_i is attached to S_j by setting $\theta_i = j$. For computing the likelihood $P(D|\Phi, \Theta)$ the distribution of E_{ik} is determined by using the silencing scheme Φ and the error probabilities α and β. These error probabilities are estimated from the positive and negative controls, where type I error α and type II error β are the number of positive controls discretized to 1 and the number of negative controls discretized to 0, respectively.
Since the silencing scheme Φ and the model parameters Θ are not known, the authors make three assumptions:

1. Parameter independence:
$$P(\Theta|\Phi) = \prod_{i=1}^{m} P(\theta_i|\Phi). \tag{4.3}$$
2. Uniform prior:
$$P(\theta_i = j|\Phi) = \frac{1}{n} \text{ for all } i \text{ and } j. \tag{4.4}$$
3. The observations in D are sampled independently and identically distributed given the silencing scheme Φ and the model parameters Θ:
$$P(D|\Phi, \Theta) = \prod_{i=1}^{m} P(D_i|\Phi, \theta_i). \tag{4.5}$$

Using these assumptions the marginal likelihood $P(D|\Phi)$ of a silencing scheme can be calculated with:

$$P(D|\Phi) = \frac{1}{n^m} \prod_{i=1}^{m} \sum_{j=1}^{n} \prod_{k=1}^{l} P(e_{ik}|\Phi, \theta_i = j). \tag{4.6}$$

To evaluate how good a silencing scheme Φ fits the data, Bayes' formula is used to score the silencing schemes by computing the posterior probability with:

$$P(\Phi|D) = \frac{PD|\Phi)P(\Phi)}{P(D)}, \tag{4.7}$$

where $P(D)$ is a normalizing constant which is the same for all silencing schemes. The prior $P(\Phi)$ can be chosen to incorporate prior knowledge and the marginal likelihood is computed as shown above.

If a silencing scheme Φ is given, the posterior probability for an edge between S_j and E_i can be easily calculated with:

$$P(\theta_i = j|\Phi, D) = \frac{1}{Z} \prod_{k=1}^{l} P(e_{ik}|\Phi, \theta_i = j), \tag{4.8}$$

using a uniform prior for the E-gene position. Z is a normalizing constant chosen that all probabilities for E_i sum up to 1 over all S-genes.

Different topologies can result in the same downstream effects and therefore, they are summarized into the same silencing scheme. The schemes are sorted according their scores to find pathways which have maximum posterior probabilities and thus, which best represent the data.

The authors applied their method on simulated data and on real biological data studying the *Drosophila* immune response. The simulation studies show that correct topologies can be reconstructed more easily if there are more replicate measurements available. Having five replicates and low noise levels, more than 90% of the networks could be reconstructed.
For the *Drosophila* data, a top scoring silencing scheme was computed for four different S-genes, which fits perfectly to the assumed true topology. This shows, that NEMs can be used to model and infer networks based on gene expression data after RNAi gene perturbations.

However, there are several limitations of the model:

- The approach can be used only if the number of perturbations is much smaller than the number of measured downstream effects.
- Several topologies may explain the data equally well and no unique solution can be stated.
- Continuous expression data has to be transformed into binary values, which results in a loss of information.
- For high number of S-genes (more than five) heuristics have to be used to search the model space for the best-fitting topology.
- Only features of a pathway can be reconstructed from the indirect observations allowing only a rough recovery.
- No distinction between activating and deactivating edges is done.

Apart from these disadvantages, the method which is presented in this thesis is designed for another scenario. Instead of having only a small number of perturbations which monitor indirect, high-dimensional downstream effects, we aim at inferring networks from many interventions. These interventions can come from single, double or even multiple knockdowns. Whereas in NEMs each node correspond to one perturbation experiment, we consider each node to be a single protein which may be influenced by others. In addition, we want to learn activating as well as deactivating edges, and the data is allowed to have missing values as well as nodes without any measurements.

Improvements of NEMs

There exist several improvements and extensions of the NEM approach:

- The original version of NEMs [MBS05] has been improved two years later by Markowetz *et al.* [MKTS07] which makes the method feasible also for larger number of perturbed genes. The authors use a divide-and-conquer approach to infer the topology of all genes by constructing it from sub-models which consist of only pairs or triples of genes. The authors show on simulated data that they can accurately reconstruct network models by using smaller sub-models. Furthermore, using two real data sets studying the response to microbial challenge in *Drosophila melanogaster* and the compendium of expression profiles of *Saccharomyces cerevisiae*, the authors show that their results reflect the functions of the involved genes.
  This approach is developed only for binary effects and it cannot distinguish between activation and deactivation.

- In 2007, Froehlich *et al.* [FFS+07] proposed three further extensions of NEMs originally introduced by Markowetz *et al.* [MBS05]. First, they show how to omit the data discretization step by using p-values which define the likelihood of an E-gene being differentially expressed after a certain perturbation.
  Furthermore, the basic assumption of the model prior $P(\Phi)$ being uniform over all possible models is enhanced. This allows the integration of prior knowledge. The scoring scheme introduced by Froehlich *et al.* incorporates prior assumptions for each individual edge. In order to avoid overfitting by simply believing in the data and biasing network scores towards the prior, regularization techniques like the Akaike information criterion (AIC) [HTF01] are used.
  The last, and most important, enhancement makes the method also applicable to high-order networks. For this, the authors present two different approaches: a stochastic sampling approach called simulated

annealing (SA) [KGV83] and a divide-and-conquer method called *module networks*. In the module network approach complete networks are recursively reconstructed from smaller sub-networks (*modules*) similar to the improvement by Markowetz *et al.* proposed in 2007 [MKTS07] discussed above.
Froehlich and colleagues applied their method on artificial data as well as on the data already used in [MBS05] where they showed identical results. Furthermore, they used data from the Human ER-α pathway to show performance of their approach on a higher scaled inference problem of 13 genes. The learned network topology was in good agreement with literature knowledge.

- Tresch and partners extended in 2008 [TM08] the original definition of NEMs in four ways. First, they use a different likelihood function of a NEM which makes it applicable not only for binary data, but also for p-values or other statistics which can be converted into a likelihood ratio. Second, Tresch *et al.* show that this new likelihood formulation allows to efficiently traverse the model space. Third, the authors show that the maximum likelihood estimator recovers the true structure of the graph if the data has a sufficient number of replicate measurements and thus, that under mild conditions the model is identifiable. Fourth, it is shown how prior knowledge can be incorporated and how measurement noise can be decreased by feature selection and regularization.
  The original NEM version [MBS05] can learn only transitively closed graphs. In the formulation given by Tresch and colleagues this is expanded to all directed graphs which reduces model bias. Based on the data studying the *Drosophila* immune response, which has also been used in [MBS05], the expanded version leads to results which are closer to existing biological knowledge than in the original publication. In addition, simulation studies are used to show that the version published by Tresch *et al.* reliably reconstructs interaction graphs.

- Anchang *et al.* [ASJ$^+$09] proposed in 2009 a statistical method called Dynamic Nested Effects Model (D-NEM) . This enhances the approach of NEMs by taking temporal gene expression data into account. Furthermore, it allows the modeling of the temporal interplay in the cell signaling, and gene expression after observed time delays. The authors could identify a feed-forward loop dominated network important during embryonic stem cell self-renewal, showing high biological significance.

- Recently, Froehlich and colleagues [FPT11] presented “dynoNEMs”. This is an extension of NEMs which enable the analysis of perturbation time series data. It is complementing the attempt of the D-NEMs

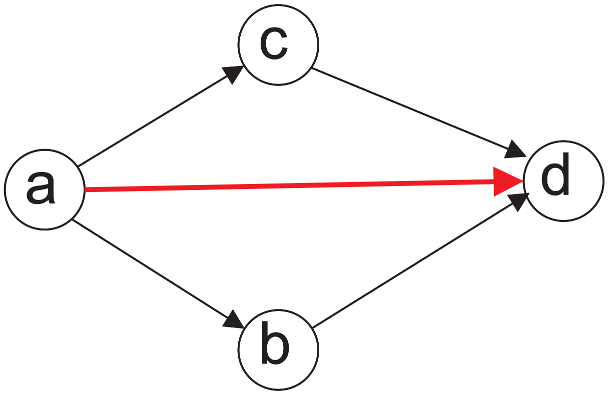


Figure 4.1: Example network topology with four nodes. The existence of the red edge can be inferred only if there exists a double knockdown of nodes b and c. See text for details.

> and allows for the resolution of feedback loops as well as for the discrimination of direct and indirect signaling. In dynoNEMs the signal flow is unrolled over time. The authors applied their method on the same dataset like Anchang and colleagues which studies the molecular mechanisms of self-renewal in murine embryonic stem cells. Their results were in a good agreement to the results of D-NEMs and with biological literature.

Beside the drawbacks already mentioned, the improvements and extensions of the NEMs discussed above have still a severe limitation. They do not use perturbation data of multiple genes at the same time. This kind of data, however, can be essentially to distinguish between some topologies without using time-resolved data. Assume for example a network of four nodes as given in Figure 4.1. It can be seen immediately, that the existence of the red edge from node a to c can be learned only if the double knockdown of the nodes b and c or time-resolved data is given.

Although it is generally possible to include double and multiple knockdowns into NEMs, it has not been implemented so far. Instead, they have been extended to D-NEMs and dynoNEMs which use time-resolved data. This makes them not usable for our scenario: we aim at inferring networks purely from steady-state data since time-course data are not always available. In addition, we want to reconstruct a network without the measurement of additional E-genes. Furthermore, the approach shall be able to deal with large-scale problems with twenty, thirty or more nodes, and to learn activating as well as deactivating interactions.

4.2.2 A Model using Probabilistic Boolean Threshold Networks

Kaderali *et al.* presented in 2009 a Bayesian network approach for inferring networks out of gene knockdown data [KDZ+09]. Nodes are modeled as Boolean random variables which can take two states: "on" or "off". Assuming discrete time steps t, the probabilities of the states are calculated as sigmoid functions. These functions take the weighted sum of regulations from the parent nodes at the previous time point as input:

$$p\{x_i(t) = 1|x(t-1)\} = \frac{1}{1 + exp(-\lambda(w_i^0 + \sum_{j=1}^{n} w_{j,i}x_j))}, \tag{4.9}$$

where the Boolean random variable $x_i \in \{0,1\}, i \in \{1,\ldots,n\}$ corresponds to a protein which can be activated or inactivated (1 and 0, respectively). The model parameter $w_{j,i} \in \mathbb{R}$ represents the strength of the effect of protein j on protein i. The strength is zero if there is no effect and greater, respectively, smaller than zero if there is an activation, respectively, a deactivation. The parameters $w_i^0 \in \mathbb{R}$ model the behavior of a protein i if no other proteins regulate it. The stochasticity of the network with given w and w^0 is controlled by parameter λ.

The more parent nodes are activated, the higher is the probability of the child node to be activated. Activation of a node can result in both, enhancing and inhibiting effects on child nodes, depending on the sign of the regulation weight. Obviously, the probability of a protein i being inactive at time point t is given by:

$$p\{x_i(t) = 0|x(t-1)\} = 1 - p\{x_i(t) = 1|x(t-1)\}. \tag{4.10}$$

If parameters w and w^0 are known, a state transition probability matrix $M \in \mathbb{R}^{2^n \times 2^n}$ is defined with

$$\begin{aligned} M_{i,j} &= p\left\{x(t) = \eta^{(j)}|x(t-1) = \eta^{(i)}\right\} && (4.11)\\ &= \prod_{k=1}^{n} p\left\{x_k(t) = active(k,\eta^{(j)})|x(t-1) = \eta^{(i)}\right\}, && (4.12)\end{aligned}$$

where $x(t) = \eta^{(i)}$ indicates that the system is in state $\eta^{(i)}$ at time t and $active(k,\eta^{(i)})$ is an indicator function which is 1 if $x_k(t)$ is active in state $\eta^{(i)}$ and 0 if it is inactive.

Assume an initial distribution $p_0 = p(\eta(0))$ at time 0 and M to be given. The probability distribution $p(\eta(T)|M,p_0)$ over the states of the system at time $T > 0$ can be computed with:

$$p(\eta(T)|M,p_0) = p_0' M^T \delta(\eta(T)), \tag{4.13}$$

with p'_0 being the vector transpose of the column vector p_0 of the initial state distribution and $\delta(\eta) \in \mathbb{R}^{2^n}$ is defined with $\delta_i(\eta) = 1$ if $\eta = j$ and $\delta_i(\eta) = 0$ otherwise.

In terms of RNAi data, the silencing of a protein k results in the deactivation of the corresponding node in the network. Then, x_k is fixed to be 0 and an updated matrix M^{-k} can be computed by removing all rows corresponding to states where k is active and by marginalizing over the corresponding columns. Given the information of systematic single or multiple gene knockdowns and initial state distributions, the network topology which fits the data most likely can be determined using a Maximum likelihood approach:

$$p\left\{D|w, w^0, T\right\} = \prod_{k=1}^{K} p\left(\eta^{(k)}(T)|M^{-k}, p_0\right), \tag{4.14}$$

where $\left\{\eta^{(k)}(T)\right\}_{k=1}^{K}$ are the observed states of the system with $\eta^{(k)}(T)$ being the observed state of the system at time T after knockdown k.

The authors use Bayes' theorem to infer the model parameters given the observed data by computing the posterior distribution:

$$p\{w, w^0, T|D\} = \frac{p\{D|w, w^0, T\}\pi(w, w^0, T)}{p(D)}, \tag{4.15}$$

where $p\{D|w, w^0, T\}$ is the likelihood, $\pi(w, w^0, T)$ is a prior distribution on the model parameters and $p(D)$is a normalizing factor. Mode hopping Markov chain Monte Carlo is used to evaluate the posterior distribution over the model parameters. For small networks the exact likelihood is computed whereas for larger networks it is approximated using stochastic simulation. The prior distribution on model parameters is calculated using the L_q norm. This forces the network topology to be sparse, which is commonly assumed to be true for biological networks. For more details see [KDZ$^+$09].

Evaluation of the proposed method has been done on simulated data for five genes. The authors could identify the true as well as an alternative topology even for parameter perturbations of up to 50%. In addition, the Janus Kinases and Signal Transducers and Activators of Transcription (JAK/STAT) pathway has been used to assess the performance on a real biological data set. Kaderali and partners were able to correctly reconstruct the core topology of the JAK/STAT pathway as given in [Pla05].

Similar to NEMs (Section 4.2.1) one of the disadvantage of this approach is that data needs to be discretized to distinguish whether a node is activated or not. Moreover, this approach suffers from a high run time and computational complexity. Therefore, it is only feasible for small-scaled networks even if stochastic simulations are used for likelihood computations, which makes the method not useful for the network inference on large-scale data, which is the objective in the work presented in this thesis.

4.2.3 Deterministic Effects Propagation Networks

Deterministic Effects Propagation Networks (DEPNs) are a special Bayesian network approach which allows the network inference from multiple intervention data. The method was introduced by Froehlich and colleagues in 2009 [FSA+09]. Effects of perturbations of one or multiple genes at the same time are studied by using protein expression level measurements from Reverse Phase Protein Arrays (RPPAs) [TQL+06].

For the network inference three things are necessary: First, each protein of interest is perturbed at least once. Second, the intervention effects on all other proteins are directly measured, and third, there is one experiment where no protein has been perturbed. If these conditions are fulfilled, the most likely network topology can be reconstructed.

Even if there are latent nodes (correspond to proteins where no measurements are available, but which have been perturbed), possible network topologies can be inferred. Furthermore, the approach of DEPNs can deal with missing data points.

In DEPNs, each protein corresponds to a node in the network graph Φ with two possible states: perturbed and unperturbed. Measured effects $x_i^{(t,p)}$ of interventions p on individual nodes i after time point or experimental condition t are modeled as two Gaussian distributions (perturbed/unperturbed) with unknown mean and variance:

$$x_i^{(t,p)}|t, pa(i), p \sim \begin{cases} \mathcal{N}(\mu_i^t, \sigma_i^{t^2}) & p \subseteq pa(i) \cup \{i\} \\ \mathcal{N}(\widetilde{\mu}_i^t, \widetilde{\sigma}_i^{t^2}) & \text{otherwise,} \end{cases} \tag{4.16}$$

where $p(a)$ corresponds to the set of parents of node i.

For each perturbation experiment the effects are calculated by defining a node i to be perturbed if itself or its parents are perturbed. Thus, perturbations are assumed to be deterministically propagated from top to bottom through the network, where always transitively closed graphs are learned.

Furthermore, each protein measurement can be classified in being perturbed or unperturbed, based on the control experiment with no gene perturbations. And finally, the parameters $(\mu_i^t, \sigma_i^{t^2})$ and $(\widetilde{\mu}_i^t, \widetilde{\sigma}_i^{t^2})$ of the two distributions (perturbed and unperturbed) can be estimated, based on the measurement classifications if the network structure Φ is known, in two ways:

1. Maximum likelihood estimate: $\mu_i^t = m_i^t$ and $\sigma_i^{t^2} = s_i^{t^2}$ where m_i^t and s_i^t are the empirical mean and standard deviation, respectively.

2. Bayesian estimate: it is supposed that $\mu_i^t | \sigma_i^{t^2} \sim \mathcal{N}(\mu_0, \sigma_i^{t^2}/\lambda_0)$ and $\sigma_i^{t^2} \sim$ Inv-$\chi^2(\alpha_0, \beta_0)$ where μ_0, λ_0, α_0 and β_0 are chosen in dependency of the perturbation state. The marginal posterior distributions for μ_i^t and $\sigma_i^{t^2}$ are calculated in analytical form:

$$\mu_i^t | x_i^{(t,p)} \propto t_{\alpha_n}(\mu_n, \beta_n/\lambda_n) \tag{4.17}$$

$$\sigma_i^{t^2} | x_i^{(t,p)} \sim \text{Inv-}\chi^2(\alpha_n, \beta_n), \tag{4.18}$$

   where $t_{\alpha_n}(\mu_n, \beta_n/\lambda_n)$ denotes the Student-t distribution with α_n degrees of freedom, location μ_n and scale β_n/λ_n. The remaining parameters are given by:

$$\mu_n = \frac{\lambda_0}{\lambda_0 + n_i}\mu_0 + \frac{n_i}{\lambda_0 + n_i} m_i \tag{4.19}$$

$$\lambda_n = \lambda_0 + n_i \tag{4.20}$$

$$\alpha_n = \alpha_0 + n_i \tag{4.21}$$

$$\beta_n = \frac{1}{\alpha_n}\left(\alpha_0\beta_0 + (n_i - 1)s_i^2 + \frac{\lambda_0 n_i}{\lambda_0 + n_i}(m_i - \mu_0)^2\right), \tag{4.22}$$

   with n_i being the number of observations which are used to compute the conditional density for $x_i^{(t,p)}$. Furthermore, Inv-χ^2 is the scale inverse of the χ^2 distribution, and the posterior modes of $\mu_i^t | x_i^{(t,p)}$ are μ_n and $\frac{\alpha_n}{\alpha_n + 2}\beta_n$.

If a certain network topology is given, the above calculated parameters are used to determine how likely it is that this is the true topology. Assume a data set $D = \{x_i^{(t,p)}\}$ with measurements of all proteins at all time points T under perturbations P, and the vector of all estimated posterior mode parameters Θ to be given. Then, the likelihood of a network hypothesis is calculated as:

$$p(D|\Phi, \Theta, P) = \prod_{t=1}^{T} \prod_{i=1}^{n} \prod_{p \in P} \prod_{j=1}^{r} p(x_i^{(t,p)} | t, pa(i), \Theta_i, p), \tag{4.23}$$

where n is the number of nodes and r the number of replicate measurements.

Froehlich *et al.* employed the method on simulated as well as on real biological data studying the ErbB receptor-regulated G1/S transition in HCC1954 human breast cancer cells [SFL$^+$09]. They showed that their method can correctly reconstruct most interactions like given in the literature.

Although the DEPNs do currently not distinguish between activating and inactivating edges, they are designed for network inference problems similar to those used in this thesis. They can deal with single as well as with multiple knockdowns, they do not require time-resolved data and they can be applied to problems which have up to dozens of nodes.

Dynamic Deterministic Effects Propagation Networks

Just recently, Bender and co-authors published an advanced DEPNs approach called dynamic deterministic effects propagation networks (DDEPN) [BHF$^+$10]. DDEPNs use time-resolved data to model the time-dependent behavior of biological systems explicitly. The method seems to be promising on simulated data with sensitivity and specificity values larger than random guessing as well as on real breast cancer data studying ErbB signaling where it predicts several already known signaling cascades. Nevertheless, it is designed for a different scenario than for the model developed in this thesis since it needs time-resolved data.

Chapter 5

Optimization Problems and Solvers

The model to infer signaling networks from RNAi data introduced in this thesis is formulated as a linear optimization problem. Therefore, we introduce optimization problems and their solvers in this Chapter. We note, that the Chapter is mainly based on [Hau03] and addresses the interested reader only. Those who are already familiar with optimization problems and solvers, we advice to directly go on to Chapter 6.

In the first Section of this Chapter, optimization problems are introduced in general. Then, linear programming and the simplex algorithm which is used to solve linear optimization problems are explained in more detail.

5.1 Optimization Problems

An optimization problem consists of:

1. The set L of feasible solutions, described by several equalities or inequalities.
2. A real-valued objective function z which is minimized or maximized in L.

That means, we are looking for:

$$x \in L : z(x) \leq z(y) \text{ or } z(x) \geq z(y) \; \forall \; y \in L, \tag{5.1}$$

for minimization or maximization problems, respectively. In linear Optimization, the objective function is a linear function and L is described with linear equalities and inequalities. The solutions of linear optimization problems

are in most cases non-integers. Linear optimization problems are solvable in polynomial time [Sch99].
Integer linear programming means linear optimization with the restriction that the solution has to be integer. Integer linear programming is NP-hard [GJ79, Sch99, Hau03].

5.2 Linear Optimization Problems

The idea of linear programming (LP) has been first developed in the 1820's by Fourier. Later, in the 1940's, its fundamental importance and usage was shown by the work of Dantzig, Kantorovich, Koopmans, and von Neumann (see [Sch99]).

Standard Form

LPs aim at the optimization of the objective function given by z. In general every linear problem can be expressed in a standard form:

Definition 5.1*:*
Assume $A \in \mathbb{R}^{m \times n}$, $b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$, $c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \in \mathbb{R}^n$ are given, then a *standard-maximum-program* aims at finding $x \in \mathbb{R}^n$ which

$$\begin{aligned} \text{maximizes } z(x) &:= c^T x \\ \text{s.t. } Ax &\leq b \\ x &\geq 0. \end{aligned} \tag{5.2}$$

An $x^* \in \mathbb{R}^n$ is *feasible* for an LP if it satisfies Equation 5.2 and $x^* \in \mathbb{R}^n$ is *optimal* if it is feasible and optimizes the objective function z over feasible x.

Remark 5.1*:*
A standard-minimum-program is equivalent to a standard-maximum-program, since $c^T x$ maximal $\Leftrightarrow -c^T x$ minimal.

Remark 5.2*:*
Every linear optimization problem can be formulated in the standard form using the following transformations:

1. Inversion of an inequality by multiplication with -1.

2. Replacement of an equation $f(x_1, \dots, x_n) = d$ by the two inequalities $f(x_1, \dots, x_n) \geq d$ and $f(x_1, \dots, x_n) \leq d$.

3. Conversion of strict inequality restrictions "less than" or "greater than", i.e. $f(x_1, \dots, x_n) < d$ or $f(x_1, \dots, x_n) > d$, respectively, to "less or equal to" or "greater or equal to" by adding or subtracting extra non-negative variables (*slack variables* ξ), i.e. $f(x_1, \dots, x_n) + \xi \leq d$ or $f(x_1, \dots, x_n) - \xi \geq d$, respectively. The cost of slack variables is zero in the appropriate position in the linear program objective function (i.e. $z(x) = c^T x + 0\xi$).

4. Replacement of non-restricted variables (thus, neither $x_j \leq 0$ nor $x_j \geq 0$ is required) by $x_j = x'_j - x''_j$ with the restrictions $x'_j \geq 0$ and $x''_j \geq 0$.

Canonical Form

Definition 5.2*:*
Given is $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. A *canonical maximum program* searches an $x \in \mathbb{R}^n$ which

$$\begin{aligned} \text{maximizes } z(x) &:= c^T x \\ \text{s.t. } Ax &= b \\ x &\geq 0. \end{aligned} \tag{5.3}$$

Similar, a *canonical minimum program* is defined for an $x \in \mathbb{R}^n$ which

$$\begin{aligned} \text{minimizes } z(x) &:= c^T x \\ \text{s.t. } Ax &= b \\ x &\geq 0. \end{aligned} \tag{5.4}$$

Remark 5.3*:*
Each canonical maximum program can be transfered into a minimum program by exchanging c by $-c$.

Remark 5.4*:*
Each standard maximum program of the form given in Definition 5.1 can be converted into a canonical maximum program by introducing additional variables (slack-variables) $(\xi_1, ..., \xi_m) = \xi^T$:

$$\begin{aligned} \text{maximize } z(x) &:= c^T x + 0^T \xi \\ \text{s.t. } Ax + \xi &= b \\ x &\geq 0 \\ \xi &\geq 0. \end{aligned} \tag{5.5}$$

Analogously, each canonical maximum program like given in 5.3 can be transformed into an equivalent standard maximum program by defining two inequalities instead of each equality:

$$\begin{aligned} \text{maximize } z(x) &:= c^T x \\ \text{s.t. } Ax &\leq b \\ -Ax &\leq -b \\ x &\geq 0. \end{aligned} \tag{5.6}$$

Solution Space

To define the solution space in more detail, further definitions are required.

Definition 5.3*:*
$K \subseteq \mathbb{R}^n$ is called *convex*, if $\forall\ x_1$,$x_2 \in K$ and $\forall\ \lambda \in [0,1]$ the following holds:

$$\lambda x_1 + (1-\lambda)x_2 \in K. \tag{5.7}$$

Definition 5.4*:*
The *convex hull* of K is the smallest convex set containing K and is denoted with:

$$\begin{aligned} conv(K) = \{\lambda_1 x_1, \ldots, \lambda_t x_t | t \geq 1; x_1, \ldots, x_t \in K; \\ \lambda_1, \ldots, \lambda_t \geq 0; \lambda_1 + \cdots + \lambda_t = 1\}. \end{aligned} \tag{5.8}$$

Definition 5.5*:*
Let $conv(K)$ be the convex hull of K, then $p \in conv(K)$ is called an *edge* of K if $p = \lambda x_1 + (1-\lambda)x_2$ with $x_1, x_2 \in conv(K)$ and $\lambda = 1$ $(p = x_1)$ or $\lambda = 0$ $(p = x_2)$.

Remark 5.5*:*
The set of edges can be finite, infinite (i.e. if $K \subset \mathbb{R}^n$ is circular) or empty (i.e. if $K = \mathbb{R}^n$).

Definition 5.6*:*
Let L be the non-empty set of feasible solutions of the canonical optimization program

$$\begin{aligned} Ax = b,\ x &\geq 0 \\ z(x) = c^T x &\text{ minimal,} \end{aligned} \tag{5.9}$$

with A being a (m, n)-matrix with $m < n$ and $\text{rank}(A) = m$. Then, $A_B = \{a^k | k \in B\}$ is called the *set of column vectors belonging* to $x \in L$, if

$$\sum_{k \in B} a^k x = \sum_{j=1}^{n} a^j x = Ax = b, \tag{5.10}$$

where $B \subseteq \{1, \ldots, n\}$ is the set of indices k with $x_k > 0$ (so $x_j = 0$ for $j \in \{1, \ldots, n\} \setminus B$).

Theorem 5.1*:*
Let L be the set of feasible solutions of an LP and L_{opt} the optimal solutions. If L is non-empty, then L has a finite number of edges. There is an optimal solution if and only if there are feasible solutions and if the set of feasible solutions has an upper or lower bound for minimization or maximization problems, respectively.

Proof. See [Hau03] Chapter 2.5. □

Theorem 5.2*:*
Assume $x \in L$, then x is an edge of L if and only if the set of column vectors belonging to x are linearly independent.

Proof. See [Hau03] Chapter 2.5. □

Definition 5.7*:*
Given are $Ax = b$ and A_B as in Definition 5.6.

(a) If A_B is non-singular, then A_B is called a *basis* of A.

(b) If A_B is a basis, then the variables which belong to the columns of the basis are called *basis variables* and the others *non-basis variables*.

(c) The *basis solution* of $Ax = b$ is the distinct vector x_B, which basis variables are defined by $x_B = A_B^{-1} b$ and all non-basis variables are zero. The basis solution is therefore a feasible solution of $Ax = b$.

(d) A basis solution belonging to x with basis A_B is called *non-degenerated edge* if there are m entries unequal to zero (so $x_B = A_B^{-1} b \geq 0$), and *degenerated edge* otherwise. Non-degenerated edges have exactly one basis, which is exactly the set of column vectors.

Theorem 5.3*:*
A basic solution $x_B = A_B^{-1} b$ is a feasible solution of $Ax = b$ with at most m non-zero entries. If all elements of x_B are non-zero, then x is also an optimal solution of the respective LP.

Proof. See [Hau03] Chapter 2.5. □

5.2.1 LP-Solver: the Simplex Algorithm

The *simplex* method was first introduced by George Dantzig in 1947 and algebraically formulated in 1951 [Dan51]. Geometrically, the feasible region of $Ax = b$, $x \geq 0$ is a (possibly unbounded) convex polyhedron, whose dimension is limited by the number of variables. The extreme points are exactly the edges of the polyhedron. If a linear program in standard form has a minimum value on the feasible region, this is one of the edges. Thus, the optimization problem can be solved by going along the lines of the polyhedron and finding neighboring (more optimal) edges until the global optimal edge is found. Since the polyhedron is convex, a finite extreme point which is not optimizing the objective function is connected to an edge which has a better solution of the objective function. Otherwise, the objective function is unbounded. There are three cases of solutions:

- LP is not feasible, which means the polyhedron is empty.
- LP is unbounded, which means solutions can be infinitely high/small for maximization/minimization programs.
- There is exactly one, or infinitely many solutions which are all lying on a common line of the polyhedron.

In short, the simplex algorithm simply walks along the edges until the optimum or an unbounded edge is found. However, the number of edges can increase exponentially with the number of variables. Thus, the number of extreme points can be huge even for small linear programs. For more details see [Grö04, Hau03, Sch99].

More formally, assume a canonical minimization program as in Definition 5.4 to be given. The simplex consists of two phases:

1. Phase I:
   Find a starting extreme point (edge) x^0.
2. Phase II:
   Use the basic feasible solution x^0 as starting point. If it is optimal, the algorithm is finished. Otherwise, find $x^1 \in E(L)$ with $z(x^1) < z(x^0)$, or there is no optimal solution and the linear program is called *infeasible*.

Phase I

Finding an edge x^0 of L depends on the optimization problem. Two scenarios are possible:

1. Problem is defined as a standard maximization program:

$$\begin{aligned} Ax \leq b,\, x \geq 0 \\ z(x) = c^T x \text{ maximal.} \end{aligned} \tag{5.11}$$

   WLOG assume $b \geq 0$. Introduce slack variables $\xi = \xi_1, \ldots, \xi_m$ that

$$\begin{aligned} (A|I_m) \begin{pmatrix} x \\ \hline \xi \end{pmatrix} = b,\ \begin{pmatrix} x \\ \hline \xi \end{pmatrix} \geq 0 \\ (-c^T|0^T) \begin{pmatrix} x \\ \hline \xi \end{pmatrix} \text{ minimal,} \end{aligned} \tag{5.12}$$

   where I_m is the m-dimensional identity matrix. Then $x^0 = \left(\frac{0}{b}\right)$ is a feasible solution since the column vectors belonging to x^0 consist of I_m and thus are linearly independent. Therefore, x^0 is an edge of L.

2. The program is already in the form of a canonical minimization problem:

$$\begin{aligned} Ax = b,\, x \geq 0 \\ z(x) = c^T x \text{ minimal.} \end{aligned} \tag{5.13}$$

   WLOG assume $b \geq 0$. Then, solve

$$\begin{aligned} (A|I_m) \begin{pmatrix} x \\ \hline \xi \end{pmatrix} = b,\ \begin{pmatrix} x \\ \hline \xi \end{pmatrix} \geq 0 \\ 0x_1 + \cdots + 0x_n + \xi_1 + \cdots + \xi_m \text{ minimal,} \end{aligned} \tag{5.14}$$

   where I_m is the m-dimensional identity matrix. Then $x^0 = \left(\frac{0}{b}\right)$ is a feasible solution since the column vectors belonging to x^0 consist of I_m and thus, they are linearly independent. Therefore, x^0 is an edge of L. The objective function of 5.14 is in the set of feasible solutions restricted to zero and thus, there exists an optimal solution in the edge $\left(\frac{\tilde{x}}{\tilde{\xi}}\right)$ of L. If $\tilde{\xi} \neq 0$, the objective function of 5.14 is greater than zero and there is no possible solution for 5.13. If $\tilde{\xi} = 0$, then $\tilde{x} \in L$ and $\tilde{x}$ is an edge of the polyhedron of 5.13, since

$$b = (A|I_m) \begin{pmatrix} \tilde{x} \\ \hline \tilde{\xi} \end{pmatrix} = (A|I_m) \begin{pmatrix} \tilde{x} \\ \hline 0 \end{pmatrix} = A\tilde{x}, \tag{5.15}$$

   and $\left(\frac{\tilde{x}}{0}\right)$ is an edge of 5.14 with linearly independent columns of A.

For details see [Hau03].

Phase II

Assume $x^0 = (x_1^0, \dots, x_n^0)^T \in L$ being an edge. Let furthermore $B \subseteq \{1, \dots, n\}$ be the set of indices i with $x_i^0 > 0$ and $\bar{B} \supseteq B$, $|\bar{B}| = m$, with $\{a^s | s \in \bar{B}\}$ being a basis of x^0. Then,

$$a^j = \sum_{s \in \bar{B}} \tau_{sj} a^s, \; \forall j \in \{1, \dots, n\}, \tag{5.16}$$

with $\tau_{ss} = 1$ and $\tau_{sj} = 0$ if $s, j \in \bar{B}$ and $j \neq s$. For any $x \in L$, the following holds:

$$b = \sum_{s \in \bar{B}} x_s^0 a^s = \sum_{j=1}^{n} x_j a^j = \sum_{j=1}^{n} \left(\sum_{s \in \bar{B}} \tau_{sj} a^s \right) = \sum_{s \in \bar{B}} \left(\sum_{j=1}^{n} \tau_{sj} x_j \right) a^s. \tag{5.17}$$

Since a^s is a basis of x^0, a comparison of the coefficients is possible:

$$x_s^0 = \sum_{j=1}^{n} \tau_{sj} x_j = \sum_{j \notin \bar{B}} \tau_{sj} x_j + x_s, \tag{5.18}$$

for $s \in \bar{B}$. This can be reformulated with

$$x_s = x_s^0 - \sum_{j \notin \bar{B}} \tau_{sj} x_j. \tag{5.19}$$

This can be used to define the objective function with:

$$\begin{aligned} z(x) &= \sum_{j=1}^{n} c_j x_j \qquad (5.20) \\ &= \sum_{s \in \bar{B}} c_s x_s + \sum_{j \notin \bar{B}} c_j x_j \\ &= \sum_{s \in \bar{B}} c_s x_s^0 - \sum_{s \in \bar{B}} c_s \sum_{j \notin \bar{B}} \tau_{sj} x_j + \sum_{j \notin \bar{B}} c_j x_j \text{ (see Equation 5.19)} \\ &= \sum_{s \in \bar{B}} c_s x_s^0 - \sum_{j \notin \bar{B}} \left(\sum_{s \in \bar{B}} c_s \tau_{sj} - c_j \right) x_j \\ &= z(x^0) - \sum_{j \in \bar{B}} (d_j - c_j) x_j, \end{aligned}$$

where

$$d_j = \sum_{s \in \bar{B}} \tau_{sj} c_s, \text{ for } j \notin \bar{B}. \tag{5.21}$$

There are three possibilities:

1. $\forall\, j \notin \bar{B},\ d_j \leq c_j$, and thus, $\forall\, x \in L,\ z(x) \geq z(x^0)$.
   Thus, x^0 is an optimal solution.

2. $\exists\, i \notin \bar{B},\ d_i > c_i$ and $\forall\, s \in \bar{B},\ \tau_{si} \leq 0$. If all feasible solutions x^δ hold that $z(x^\delta) < z(x^0)$, then the objective function is not lower bounded. Thus, there is no optimal solution (for more details see [Hau03]).

3. $\exists\, i \notin \bar{B},\ d_i > c_i$ and $\forall\, k \in \bar{B},\ \tau_{ki} > 0$, then there exists $\delta = min\{x_l^0/\tau_{ri} | l \in \bar{B}, \tau_{li}\}$ where x^δ is a solution of $Ax = b$, $x^\delta \in L$ and $z(x^\delta) < z(x^0)$. Based on Theorem 5.1 the algorithm terminates after a finite number of steps with the optimal solution, or it terminates with the result that there is no optimal solution (solution space not bounded).

For more details see [Grö04, Hau03, Sch99].

Complexity

Like mentioned at the beginning of this Chapter, the number of edges of a polyhedron can increase exponentially with the number of variables and inequalities. Klee and Minty showed in 1972 that the worst-case complexity of the simplex algorithm is therefore also exponential ($O(2^n)$) [KM72]. In degenerated linear programs (there exist two bases with the same basic feasible solution), there can be the problem of cycles, where always the same edge is calculated. Thus, the algorithm cannot terminate. However, in 1955 Dantzig, Orden and Wolfe developed a generalized simplex method where cycling can be avoided (see [DOW55]).
Borgwardt was able to show in 1982 that the average run time of the simplex algorithm is only polynomial with performing $O(n^3 m^{\frac{1}{n-1}})$ many changes of the basis [Bor82]. In practice, the simplex algorithm has proven itself to be even more efficient than the *ellipsoid* method developed in 1979 by Leonid Khachiyan, which is like the *Karmarkar algorithm* [Kar84], a polynomial algorithm. However, both methods are not discussed here, for more details see for example [Sch99].

Chapter 6

A Linear Model for Network Inference

To learn the signaling network of genes it is necessary to find how they interact with each other. In terms of a network graph, this results in finding the weights of the edges. Since there can be more than one possible networks which best represent the data, the model presented here in detecting the simplest and thus, the minimal one.

We assume that the signal transduction within the network is given as an information flow. The flow begins from one or several *source nodes* S and it is then propagated down through the network until it reaches one or several *sink nodes* F. Thus, a protein a influences other proteins which are further down in the network topology i.e. b, if there exists a path from a to b.
After the knockdown of a certain gene, its child nodes is supposed to show a phenotypic effect. Therefore, we classify all genes in the study whether they are *active* or *inactive* after each knockdown. Furthermore, the model assumes that nodes are active, if the sum of incoming edge weights from parent nodes are higher than a pre-given threshold and inactive otherwise. Based on these assumptions, we formulate the network inference problem as an optimization problem which uses the observed data to find a network topology that minimizes the edge weights and fits the data best. It is flexible since additional constraints can be easily formulated and only little restrictions are imposed on the network structures such as the exclusion of self-regulating edges. Cycles are allowed.

This Chapter starts with a Section where the linear model for the inference of signaling networks from perturbation data is described. The second Section describes how the theoretical and empirical sensitivity, specificity and precision values are calculated.

6.1 Linear Programming Model

Given are $n \in \mathbb{N}$ different genes and $K \in \mathbb{N}$ different knockdowns of one or several genes at the same time. The model presented here can deal with data where each gene has been silenced at least once, so $K = n$, but also missing data with $K < n$ or data with additional double or multiple knockdowns ($K > n$) is possible. The Model is developed for the use with RNA interference data, however, all kinds of perturbations can be used as long as the phenotypic response of the remaining genes can be quantified in an *observation matrix*:

Definition 6.1*:*
An *observation matrix* X is defined as the measurements of genes $i \in \{1, \dots, n\}$ after knockdowns $k \in \{1, \dots, K\}$ with $x_{ik} \in \mathbb{R}_{\geq 0}$, $\forall \{i, k\}$ where:

$$x_{ik} \begin{cases} \geq \delta_i & \text{if gene } i \text{ is inactive after knockdown } k \\ < \delta_i & \text{otherwise,} \end{cases} \tag{6.1}$$

and δ_i is calculated from the data.

The δ_i are for example chosen as the mean of δ_{ik} for all k knockdowns of gene i. Another possibility is to use a reference value for each gene where no genes have been perturbed. We note that the δ_i can be different for different genes i.

Definition 6.2*:*
An *activation matrix* B is defined with $b_{ik} \in \{0, 1\}$ where

$$b_{ik} = \begin{cases} 0 & \text{if gene } i \text{ has been inactivated in knockdown } k \\ 1 & \text{otherwise.} \end{cases} \tag{6.2}$$

Double or even multiple knockdowns can be easily handled by setting the respective entries of B to zero.

The signaling network is represented as a weighted directed graph $G(V, W)$ where the vertex $v_i \in V$ represents a gene and the edge $w_{ij} \in W$ the connection of gene i to gene j. If i and j are connected $w_{ij} \neq 0$ and $w_{ij} = 0$ otherwise. If the edge weight $w_{ji} > 0$ node j activates i and if $w_{ji} < 0$ node j deactivates i. The larger $|w_{ij}|$ the higher the evidence that gene i influences gene j.

Remark 6.1*:*
The observations of all nodes are assumed to be in steady state, so $x_{ik}(t+1) = x_{ik}(t)$, $\forall \{i, k\}$. Moreover, there are no self-regulating edges, so $w_{ii} = 0$, $\forall i$.

The LP Model

Assume we are looking for a network topology which has at least one given source node S and one given sink node F (the case if no source and/or sink node are given is discussed in Section 7.2.2). The LP model is defined as:

Definition 6.3*:*

$$\min z^*(w_{ji}^+, w_{ji}^-, w_i^0, \xi_l) := \left(\sum_{i,j} (w_{ji}^+ + w_{ji}^-) + \sum_i w_i^0 + \frac{1}{\lambda} \sum_l \xi_l \right) \tag{6.3}$$

$$\begin{aligned} \text{s.t. if } x_{ik} \geq \delta_i \text{ and } b_{ik} = 1: &\quad w_i^0 + \sum_{j \neq i} (w_{ji}^+ - w_{ji}^-) x_{jk} \geq \delta_i && (6.4) \\ \text{if } x_{ik} < \delta_i \text{ and } b_{ik} = 1: &\quad w_i^0 + \sum_{j \neq i} (w_{ji}^+ - w_{ji}^-) x_{jk} \leq 0 + \xi_l && (6.5) \\ \text{if } i \in V \setminus S: &\quad \sum_{j \in V, j \neq i} (w_{ji}^+ + w_{ji}^-) \geq \delta_i && (6.6) \\ \text{if } i \in V \setminus F: &\quad \sum_{j \in V, j \neq i} (w_{ij}^+ + w_{ij}^-) \geq \delta_i, && (6.7) \end{aligned}$$

where the objective function z in equation 6.3 is minimized, to keep the network sparse with most edge weights being zero. This minimization is done over three terms: the first term accounts for the absolute edge weights $w_{ji} = w_{ji}^+ + w_{ji}^-$ with $w_{ji}^+, w_{ji}^- \in \mathbb{R}_{\geq 0}$ for $i, j \in \{1, \ldots, n\}$ modeling whether there is a gene-gene connection between i and j ($w_{ji} \neq 0$) or not ($w_{ji} = 0$). The second term optimizes *offset* variables $w_i^0 \in \mathbb{R}_{\geq 0}$ with $i \in \{1, \ldots, n\}$ which denote the baseline activity of the genes. The third term enables to deal with noisy data by using *slack* variables $\xi_l \in \mathbb{R}_{\geq 0}$ with $l \in \{1, \ldots, |\Xi|\}$, where $\Xi = \{x_{ik} | x_{ik} = 0, \forall \{i, k\}\}$.

The "penalty" parameter $\lambda \in \mathbb{R}_{\geq 0}$ is defined as a non-negative parameter to control the introduction of the slack variables ξ. Intuitively, if $\lambda = \infty$ slack variables can be infinitely high, and if $\lambda = 0$ slack variables are not allowed.

To determine parameter λ we use Leave-One-Out Cross-Validation (LOOCV). For this, every single observation x_{ik} $\forall i, k$ is removed once. The remaining data is then used to infer a network topology with the LP model. Using the learned topology, the state of the removed gene is predicted by using one of two Gaussian distributions. The choice of the distribution depends on whether the state is assumed to be active or inactive. Then, the mean squared error (MSE) between the prediction and the real observation is computed for 100 predictions in each cross validation step.

To restrict the range for the parameters λ, we define an upper bound on λ. It is not allowed to be larger than $|\Xi| * \sigma^2(x_{ik})$ with $\sigma^2(x_{ik})$ being the variance of the observations x_{ik}, $\forall\{i,k\}$. This bound is chosen based on the worst case where are all $|\Xi|$ slack variables are unequal to zero. The introduction of slack variables is necessary whenever there are contradictory equations which are most probably due to noisy data. Thus, the higher the variance of the data the higher the slack variables can become.

The constraints 6.4 and 6.5 specify the effects of the knockdowns. We assume that the activity of each gene i is given by the sum of its baseline activity (w_i^0), and the activity of its parents ($x_{jk},\ j \neq i$). The activities of the parents are multiplied with the corresponding edge weights ($w_{ji}^+ - w_{ji}^-$) after knockdown k. If gene i is observed to be active after the knockdown k, so $x_{ik} = 1$ (and it has not been silenced, so $b_{ik} = 1$), the sum of all incoming edges and its baseline activity has to be greater or equal to δ_i (constraint 6.4) and smaller or equal to zero otherwise (constraint 6.5). Since we want to model activating, inactivating and no gene interactions (so edge weights being positive, negative or zero), the edge weights w_{ji} have been replaced by $w_{ji}^+ - w_{ji}^-$ with $w_{ji}^+, w_{ji}^- \in \mathbb{R}_{\geq 0}$ according to Section 5 Remark 5.2 in the constraints 6.4 and 6.5.

The inequalities given in 6.6 and 6.7, respectively, force each node which is not a source or sink node to have at least one incoming and one outgoing edge, respectively. Since the edge weights can be positive or negative, the variables w_{ji}^+ and w_{ji}^- have to be added like in the optimization function z, to ensure that the absolute values of w_{ji} are considered.
Both constraints (6.6 and 6.7) are necessary to avoid *lose ends*. By lose ends we mean for example a node which is not a source node but has no incoming information flow, or a node which is not an sink node but has no outgoing information flow. Although these inequalities help, they do not ensure that the resulting network topology is connected. For example if cycles are learned (i.e. node a activates node b, which activates node c, and c activates a in turn) each node might have an incoming and outgoing edge but nevertheless, they are not connected to the source and sink nodes. This problem cannot be solved using a linear model approach as presented here, since it would make edge weights dependent on each other. Consider for example a network with four nodes a,b,c and d, where a is a source and d a sink node. Then, there are three possible network topologies (irrespective whether an edge is activating or deactivating) which do not have lose ends (see Figure 6.1). The edge between a and b can be zero if and only if there is a connection from a to c and from c to b (Figure 6.1, second topology). Otherwise, the network topology has to be like the first or the third topology given in Figure 6.1. Hence, the connection from a to b depends on the other inferred connections and therefore, the inference of edges of a network where lose ends are not allowed is dependent on a given topology. Thus, to force the inference of connected networks it is necessary to explore the whole search space. This,

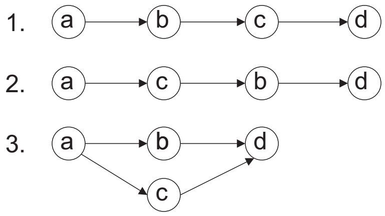


Figure 6.1: Possible connected networks with four nodes. There are three possible network topologies, where the assumption that node a is a source and node d a sink node holds, and where no lose ends (so all nodes are connected with the source and sink node) are given.

however, we avoid as we aim at inferring signaling networks in a fast and efficient manner. This makes our model useful for large-scale problems of dozens of nodes.

We note, that in contrast to DEPNs [FSA$^+$09] the model presented here does not require a reference experiment with no knockdown. In addition, it is not necessary that each gene of interest has been silenced at least once.

6.2 Performance Evaluation

To quantify the performance of a model on simulated data, sensitivity, specificity and precision values are calculated. The model presented here is able to infer three types of edge weights: negative, positive and zero. Therefore, the classical two class receiver operating characteristic (ROC) [SH06, Faw06] can be extended to the three class problem .
For the calculation of the performance analysis we first define how predicted edges are classified (along the lines of [MRRK09]). A predicted edge is defined to be a true positive (TP) if it is actually existing in the underlying true network which has been used to simulate the data, and if its sign is correct (so positive if it is actually an activating edge and negative if it is an inactivating edge). True negatives (TN) are given by non-existent edges of the learned network which are also not present in the true topology. Furthermore, we denote false negatives (FN) as falsely predicted non-existent edges (independent whether they are activating or inactivating in the true topology). Finally, false positives (FP) are all edges which are predicted with a wrong sign or which are predicted to exist when they are non-existent in the true topology. This can be summarized in a confusion matrix given in Table 6.1.
After calculating the TP, FP, TN and FN values, they are used to compute

		Predicted		
		positive edge	negative edge	non-existent edge
	positive edge	TP	FP	FN
True	negative edge	FP	TP	FN
	non-existent edge	FP	FP	TN

Table 6.1: Confusion matrix of the three-class classification problem.

the performance measurements with:

$$Specificity = \frac{TN}{TN + FP} \tag{6.8}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{6.9}$$

$$Precision = \frac{TP}{TP + FP}. \tag{6.10}$$

This allows the calculation of the ROC of an inferred network. Furthermore, it enables the computation of the area under the curve (AUC) value for a ROC-curve as well as for a precision to recall curve (PR-curve), where recall is equal to sensitivity.

For the calculation of the AUC values for random networks which have the same properties (number of positive and negative edges) like the true underlying topology, we permute the edges of a given network and assign them to nodes chosen randomly among the network nodes. This is repeated 1000 times and every time the AUC is computed for the random topologies. The average AUC of the 1000 runs represent the performance of randomly generated networks.

6.2.1 Implementation and used Hardware and Software

All calculations of this thesis have been carried out using a Linux cluster with dual-processor 3.1 GHz XEON quadcore machines with 16 GB RAM. R version 2.12.1 has been used to implement all functions necessary for the LP model given in 6.1 and the cran [R D09] package "lpSolve" version 5.6.5 [Bo10] has been used to solve the LP models. lpSolve is an interface to the open source linear and integer programming solver lp_solve version 5.5. which uses the simplex algorithm for calculating the optimal solution of LPs.

Chapter 7

Applications on Simulated and Real Data

In this Chapter, we evaluate the performance of our LP model based on different types of simulated data. We compare results with those derived with the already published method DEPN [FSA$^+$09] and random guessing. First, we simulate data for a small-scale artificial five-node network to test the performance of different settings of noise and missing data in Section 7.1. Then, networks of different sizes are extracted from existing interaction pathways given in KEGG [KG00] in Section 7.2. Based on these networks, data is simulated and the DEPN approach and the LP model are to infer the true topologies from this data. We show that the LP model outperforms the DEPN approach in terms of prediction performance and computation time (Section 7.3). A real data set studying ErbB signaling is used to apply the LP model on a biological problem in Section 7.4. We identified several already known interactions of the ErbB signaling pathway as well as identify new ones. Finally, the Section 7.5 presents a discussion of this Chapter.

7.1 Five-Node Example Network

Assume the artificial network topology of Figure 7.1 to be given. For this network we simulated experimental measurements for five single, and one double knockdown as well as for one reference experiment as listed in the perturbation matrix (Table 7.1).

For each of these experiments we computed the expected downstream effects. This results in an observation matrix where the observations are coming from two Gaussian distributions, one for activated and one for inactivated genes. A node i is assumed to be active if i is not perturbed and if among its parents more activating than inhibiting parents are active, so if its “inflow” is activating and node i is assumed to be inactive otherwise. Root nodes are

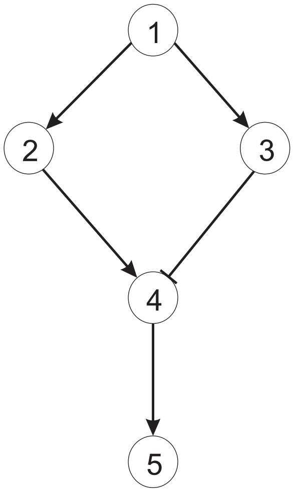


Figure 7.1: Five-node example network topology. Arrows indicate activation and between the node 3 and the node 4 an inactivation is shown.

	gene i				
knockdown k	**1**	**2**	**3**	**4**	**5**
1	0	1	1	1	1
2	1	0	1	1	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	0
6	1	0	0	1	1
7	1	1	0	1	1

Table 7.1: Perturbation matrix B of the five-node example, where $b_{ik} = 0$ if gene i has been silenced in knockdown experiment k with $i \in \{1, \ldots, 5\}$ and $k \in \{1, \ldots, 6\}$. The sixth row corresponds to the double knockdown of gene 2 and 3, and the last row represents a reference experiment without any knockdown.

assumed to be active if they are not perturbed.
For nodes which are determined to be active after a perturbation, we model their activity to come from the distribution $\mathcal{N}(0.95, \sigma)$, and from the distribution $\mathcal{N}(0.56, \sigma)$ for inactive ones. These distributions are chosen in agreement to the average of the activation and the inactivation, respectively, as determined for the real data (Section 7.4). Furthermore, $\sigma \in \{0, 0.01, 0.1, 0.2, 0.27, 0.3, 0.4, 0.5\}$ is used to simulate different amounts of noise level in the data. The parameters δ_i are defined to come from a Gaussian with $\mathcal{N}(0.755, \sigma)$, where 0.755 is determined by taking the average of the means of the activating and the deactivating state distributions. In the following we show network inference results of data simulations for noisy and incomplete data.

7.1.1 Noisy Data

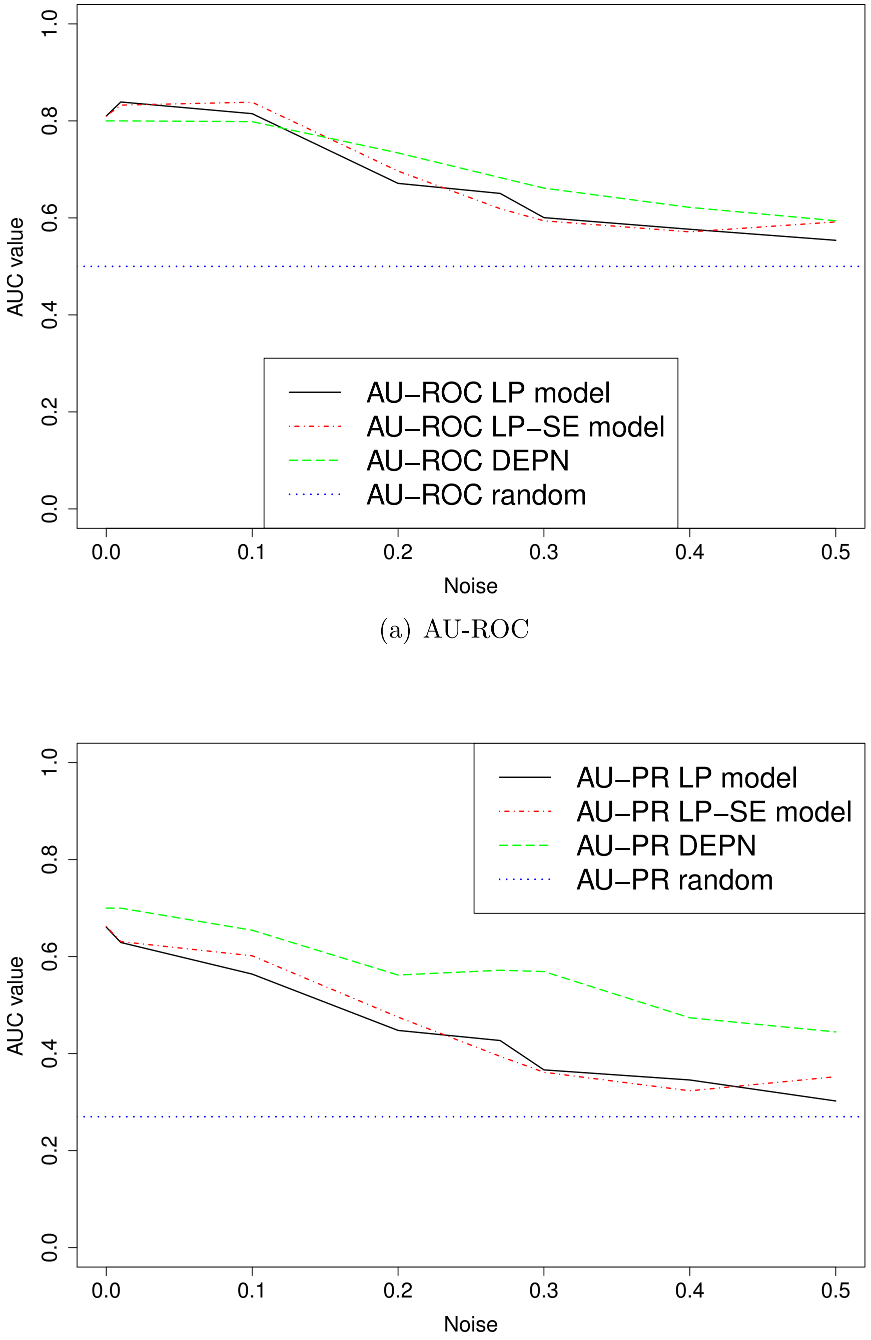
To simulate noisy data we generate 100 times a dataset with three replicate measurements using the activation and deactivation conditions as explained above. For the network inference, we first use the LP model on each generated data. Replicates are summarized by taking the average. The corresponding LPs have been solved with λ being $0 \leq \lambda \leq |\Xi| * \sigma^2(x_{ik})$ (step size of 0.1). For each λ, we perform LOOCV and compute the MSE. The λ which minimizes the MSE is chosen for the corresponding LP model. Since in each cross validation step, different weights for individual interactions can be learned, we compute the median and the median absolute deviations (MAD) of all edge weights learned in the LOOCV. If the absolute value of the computed median is higher than the respective MAD, the edge weight is assumed to be reliable and to be not due to noise in the data. In this case, we use the learned median edge weight as the final interaction strengths. Otherwise, we define the edge weight to be zero, since it cannot be distinguished from noise and thus, it is unreliable.
Our LP model assumes that the source and sink nodes are known in advance. Since this is not always given for applications of real data, we secondly applied our LP model on the generated data without an a priori definition of the source and sink nodes (LP-SE). This implies that the constraints 6.6 and 6.7 cannot be formulated. Therefore, they are not used to infer the network topology for the LP-SE model.

In addition, we use a third approach called DEPN which has been recently published by Froehlich *et al.*, to learn the network topology used to generate the data. To assign each edge with a weight we use greedy hillclimbing and bootstrapping (resampling with replacement) with 100 bootstrap samples as proposed in the DEPN implementation [FMT$^+$].

For each of the 100 generated data sets, we therefore infer network topologies using the three methods (LP model, LP-SE model and DEPN). We make ROC curves of the learned interactions. Each time we calculate the AUC values. Since the DEPNs are not able to learn negative interactions we treat our LP and LP-SE model like in a typical two-class ROC analysis (see Section 6.2). For this, we ignore the signs of the learned interactions by taking the absolute values.

In Figure 7.2 the area under the ROC- (AU-ROC) and under the PR-curve (AU-PR) are plotted for increasing noise ($\sigma \in \{0, 0.01, 0.1, 0.2, 0.27, 0.3, 0.4, 0.5\}$) for the three methods. Moreover, the Figure shows AUC values for randomly generated networks computed like explained in Section 6.2. As we use a two-class ROC analysis for the DEPN, the LP and the LP-SE model, we take only positive interactions for the random guessing AUC value computations, too. The Figure demonstrates that there are only little differences between the LP and the LP-SE model for computed AU-ROC and AU-PR values. Both methods perform well with AU-ROC values larger than 0.8 and AU-PR values larger than 0.6 up to a noise of σ around 0.1. Then, both AUC values decrease until they approach the random guessing line at a noise of σ around 0.5.
The DEPN approach performs better with its AU-PR values being each time bigger than those for the other two methods. The AU-ROC values of DEPNs are up to a noise level of around $\sigma = 0.1$ slightly worse and then slightly better than the LP and LP-SE model.

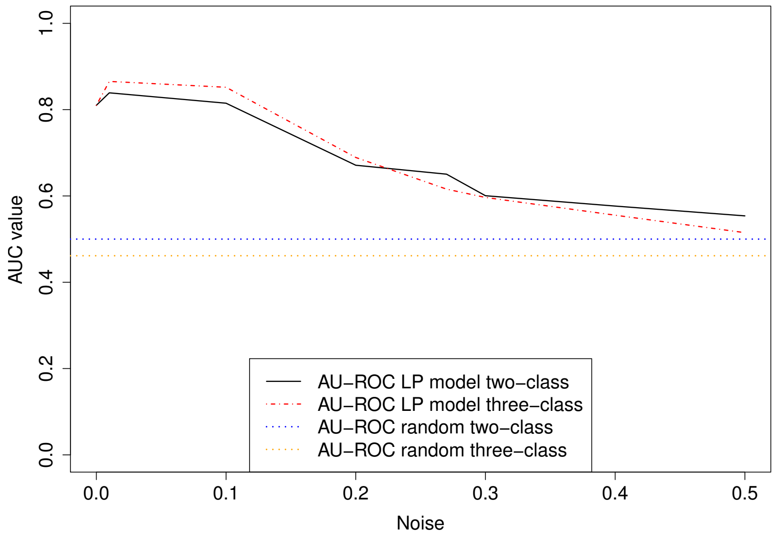
To assess whether our approach can infer also the sign of the edges with a good performance, we computed AUC values for the three-class problem additionally. The Figure 7.3 shows AU-ROC and AU-PR results exemplarily for the LP model (LP-SE model behaves similar) as well as for random guessing values in a two- and a three-class analysis. Not surprisingly, random guessing is more difficult if the edge signs of the interactions are taken into account. Thus, both, the random guessing AU-ROC and the random guessing AU-PR values, are smaller in the three-class analysis than in the two-class analysis. In contrast, our LP model performs similar in both cases. This indicates that the LP method is not able to identify whether the connection is activating or deactivating.
If we do not explicitly note, that we use a three-class classification, we restrict the AU-ROC and AU-PR computations of all problems to the classical two-class ROC analysis in the following. This allows a direct comparison of the results with those derived with the DEPN approach.



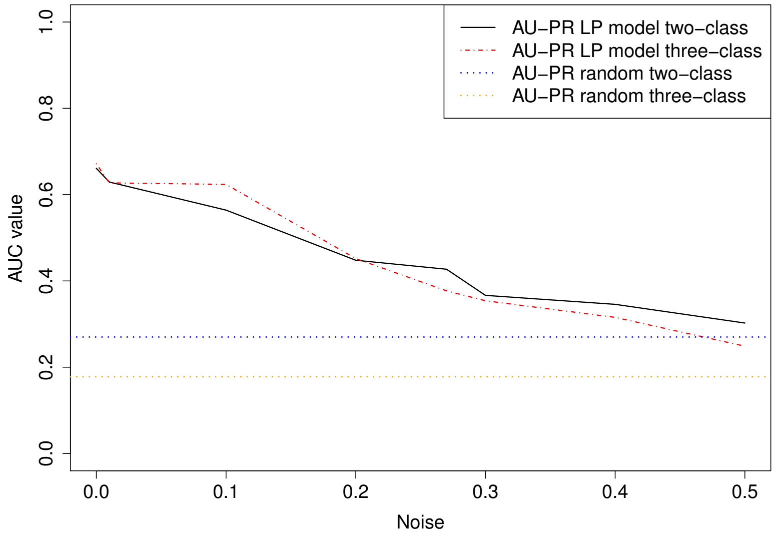
(a) AU-ROC

(b) AU-PR

Figure 7.2: Area under the ROC- and PR-curve for increasing noise $\sigma \in \{0, 0.01, 0.1, 0.2, 0.27, 0.3, 0.4, 0.5\}$ of the LP model, LP-SE, DEPN and random guessing.



(a) AU-ROC



(b) AU-PR

Figure 7.3: Area under the ROC- and PR-curve for increasing noise $\sigma \in \{0, 0.01, 0.1, 0.2, 0.27, 0.3, 0.4, 0.5\}$ of the LP model using a two-class and a three-class classification analysis. The dotted lines represent random guessing values for the two- and three-class classification. The red (dash-dotted) lines, respectively, the straight (black) lines correspond to results of the three-class, respectively, the two-class classification analysis of the results derived with the LP model.

7.1.2 Missing Data

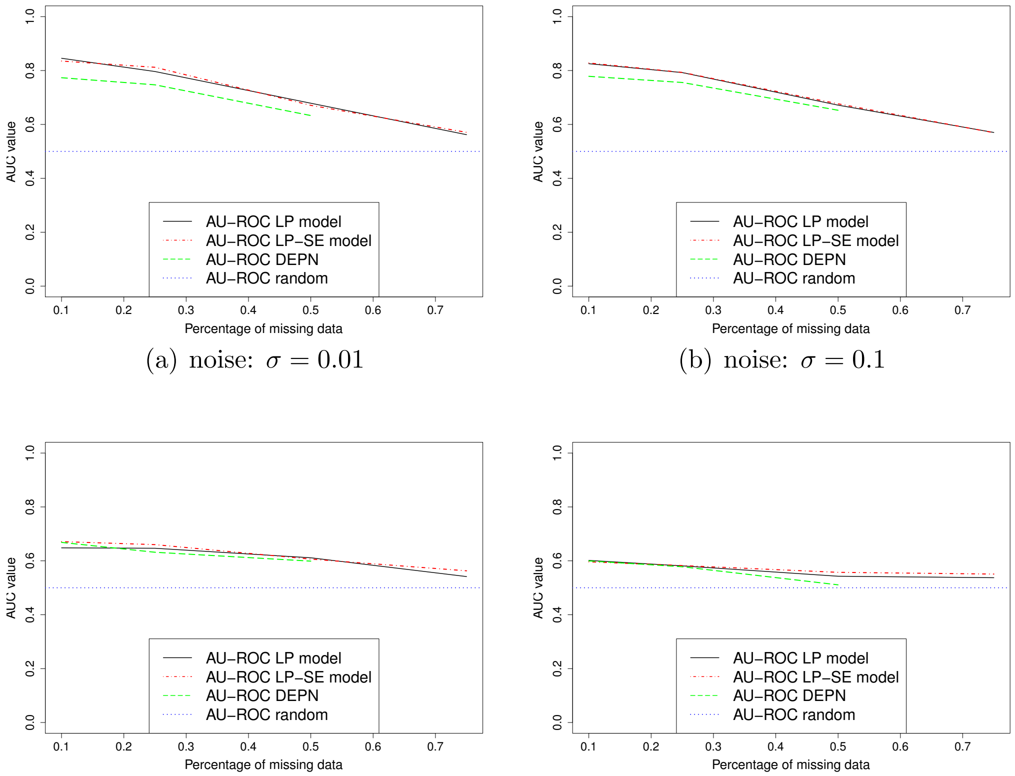
In real biological experiments data is often not only noisy, but may also be incomplete due to measurement errors, missing reagents or simply to save costs. To compare the performance of the three approaches explained in the Section 7.1.1 on incomplete data, we simulate data with different noise levels ($\sigma \in \{0.01, 0.1, 0.27, 0.5\}$) and randomly selected missing data points. Again, we generate 100 times data of three replicates. We define 10%, 25%, 50% and 75% of the data values to be not given. The missing data points are randomly chosen from the data.
For learning the network topology with the LP and the LP-SE model, we summarize replicates using the average. Then, we solve the corresponding problems with $0 \leq \lambda \leq |\Xi| * \sigma^2(x_{ik})$ (step size of 0.1) to find the λ which is minimizing the MSE of the LOOCV. We use the median of the edge weights of all cross validation steps, which has an absolute values larger than the computed MAD as the final interaction strength. In addition, we inferred the network with the DEPN approach using greedy hillclimbing and bootstrapping (100 bootstrap samples). Each time we make the corresponding ROC curves and calculate the AUC values.

The AUC values for the ROC- and PR-curve, respectively, of each different noise level are presented in the Figures 7.4 and 7.5, respectively. Not surprisingly, the three methods perform worse the more data points are missing and the higher the noise. Again, the LP and LP-SE model perform similarly. The DEPN approach cannot infer networks if there are 75% missing data points.

In contrast to the DEPN approach, the LP model learns whether an edge is activating or deactivating. Therefore, we additionally assess the performance of the LP model on noisy data by using a three-class classification analysis on the learned interactions (see Section 6.2). The AU-ROC and the AU-PR values are similar to those of the two-class classification (see Appendix C). Thus, the LP model learns the sign of an interaction with the same performance as it learns the existence of the edge.

7.2 Real Networks from KEGG

Whereas the previous Section evaluated the performance of the LP and the LP-SE model as well as the DEPN approach on a small artificial five-node network, we now use real existing networks in this Section. For this, we



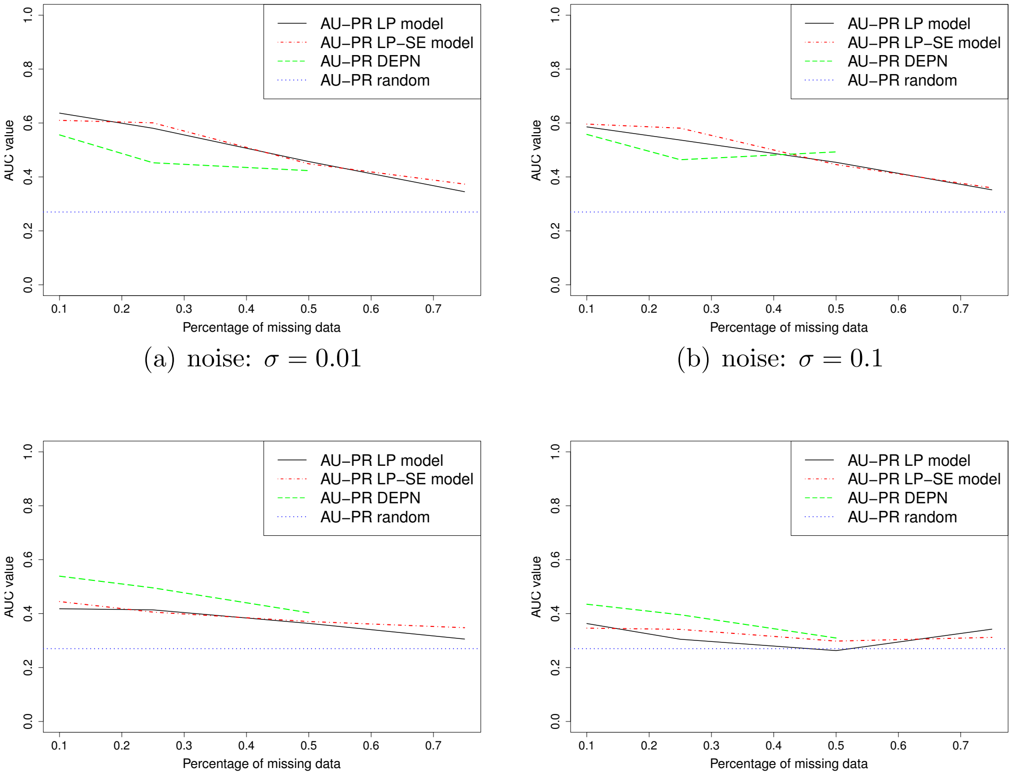
(a) noise: $\sigma = 0.01$

(b) noise: $\sigma = 0.1$

(c) noise: $\sigma = 0.27$

(d) noise: $\sigma = 0.5$

Figure 7.4: Mean area under the ROC-curve using the LP, the LP-SE and the DEPN model on missing data over 100 repetitions of four different noise settings. (a) noise: $\sigma = 0.01$ (b) noise: $\sigma = 0.1$ (c) noise: $\sigma = 0.27$ (d) noise: $\sigma = 0.5$. The dotted lines represent random guessing values.



(a) noise: $\sigma = 0.01$

(b) noise: $\sigma = 0.1$

(c) noise: $\sigma = 0.27$

(d) noise: $\sigma = 0.5$

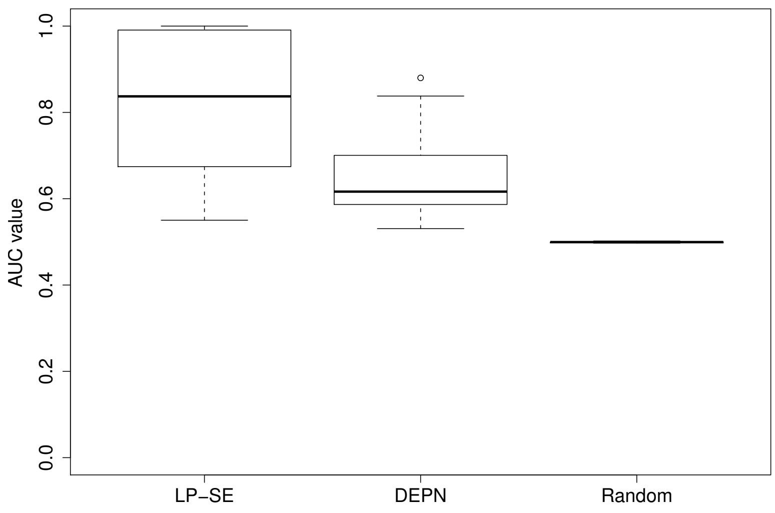
Figure 7.5: Mean area under the PR-curve using the LP, the LP-SE and the DEPN model on missing data over 100 repetitions of four different noise settings. (a) noise: $\sigma = 0.01$ (b) noise: $\sigma = 0.1$ (c) noise: $\sigma = 0.27$ (d) noise: $\sigma = 0.5$. The dotted lines represent random guessing values.

randomly extracted sub-networks of different sizes from eleven different KEGG [KG00] signal transduction networks of varying total size (KEGG IDs: hsa05212, hsa05210, hsa04630, hsa04370, hsa04350, hsa04310, hsa04210, hsa04115, hsa04110, hsa04012, hsa04010). Only gene-gene interactions are considered as edges.

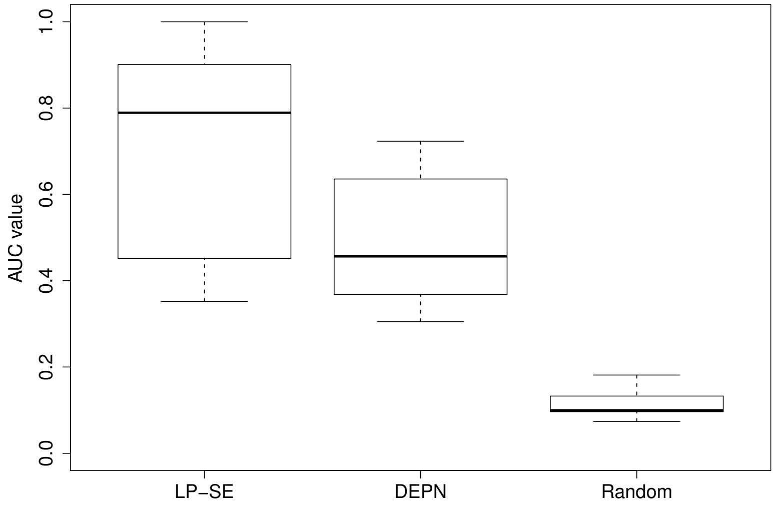
We assume a perturbation matrix for these networks with single knockdowns for each gene $i \in \{1, \ldots, n\}$ with $n \in \mathbb{N}$ to be given. Furthermore, we assume that there are perturbation experiments of $n/2$ double knockdowns, where the genes are pairwise randomly chosen among the given genes. In addition, one experiment is modeled without any perturbation. Using this, we generate data of three replicates for each type of experiment. Each observation is assumed to come from two normal distributions (active or inactive) after a given perturbation as described for the five-node example. An activated node is simulated from the normal distribution $\mathcal{N}(0.95, 0.01)$ and from $\mathcal{N}(0.56, 0.01)$ if it is inactivated. The parameter δ_i is generated using the normal distribution $\mathcal{N}(0.755, 0.01)$.
We assume that source and sink nodes are not given and therefore, network topologies are inferred using the LP-SE model. To find the best parameter λ with $0 \leq \lambda \leq |\Xi| * \sigma^2(x_{ik})$ (step size: 0.1), and to compute a range of possible weights for each edge, we use LOOCV. The median of the learned edge weights of each cross validation step is used as the final connection strength, if its absolute value is bigger than the MAD. In the following we present results of simulations for ten-node and large-scale networks.

7.2.1 Ten-Node Networks

We extract ten different networks with $n = 10$ nodes each from KEGG as explained above. The networks have a varying number of edges: five networks have 7 interactions and the remaining networks have 5, 8, 10, 12 and 13 nodes, respectively. Each connection is positive, so there are no inhibitions.
To compare our approach with the DEPNs we apply additionally the DEPN approach on the simulated data for each network. We make ROC curves and calculate AUC values on the learned edges with the DEPNs and the LP-SE model. Since DEPNs cannot learn negative interactions we use absolute values of our learned interactions, to make ROC results of both methods comparable. The results are shown as boxplots over all ten extracted networks in Figure 7.6. Additionally, the Figure shows random guessing AUC values calculated for each of the ten networks as explained in Section 6.2. Obviously, the LP-SE model performs better than random guessing for both, the AU-ROC and the AU-PR results. The random AU-PR values are small, because most of the selected sub-networks are very sparse with seven of them having less interactions



(a) AU-ROC: ten-nodes



(b) AU-PR: ten-nodes

Figure 7.6: AUC values of (a) the ROC and (b) the PR curves of the network inference using the LP-SE model, the DEPNs and random guessing values on ten-node sub-networks randomly selected from KEGG.

than nodes.

Figure 7.6 clearly shows the increased performance of our LP-SE model in comparison with DEPNs, although DEPNs performed better at the same noise levels for the AU-PR values in the five-node example network. This shows the increased performance of our model on large-scale networks in comparison to the DEPN approach.

7.2.2 Large-Scale Networks

To evaluate the performance of the LP-SE model and DEPNs on large-scale problems, we extract five networks of large sizes with $n \in \{16, 26, 28, 44, 52\}$ nodes. Again, the networks have only positive interactions with 17, 27, 31, 43 and 51 edges, respectively.

For each of the five networks we simulate 25, 40, 43, 67 and 79 perturbation experiments, respectively. These experiments consist of single knockdowns of each of the n nodes, $n/2$ double knockdowns randomly chosen, and one experiment without any knockdown.

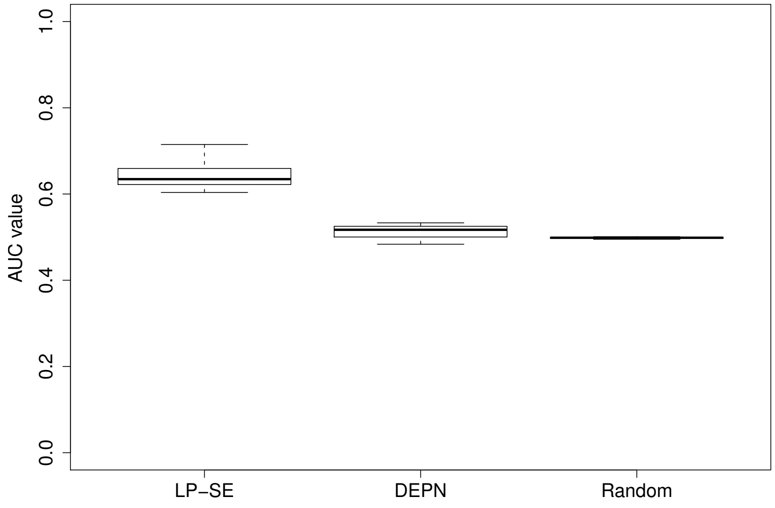
As already mentioned, the run time increases exponentially with an increasing number of nodes and perturbation experiments. Since LOOCV is very time consuming (each measurement is left out once and the network is learned for the remaining data), it slows down our network approach. Therefore, we use for this data 100 times a stratified k-fold cross validation of ten folds. This reduces the number of times the training process is repeated and thus, the total run time is decreased.

Taking the λ which minimizes the MSE, we compute AUC values of the median of all edge weights computed in the individual cross validation steps. The results are shown in Figure 7.7 as well as results derived when using DEPNs as well as with random guessing values.

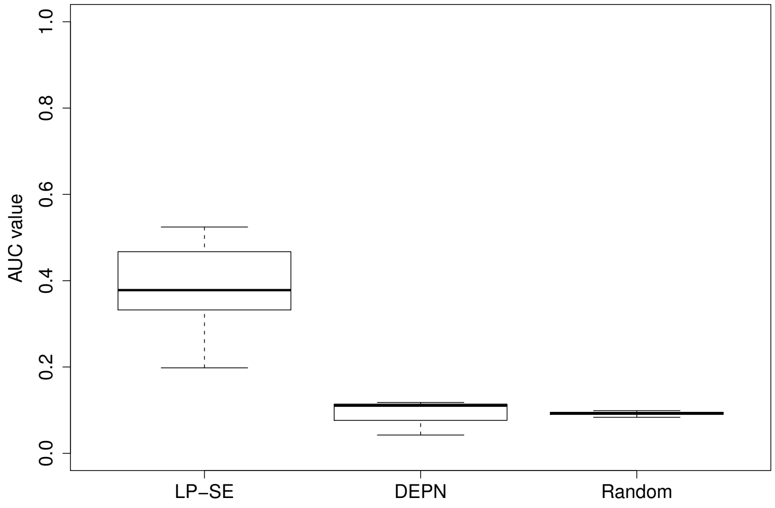
The Figure shows that in this large-scale example the performance of our LP-SE model is again much better than the DEPN approach which is only slightly better than random guessing.

7.3 Run Time

The dimensionality of possible networks increase exponentially with an increasing number of nodes. To assess whether this is true also for the computation time, we measure the used run time of each network inference process. All calculations are carried out as described in Section 6.2.1.



(a) AU-ROC: large-scale



(b) AU-PR: large-scale

Figure 7.7: AUC values of (a) the ROC and (b) the PR curves of the network inference using the LP-SE model, the DEPNs and random guessing values on large-scale sub-networks randomly selected from KEGG.

7.3.1 Run Time of the Five-Node Network

For the data generated with varying noise and varying missing values Table 7.2 summarizes the computation times. Results are represented for all three methods (DEPN, LP and LP-SE model) used for the network inference. The run times are averaged over the generated 100 different data sets for different noise and for different number of missing data points, respectively. The Table shows that the run time is very small for the network inference with the LP and the LP-SE model with taking on average only around three seconds. In contrast, DEPNs take much longer with up to 200 seconds. To conclude, the DEPN approach take on average up to 100 times longer than our models.

			Run time in seconds		
	# nodes	**# exp.**	**LP model**	**LP-SE model**	**DEPN**
noisy data:	5	7	3.08 ± 1.79	3.69 ± 1.76	94.25 ± 7.53
missing data:	5	7	3.54 ± 1.52	3.39 ± 1.45	200.11 ± 132.96
10-node kegg:	10	16	-	471.6 ± 287.4	3283.8 ± 1058.4
largefold kegg:	16	25	-	2996.4	26247.04
largefold kegg:	26	40	-	14279.6	214617.398
largefold kegg:	28	43	-	41798.61	465184.138
largefold kegg:	44	67	-	292748.26	-
largefold kegg:	52	103	-	300906.93	-

Table 7.2: Number of nodes, number of experiments and the mean and standard deviation of the computation time over each run carried out for the network inference with noisy data, missing data and the sub-networks extracted from KEGG. If no standard deviation is shown, the respective network has been inferred only once from the respective data.

7.3.2 Run Time of the Ten-Nodes and the Large-Scale Networks

Similarly to the artificial five-node example given above, we compute the run time the LP-SE model and the DEPN approach need to infer the underlying network topology of the sub-networks extracted from KEGG. The results are represented in Table 7.2.

Whereas the network inference takes only several seconds for the LP-SE model on five nodes, it takes already on average 7.86 ± 4.79 minutes for the ten-node data sets with the 16 simulated perturbation experiments. This reflects the increasing dimensionality of the network inference problem for an increasing number of nodes. The DEPN approach takes even longer than the LP-SE model, and it takes almost one hour (54.73 ± 17.64 minutes) to infer the network topology. Nevertheless, the performance of DEPNs is worse than the LP-SE model in terms of AU-ROC and AU-PR values. This large increase of computation time when increasing the number of nodes has already been reported for the DEPN approach in [FSA+09].

The run time of our LP-SE model for the large-scale networks is 0.83, 3.97, 11.61, 81.32 and 83.59 hours, which means that it takes now considerably longer to infer the network topology than for small examples. However, it is still feasible and much faster in comparison to the DEPN approach which takes 7.2, 73.5, 129.22 hours for the first three networks. For the large-scale problems with 44 and 52 nodes the DEPN approach was not able to finish the network inference process within 1000 hours.

To summarize, Figure 7.8 gives the run times which are needed by the DEPN and the LP-SE model to infer the network topologies of the different data sets. The time is plotted in seconds on a log-scale against the number of nodes (for exact numbers see Table 7.2). Obviously, the DEPN approach takes longer than the LP-SE model. The last two networks, with 44 and 52 nodes, cannot be reconstructed by the DEPN approach in a reasonable time at all.

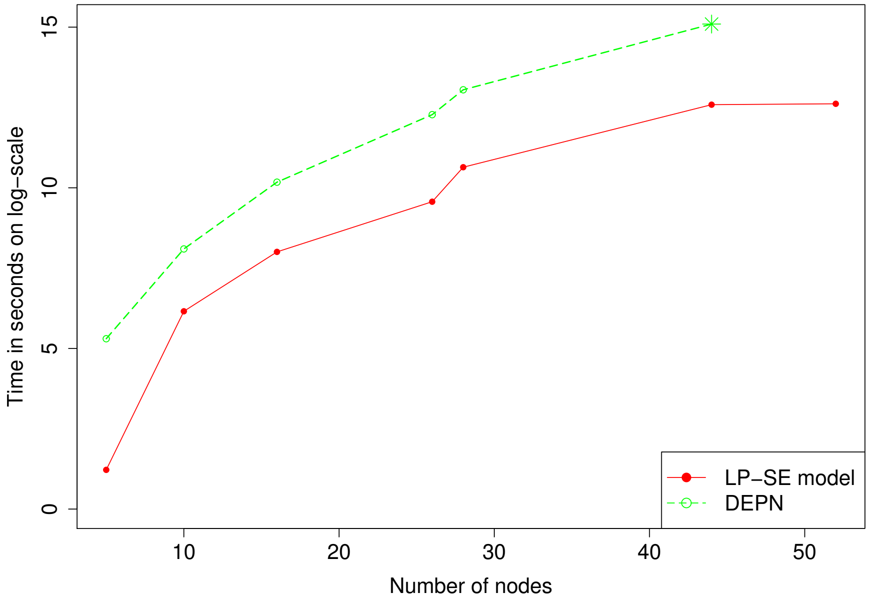


Figure 7.8: Run time in seconds on log-scale to infer the underlying networks of different sizes with the DEPN approach (green circles) and the LP-SE approach (red dots). The green and red lines connect the individual measurements. The green star denotes the time point (1000 hours) at which the network inference with the DEPN approach has been aborted for the network with 44 nodes.

The star in Figure 7.8 indicates the time point, where the DEPN approach has been aborted without finishing the network inference process for the large-scale network of 44 nodes. The LP-SE model is performing much better, as it can infer the underlying network topology within 81.32 hours.

7.4 Real Data Studying ErbB Signaling

In network inference it is difficult to quantify the performance of a proposed method on real biological data since, the underlying true network topology (gold standard) is rarely fully understood. The ErbB signaling pathways are one of the best studied signaling networks. It is known that they regulate diverse physiological responses such as cell division, motility and survival [CY06]. Therefore, we use a recently published data set on ErbB signaling for an evaluation of our LP model on real data.

7.4.1 ErbB Signaling

The ErbB protein family (or epidermal growth receptor (EGFR) family) consists of four structurally related receptor tyrosine kinases (ERBB1-ERBB4). They have the ability of forming homodimers and heterodimers upon activation by epidermal growth factors such as EGF. Thereby, they activate intracellular signaling transduction pathways. The ErbB family is highly involved in human diseases like multiple sclerosis or the Alzheimer disease where ErbB signaling is insufficient due to inactivated ErbB receptors [BY07]. In contrast, excessive ErbB signaling is associated with the development of many human cancers such as breast or lung cancer [SFL$^+$09, BY07].

ErbB proteins are extensively studied to find potential candidates for therapeutical targets. Trastuzumab for example is used to target the ERBB2 receptor in breast cancer cells which are showing ERBB2 overexpression. ERBB2 has no known direct activation ligand but forms activated heterodimer preferentially with ERBB3 but also with ERBB1 [ZBB05], activated by EGF. In normal and cancer cells, ERBB2 controls G1/S cell cycle transition by modulating G1 regulators resulting in an oscillating activity of the tumor suppressor retinoblastoma protein pRB. This protein allows the expression of genes required for S-phase entry [LBM$^+$00]. For a detailed description of the ErbB signaling to regulate of G1/S cell cycle transition see for example [ZBB05, LBM$^+$00, SFL$^+$09].

Cancer cells treated with trastuzumab show an interrupted downstream signaling of ERBB2 followed by a G1/S cell cycle arrest leading to a reduction of the abnormal cell growth and proliferation during cancer formation [SFL$^+$09]. Due to the capability of cancer cells to acquire resistance, for example by receptor-independent activation of downstream signaling molecules, the response rate of trastuzumab is only low and many patients are *de novo* resistant [SFL+09, BY07]. ErbB signaling pathways have been extensively studied, nevertheless, it remains poorly understood how the signaling is affected in treatment resistant cells. Accordingly, ongoing work has to be done to identify new additional targets which are showing a higher response rate and are guaranteeing cell cycle arrest in cancer cells.

7.4.2 ErbB Signaling Data

Froehlich *et al.* studied 16 proteins (ERBB1, ERBB2, ERBB3, IGF1R, ER-alpha, pAKT1, pERK1/2, MYC, Cyclin D1, p27, p21, Cyclin E1, CDK6, CDK4, CDK2 and pRB1) which are involved in the ErbB receptor-regulated G1/S cell cycle transition network of human cells. For a detailed description of the experimental setup see [FSA+09]. In short, the authors used RNAi knockdowns followed by a quantification of effects on the remaining network proteins using reverse phase protein arrays (RPPA) [TQL+06, CTS+02]. They performed 13 single-knockdowns (ERBB1, IGF1R, ER-alpha, pAKT1, pERK1/2, MYC, Cyclin D1, p27, p21, Cyclin E1, CDK6, CDK4, CDK2) and three double-knockdowns (ERBB1+ERBB2, ERBB2+ERBB3, ERBB1+ERBB3) with chemically synthesized siRNAs as well as one experiment with MOCK transfected cells which serves as a negative control. RPPA measurements have been done before, and twelve hours after EGF stimulation for ten intermediates of the network, namely ERBB1, ERBB2, pAKT1, pERK1/2, Cyclin D1, p27, p21, CDK4, CDK2 and pRB1, to quantify their protein expression after each individual perturbation. This has been repeated in four technical and three biological replicates which have been integrated using quantile normalization. The remaining proteins could not be quantified due to lacking antibodies suitable for RPPA.

Additionally, we process the data further by summarizing replicate measurements using averages.

7.4.3 Results

We solve the LP model corresponding to the ErbB signaling data with $\delta_i = MOCK_i$ at time zero (without EGF stimulation) for each gene $i \in \{1, \ldots, 16\}$ and with given source (ERBB1, ERBB2 and ERBB3) and sink (pRB1) nodes. Using LOOCV we identify parameter $\lambda = 1.83$ to be the one with minimal MSE with $0 \leq \lambda \leq |\Xi| * \sigma^2(x_{ik})$ and a step size of 0.01.

The median edge weight of all LOOCV-steps is taken as the resulting learned interaction. The results are illustrated in Figure 7.9 where the inferred edges are color coded with purple corresponding to no interaction (zero edge

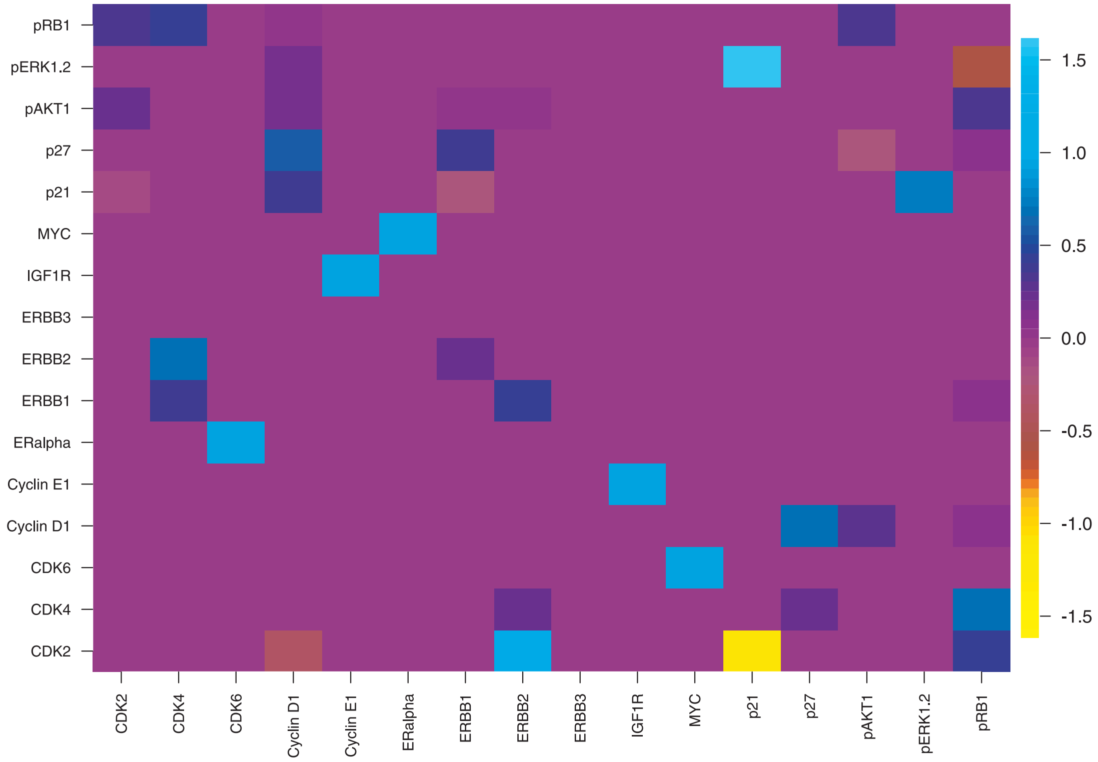


Figure 7.9: Imageplot of median of inferred edge weights w_{ij} of the ErbB signaling data of all LOOCV-steps with i corresponding to genes of the ith column and j to genes of the jth row.

weight), blue to positive and yellow to negative interactions. Tables of the exact numbers of the inferred median edge weights can be found in Appendix D.

In total, we learned 43 interactions, where 34 connections are activating and 9 are deactivating. After removing the interactions which have a MAD higher than its absolute median, 35 interactions (31 activations and 4 deactivations) remain. At first, we start to evaluate the negatively learned interactions. The most pronounced deactivation with a median edge weight of -1.07 is from p21 to CDK2. This inhibition has already been reported in the literature [HEK+95]. Next, with median edge weights of -0.6 and -0.36 we inferred deactivations of pERK1/2 by pRB1 and CDK2 by Cyclin D1. Both inhibitions seem biologically plausible feedback loops, to control the G1/S cell cycle transition, although they have not been directly reported in literature yet. However, there is some evidence that CDK2 is co-precipitated with Cyclin D1 [BBM+94].
Furthermore, there is a deactivation with median edge weight of -0.2, namely the inactivation of p21 by ERBB1. This interaction can be found in the literature as an indirect path which goes via pAKT1 and MYC [FSA+09].
The remaining inhibitions have median edge weights smaller than -0.04 or a MAD higher than its absolute median and thus, are not pursued further. We note, that the DEPN approach cannot infer negative edge weights and therefore, they are not able to learn deactivations. Nevertheless, the above mentioned interactions have been inferred using the DEPN approach, except for the interaction learned between pRB1 and pERK1/2.

The interaction which has been learned with the highest positive median edge weight of 1.62 is the activation of pERK1/2 by p21. This strongly indicates that there is a positive feedback loop controlling the G1/S cell cycle transition. However, this has not been shown in the literature yet. The median edge weight of 1 has been inferred for the activation of CDK2 by ERBB2. Although the direct connection of these proteins has not been reported in the literature, there exist two indirect signaling paths: ERBB2→pAkt1→MYC →Cyclin E1 →CDK2 and ERBB2→pERK1/2→MYC→Cyclin E1→CDK2 [FSA+09] which support our results.
There are five interactions which have been learned with a median edge weight of 0.95: MYC activates CDK6, IGF1R activates Cyclin E1 and vice versa, CDK6 activates ERalpha and ERalpha activates MYC. The last activation is known from literature [SPWM98]. The connection between IGF1R and Cyclin E1 is known by an indirect path via pERK1/2 and MYC [FSA+09]. The two other interactions between MYC and CDK6, and between CDK6 and ERalpha are newly predicted activations. Using the DEPNs on the same data an indirect path has been learned from MYC to CDK6: MYC → p27 →

CDK4 → Cyclin D1 → CDK6 [FSA+09].
Among the remaining learned activations with lower edge weights the activation of ERBB1 by ERBB2 and vice versa are worth mentioning, since the two kinases are known to form heterodimers (see for example [CY06, CSJ+09]).

To evaluate the results further, we use a reference network extracted from the String database [Str11]. For this, we use all interactions of the 16 proteins under study which have a combined confidence score higher than 0.85. Since the interactions given in String are undirected and unsigned, we remove the edge weights of our inferred network topology as well as the signs. Based on this reference network, we compute sensitivity, specificity, precision and accuracy (see Section 6.2) of our results. In addition, we calculate the evaluation values for a "random" network where we randomly permute the given interactions of the String reference network 100 times and take the average of the computed values. Furthermore, we present results of the network inferred by the DEPN approach as published in [FSA+09].
The evaluation results are shown in Table 7.3. The accuracy of both inference methods, DEPN and the LP model behave similar with an accuracy of 0.63. This is better than random guessing where an accuracy of 0.56 is achieved.

	LP model	**DEPN**	**random**
TP	9	7	13.11
TN	72	73	59.11
FP	15	14	27.9
FN	32	34	27.9
SP	0.83	0.84	0.68
SN	0.22	0.17	0.32
PR	0.38	0.33	0.32
AC	0.63	0.63	0.56

Table 7.3: Computed evaluation values of the ErbB data after network inference using the LP model, the DEPN approach and random guessing. TP= true positives, TN= true negatives, FP= false positives, SP= specificity, SN= sensitivity, PR= precision, AC= accuracy (see Section 6.2).

In conclusion, we learned several already known activations and inactivations as well as inferred potential new interactions of the ErbB signaling. The most interesting new predictions are those which indicate negative or positive feedback loops since they allow to regulate and control the the G1/S cell cycle transition. This seems to be biological plausible, since Chen *et al.* reported, for example, that the ErbB response is silenced by negative feedback from active ERK [CSJ^{+}09] which is thus supporting the idea of feedback loops in the ErbB signaling.

7.5 Discussion

We formulated the network inference task as a linear optimization program which can be solved efficiently even for large network sizes where other methods suffer from the exponentially increasing dimensionality. Our model can use measurements from single as well as from double or multiple knockdowns at the same time. In addition, prior knowledge for example from databases, protein-protein interactions or expert knowledge can be easily integrated into the linear program by formulating additional constraints. The model presented here infers not only the existence of an edge, but also its sign, so whether there is an activation or a deactivation between two nodes.

One drawback of the LP model is that time-series data cannot be integrated up to now. We are using steady-state data and thus, loops within the network topology can be resolved only up to a certain level by using double or multiple knockdowns.

Using the formulation of a linear optimization problem allows an efficient calculation of the topology which best fits the given data. However, there might be topologies which are closer to biology although they are not minimizing the given LP. To overcome this problem and to compute a range of edge weights between the genes, we use LOOCV for the small, and a stratified k-fold cross validation for the large-scale problems.

We showed based on simulated data for a five-node example that the LP and the LP-SE models are able to deal with noisy as well with incomplete data with a similar performance like the DEPN approach. However, our linear models are much faster in computing the results than the DEPNs. Using sub-networks extracted from signaling pathways given in KEGG, we showed that our LP-SE model is not only much faster than the DEPN approach but has in addition a better performance. The LP-SE model is clearly outperforming the DEPNs on the large-scale networks, and it is much better than random guessing. Our model is immensely speeding up the network inference task with producing even better results than the DEPNs.

Using RNAi in combination with RPPAs studying the ErbB receptor-regulated G1/S cell cycle transition network of human cells, we applied our LP-SE model on a real biological problem. Among the inferred interactions several have been reported in the literature before and inferred using the DEPN approach. We use a reference network extracted from String to compare our results with the current level of biological knowledge of the ErbB signaling. Our results achieve an accuracy better than random guessing. The newly identified connections indicate that there are negative and positive feedback loops within the ErbB signaling pathway. This may be important to regulate or control the signaling process in some way, however, to validate this, new experiments are necessary.

Chapter 8

Conclusion and Further Extensions

In this thesis a novel approach for the reconstruction of signaling networks from high-throughput perturbation data has been introduced. The perturbations can be caused for example by RNA interference. RNAi is especially well suited to identify the effect of gene knockdowns on a certain phenotype. However, biological experiments are always suffering from noise due to technical problems or to biological variations for example when measured cells are in different states of the cell cycle. For inferring signaling networks from real data, the number of false positives has to be decreased to a minimum. We proposed in this thesis a novel network inference model. This model allows a fast and efficient reconstruction of gene signaling pathways from RNAi data. Furthermore, we developed strategies for an optimal data analysis and statistical hit scoring of cellular based RNAi screening data. This allows to reveal the real biological signal and to diminish the effects purely due to noise.

8.1 Single-Cell Data Analysis

The technology of producing RNAi screening data has immensely improved during the last years. This allows the quantification of several phenotypic effects for individual cells of a screen at the same time. Using two high-content high-throughput virus infection RNAi screens, we studied a Hepatitis C and a Dengue virus data set. We assigned each cell of the given data with four technical, and six cell context features. We showed that the viral phenotype after individual perturbations is, beside the technical variability, highly dependent on cell context features.

Cell Context Features

For both screens, the size of the cell has been identified as the most contributing biological factor influencing virus infection. This is in large contrast to a recently published study performed by Snijder and colleagues [SSR$^+$09]. The authors showed that the infection of cells with Dengue virus is mostly influenced by the location of a cell at the middle or at the border of a local population of cells. The discrepancy with our results may be due to three factors which are different in the two studies: first, Snijder *et al.* did not use RNAi data but cellular screens without any perturbations. Second, they infected and measured effects on HeLa cells, whereas in our Dengue screen Huh7.5 cells have been used. The third and last point is given by the two different platforms used for the experimental setup. In the study performed by Snijder *et al.* well-plates were applied, and in contrast, our data has been generated using chambered coverglass slides.

Each of these factors may cause the different behavior of the cells in our screen in comparison to the data used in the Snijder *et al.* publication. The last point, using different experimental platforms, results in quantitative differences of the number of cells in each spot/well. Whereas on LabTeks there are around 100-400 cells per spot, on well-plates up to several thousands of cells are within each well. Furthermore, using well-plates the individual experiments are spatially separated. This is not the case on LabTeks where the risk of cross-contamination can be higher than for well-plates. Thus, both studies depend already on completely different technical settings.

Although, we could therefore not reproduce the results from Snijder and colleagues, our results strongly indicate that viral infection in RNAi screening data is significantly influenced by the cell context. Therefore, analysis and hit scoring strategies should take this information into account to guarantee an optimal processing of the data. This hypothesis may be further supported by already published screens, which do not consider this information and which are showing only a small overlap of identified hit genes even for the same virus.

Data Normalization

We used multivariate adaptive regression splines [Fri91] to perform a non-liner normalization of the RNAi data against the measured features. Thereby, the influence of the individual feature on virus infection has been decreased to a minimum and the true biological signal has been unraveled.

In the next step, we adapted the idea of gene set enrichment analysis for the use with RNAi data, to enable the calculation of significance values of identified hits based on individual cell measurements. This has the advantage that each cell can be viewed as a “replicate” measurement and the significance can be calculated for individual knockdowns, individual siRNAs or genes. Obviously, individual cells of one spot are not real replicates as such, since they are treated in the same way. However, as we clearly observed two different distributions of cell signal intensities within one spot, this indicates that even the cells which are treated exactly similar, show different phenotypes. Individual cells can show high variabilities in their behavior if they are for example in a different state of their cell cycle. Using thus the information of individual cells is a great advantage especially for smaller screens, where no replicate plates are available.

Performance Evaluation

For the HCV screen, we evaluated our results using positive and negative controls. In addition, we used results of a secondary screen to assess the performance of our results. Published by Reiss and co-authors [RRB$^+$11], the validation screen has been performed on hit genes identified using a normal analysis method which does not take the cell context into account. We calculated significant hits with our method, the one published by Reiss and colleagues as well as a third one published by Suratanee *et al.* [SRM$^+$10]. There is only a small overlap of six hits identified with the three competing approaches. This illustrates, once more, the need of a sophisticated analysis strategy to diminish the already reported small overlap of identified hits in different studies. Since our approach showed superior sensitivity and specificity of the data, this is clearly indicating that the use of individual cell measurements highly improves the data analysis task.
Our CELL-BASED approach consists basically of two individual methods: the normalization of the data based on the computed cell context and technical features as well as the hit scoring using the approach of a GSEA. We applied both methods, independently of each other, on the HCV data and analyzed their performance. Both performed better than RIPLEY and AVERAGE but not as good as the CELL-BASED approach. GSEA-ONLY is the factor which contributes more than MARS-ONLY to the final good performance of the combined approach.

In addition, we performed a pathway analysis using DAVID [6.711] on the identified host dependency factors of the HCV and DENV screen. Using our CELL-BASED method more than 20 pathways could be identified to be significantly enriched in HCV processing. Significantly enriched processes for both screens include endocytosis, focal adhesion, signaling in the immune system and the ErbB and MAP kinase signaling. All of them have been already reported in a study considering several RNAi screens at the same time (see Reiss *et al.* [RRB$^+$11]. This is a further indication of the increased sensitivity when using our analysis and hit scoring approach. Moreover, we identified several additional pathways such as the regulation of the actin cytoskeleton, purine metabolism, TLR signaling and several cancer-related pathways.

The large number of identified pathways with the CELL-BASED method is in huge contrast to the number of pathways which are predicted using AVERAGE or RIPLEY. For the AVERAGE method only two pathways have been identified. One of them has been learned also with the CELL-BASED method. The RIPLEY method performs even worse and found no pathways at all, although we used all genes as hits which have a p-value smaller than 0.05 and clustering scores smaller than zero here. For the other methods we uniquely defined hits based on scores which are larger than 1.5 times the standard deviation of the underlying score distribution, additionally to the thresholds on p-values. However, there are only a few genes which would fulfill this criteria for the RIPLEY approach and results of the performance analysis would be even worse.

Although there is only a small overlap of identified hit genes for the Dengue and Hepatitis C data, we identified various enriched pathways which both viruses have in common and which are demonstrating the close evolutionary relationship of the two viruses.

Non-Viral RNAi Data and Cytopathic Effects

In addition to the viral infection RNAi screens analyzing Dengue and Hepatitis C virus we used a non-virus screen to study the cells' population context effects on the phenotypic outcome. Results show similar trends as for the virus screens, with the cell context highly influencing the phenotype. Moreover, using the CELL-BASED analysis strategy results in an increased performance on the correct classification of the controls, in comparison to an approach which does not take the cell context into account.

Clearly, one of the problems in our study is that we cannot exclude that there are cytopathic effects. Cytopathic effects are virus-induced phenotypic effects which can result, for example, in an increased cell growing. Normalizing against cell context features which are due to cytopathic effects may lead to wrong results. One solution to account for this is using additional controls where cells have not been infected. Then, these controls can be used to test whether the cells show different population context effects than when treated with the virus. In the screens analyzing the Dengue and Hepatitis C virus, however, such controls are not given.

Since results of the GSEA-ONLY analysis of the HCV data indicate that this has a higher contribution on the final results than the normalization against the population context, and since both methods perform even better than the stand-alone methods, we conclude that the normalization is not destroying but revealing the true biology.

Taken together, the increased performance of our analysis highly advocates that the use of single cell information greatly enhances the identification of hit genes which are significantly involved in certain processes. However, this is only a first step towards the development of new strategies which take individual cell information into account. Further RNAi screening experiments and analyses on the produced data will allow to gain a better understanding of the ongoing biology and thereby, to improve the proposed methods.
Moreover, it is possible to extract many more cell context features for each cell than those presented in this thesis. Additional fluorescence markers during the experimental setup can be used to visualize, for example, the size of the cytoplasm or individual proteins. Using microscopy-images, the signal of these markers can be quantified and therefore, it can be used for the data analysis and normalization. This allows a more detailed data analysis and thus, it may lead to improved and more reliable results.

8.2 Network Inference

After the identification of the factors which play a role in specific phenotypes using perturbation data, we want to learn how theses factors interact with each other in the underlying signaling cascade. This allows to understand the biological processes in much more detail. However, the network inference from RNAi data is a challenging task [MS06]. One of the most pronounced difficulties is the exponentially increasing dimensionality for an increasing number of elements. This restricts the application of already published approaches such as NEMs [MBS05] and a probabilistic boolean model [KDZ$^+$09] to only small-scale networks of up to six nodes.

Advantages

In this thesis, we developed a model which overcomes the limitations due to a high dimensionality of the problem. The approach has been formulated as a linear programming problem which can be solved efficiently, even for large-scale networks, using the simplex algorithm.
The approach of the LP model is based on the assumption that the signaling of a network is represented as an information flow. Assuming that this flow starts from one or several source nodes, it is propagated down through the network, and finally it reaches one or several sink nodes. We assume furthermore, that the perturbation of a gene influences other genes which are

further down in the underlying topology in a deterministic way. This is used to identify genes which show an effect after the perturbation of another gene. This, in turn, allows to infer interactions between individual genes. In the LP model, the activity of a gene is given by the sum of the baseline activity of the respective gene, and the information of the parent nodes. For this, the incoming flow of the parents is summed up. In addition, slack variables are used to deal with noisy data. Furthermore, the LP model integrates data of double or multiple knockdowns at the same time, and a network topology with activating as well as deactivating connections is inferred. Self-regulating edges are not allowed, but feedback and feedforward loops are possible.

Since the model is formulated as a LP, additional constraints can be easily included. This allows the easy incorporation of prior knowledge information. The prior knowledge can be extracted from different sources such as the KEGG database, PPI networks, the literature or expert knowledge. For each known interaction between a pair of genes an equation or an inequality can be included into the model, as long as they are not contradictory. However, great care is needed to prevent overfitting. Adding too many additional constraints reduces the inference from data, and favors the inference from prior knowledge. If the data is supposed to be reliable, this should be avoided. On contrast, if the data is noisy or incomplete or both, the inclusion of additional information using may result in better predictions.

Evaluations on Simulated and on Real Data

For the evaluation of the LP model different types of data have been used. First, we applied our method on simulated data generated for a small artificial five-node network. This small network has been used, to analyze the robustness of the LP model on noisy and incomplete data. To do this, we simulated data with different levels of noise and with different amounts of incomplete data. Then the LP model has been used to infer the network topology using the simulated data. The LP model assumes that the source nodes and the sink nodes are given. Since this assumptions is not always fulfilled, we modified the LP model into the LP-SE model. The LP-SE model can be used if the information about source and sink nodes are not given. Thus, we use additionally the LP-SE model for the network inference of the simulated data of the five-node network to quantify differences of the prediction results. Both methods perform similarly well when comparing results of the ROC and the PR analysis.

Furthermore, we showed for the LP model and for the LP-SE model that the network predictions perform much better than random guessing for small noise levels. Naturally, the results are worsened the higher the noise, and the more data points are missing.

In addition, we compared the LP and the LP-SE model with the DEPN approach. Applying the DEPN method on the simulated data results AU-ROC and AU-PR values comparable to those given for the LP and the LP-SE model. The DEPN method seems to perform slightly better than the LP and the LP-SE model in terms of the AU-PR values for noisy data. However, DEPNs cannot reconstruct the topology, if more than 75% of the data are missing. In contrast, this is still possible for the LP and the LP-SE model with a performance better than random guessing if the noise is not too large. Since the DEPN approach cannot learn the sign of an edge, we computed the AU-ROC and the AU-PR values using a standard two-class classification. However, the LP and the LP-SE model learn a signed network topology. This results in a three-class classification problem, where an edge can be present, positive, or negative. To evaluate, how good the LP and the LP-SE model predict the sign of an edge, we computed the AU-ROC and the AU-PR values additionally for the three-class classification analysis. The results are similar in both cases. This indicates, that the LP and the LP-SE model cannot only distinguish whether an edge exists or not, but also whether the edge is activating or deactivating.

To assess the performance of the model on real networks given in biology, we extracted sub-networks of given signaling pathways from KEGG. First, we used ten different ten-node networks and then five different large-scale networks with 16 to 52 nodes. Assuming that source nodes and sink nodes are not known, we used the LP-SE model and the DEPN approach to infer the network topologies. The results of both methods were quantified by computing the AU-ROC and the AU-PR values, and this clearly shows that the LP-SE model performs much better than the DEPN approach. Furthermore, both methods are better than random guessing for the ten-node networks. For the large-scale problems however, the DEPN approach behaves not better than guessing.

Next, we analyzed the run time which is needed to infer the underlying network topologies with the DEPN and the LP model. Clearly, the LP model is much faster than DEPNs for all given data sets. For the LP and the LP-SE model, the network inference takes only several seconds using the five-node data, whereas the DEPNs need up to one and a half minutes. For the ten-node networks, both methods need in general longer to learn the topologies than for the five-node example. However, the LP-SE model is clearly outperforming the DEPN approach. The LP-SE model takes only around 8 minutes on average for the ten-node networks in comparison to 55 minutes the DEPN method takes. Certainly, the network inference process takes longer the more nodes are given since the dimensionality of the problem increases. This trend is more pronounced for the DEPN approach than for the

LP model. The DEPN approach for example cannot reconstruct the network with 52 nodes within a reasonable time. Thus, the inference process has been aborted after 1000 hours. For this 52-node network, the LP-SE model needs a long time, too. Nevertheless, the LP-SE model is able to infer the network and it needs around 84 hours to do this.

Finally, we applied the LP model on a real data set. For this, we used data studying 16 proteins involved in the ErbB receptor-regulated G1/S cell cycle transition signaling of human cells. The data included 13 single, and three double knockdowns, as well as a reference experiment without any perturbations. We were able to learn several already known positive and negative interactions. In addition, some of the inferred edges using the LP model have been reconstructed with the DEPN approach, too. However, whereas the DEPN approach does not infer the signs of the edges, we correctly identified whether interactions are activating or deactivating for several of the already known connections. We compared the results derived with the LP model with a reference network extracted from the String database [Str11]. We achieved an accuracy of our predictions of 0.63. This is similar to the accuracy of the ErbB network as published in [FSA+09] which has been inferred using the DEPN method. Both, the results of the LP model and the DEPN approach, are better than random guessing (accuracy of 0.56).
Beside the already known edges, the LP model identified yet unknown interactions which are strongly indicating that the ErbB signaling is controlled by internal feedback loops. Since ErbB signaling is involved in several human diseases, this may be a first attempt for the development of new drug targets. Obviously, further experiments are necessary to allow a better judgment of this indication.

Limitations and Drawbacks

Our LP model infers network topologies by using a linear model. The learned network topology is the one which minimizes the linear program and thus, the LP model gives only one final solution. In biology, however, there may be other solutions which are, although not minimizing the LP, representing the true underlying biology in a better way. Therefore, we used LOOCV to compute the range of the edge weights for all possible connections. LOOCV allows furthermore to find the parameter λ which minimizes the mean squared error.
Moreover, using the LP approach, it cannot be ensured that the learned topologies are completely connected. Therefore, we defined specific ranges for λ, which allowed to control the trade-off between network sparseness and connectivity. To force a network to be connected, an exploration of

the full search space of connected topologies would be required. However, we are aiming at finding suitable networks fast and efficiently. This makes our approach also useful for large scale networks of up to dozens of genes. Therefore, we preferred to learn the networks in a feasible time, instead of demanding them to be connected.

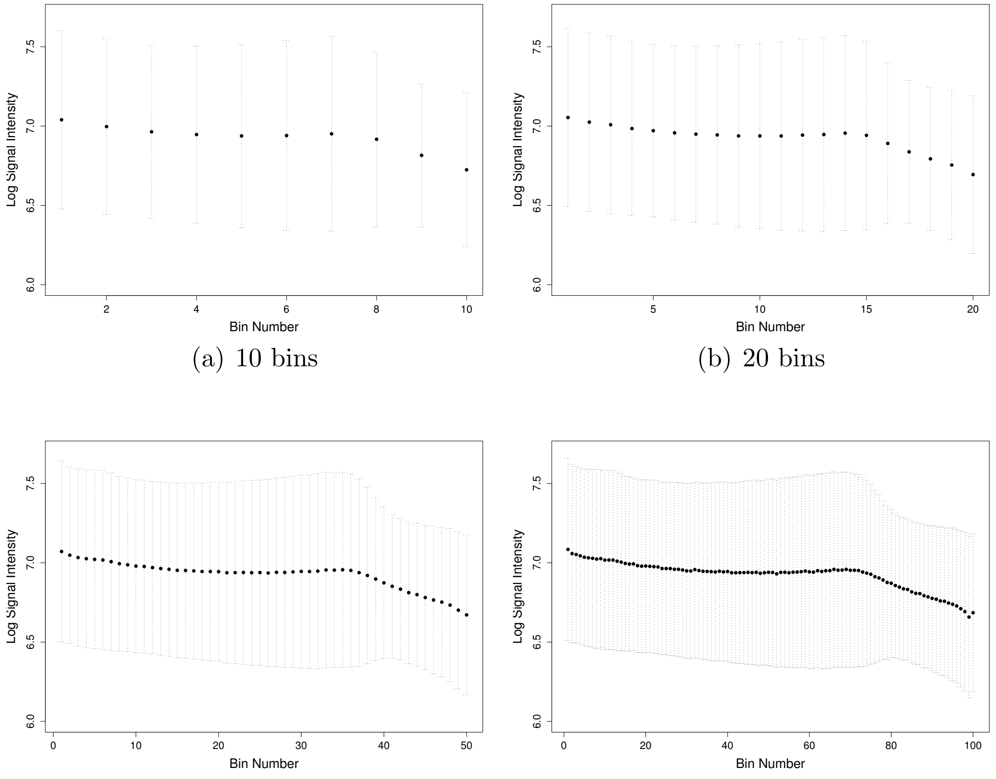
A further drawback of the LP model is the fact that it cannot be applied on data with different read-outs. Even replicate measurements of the same perturbation experiment are at the moment summarized for the use with our LP model. This can result in an information loss and it may be advantageous to use replicates implicitly in the model. Similarly, it may improve the results if different read-outs are modeled implicitly during the network inference process.

Moreover, time-series data cannot be integrated into the inference task as the LP model assumes steady-state conditions. One possibility to use time resolved data in our model may be via the modification of the baseline parameter of each gene. Instead of having only one baseline parameter, several can be used to model the effect after different time-points. However, the baseline activity of a later time-point depends on previous time-points, and as mentioned in Section 6.1 dependencies of the parameters cannot be modeled using a linear approach without enumerating the full search space. This, in turn, would result in an increased run time and thus, this would abate one of the most pronounced advantages of our method.

Appendix A

Feature Plots of DENV Data

Figure A.1 shows the mean and standard deviation of feature "cell size" of the DENV screen for different bin sizes with $n = \{10, 20, 50, 100\}$ bins.



(a) 10 bins

(b) 20 bins

(c) 50 bins

(d) 100 bins

Figure A.1: Mean and standard deviation of feature "cell size" of the DENV screen for different bin sizes.

Appendix B

Hit Lists

B.1 Hit List of the HCV Screen

Table B.1 lists the hits of virus host dependency factors of the HCV screen after CELL-BASED, AVERAGE and RIPLEY analysis.

Table B.1: Gene hit list of the HCV screen after CELL-BASED, AVERAGE, RIPLEY, GSEA-ONLY, and MARS-ONLY analysis. Shown are genes which scored at least in one of the methods to be significantly reducing HCV infection. For CELL-BASED, respectively, RIPLEY the *ES*, respectively, the clustering scores are shown. For AVERAGE, MARS-ONLY and GSEA-ONLY the respective z-scores are given. If the values are not significant in one type of analysis they are represented by empty cells. The last column shows whether a gene also came out as a hit in the DENV (indicated by the *ES* of the DENV data) or not (indicated by empty cells).

Gene	ENTREZ ID	RIPLEY	AVERAGE	CELL-BASED	GSEA-ONLY	MARS-ONLY	HIT DENV
ACK1		-0.7748348125	0	0.1131170606	0.1077080164	-0.0603939278	0
ACVR1C	130399	0	0	0.1176341058	0	0	0
ADK	132	0	1.8607851501	0.10779213	0.1279761403	-0.0584024124	0
AKAP12	9590	0	1.6605915884	0.136484711	0.171599855	-0.1042494399	0
AKAP8L	26993	0	1.5756255801	0.1377499254	0.153155406	-0.0788223588	0.1171384091
AKT3	10000	0	0	0.119397414	0.1257064011	0	0
ANKK1	255239	-0.644279625	0	0	0	0	0
AURKB	9212	-0.5961612353	0	0	0	0	0
BMPR2	659	-0.6404928667	0	0	0	0	0
CAMK1G	57172	0	4.1367581856	0.313214574	0.3162151265	-0.2462775418	0
CAMK2B	816	0	1.8788602376	0.1343924019	0	-0.0914607818	0
CARKL		0	2.5897285903	0.1469633834	0.1793797178	-0.1155279769	0
CD4	920	0	0	0	0.1155559384	0	0
CD81	975	-2.1275204792	3.0996941663	0.2406700954	0.2777782757	-0.1847533501	0
CDC42BPA	8476	0	1.5619325443	0	0	-0.0941096449	0
CDK2	1017	0	1.9173104848	0.1312752438	0.1315034884	-0.0930829879	0
CDK4	1019	0	0	0.1451628991	0	-0.1043589479	0.1218911378
CDKL2	8999	0	0	0	0.1206765735	-0.1050797967	0
CFL1	1072	0	1.8981202251	0.1330411821	0.1385308148	-0.1012358681	0
CHKA	1119	0	2.1643100012	0.1655656259	0.1710879978	-0.128815603	0
CHKB	1375	0	0	0	0.114662443	0	0
CKI-alpha		0	1.582496722	0	0.1127611138	-0.0599394343	0.1377499254
CKMT1		-0.6091872353	0	0	0	0	0.2521418998
CNKSR1	10256	-0.324138875	0	0	0	0	0
CSF1R	1436	0	0	0.1190093497	0.1184299761	-0.0885108991	0
CSNK2A1	283106	-0.6258927647	2.1655126694	0.1617886468	0	-0.0949020828	0
CXCR4	7852	-0.6284616133	0	0.1533465648	0	-0.076305372	0

Continued on next page

Table B.1 – continued from previous page

Gene	ENTREZ ID	RIPLEY	AVERAGE	CELL-BASED	GSEA-ONLY	MARS-ONLY	HIT DENV
DGKZ	8525	0	0	0	0	-0.0782614223	0
DYRK3	8444	0	2.7161391588	0.1916018688	0.1951528235	-0.1449176879	0
EIF2AK4	440275	-0.7116005882	0	0	0	0	0
EPHA7	2045	0	0	0	0.1444749298	0	0
ERK8		0	1.5476795105	0.1090417486	0	-0.07133	0
ERN2	10595	0	0	0	0	-0.0806014946	0
ETNK1	55500	0	0	0	0.1185938224	-0.0683433953	0
FGFR2	2263	0	1.7174930582	0.1489351892	0.1252106513	-0.1155249481	0
FLJ25006		0	1.5335072739	0	0	0	0
FLT4	2324	-0.5479343571	2.2900936419	0.1352398454	0.1255391104	-0.1239510965	0
FN3K	64122	0	0	0	0.1115682121	0	0
GALK1	2584	0	3.0523974587	0.2441906943	0.2021668493	-0.1888352368	0
GCK	2645	-0.6752924375	0	0	0	-0.0543672622	0
GLYCTK	132158	-0.5697323529	0	0	0.1253687942	0	0
HCV_138		-2.6394967123	3.6479370634	0.2521418998	0.2548755564	-0.2008731061	0
HCV_321		-2.9216648356	3.6368455651	0.261232309	0.2909695944	-0.1994508322	0.1469633834
HIPK4	147746	0	0	0	0.1207865579	0	0
HK2	3099	0	0	0	0.1038299172	0	0
HK3	3101	0	0	0	0	-0.0742196973	0
IHPK2		-0.6379886667	0	0	0	0	0
IRAK1	3654	0	1.571171456	0	0.1500652951	-0.0779952425	0
LAK		-0.8162185882	1.6598044635	0	0	0	0
LIMK2	3985	0	2.3567623916	0.1509813226	0.1430903169	-0.1020785759	0
LOC340156		0	0	0	0	-0.0596646094	0
LOC390777		0	0	0.1025662816	0.1267588355	-0.0631485659	0
LOC391533	391533	0	0	0	0.1497439457	-0.0696283635	0
LOC400301		0	0	0	0.1434720999	-0.0900677638	0
LOC442141		0	0	0.1171384091	0.1305503104	-0.0902587809	0
LOC91807		0	1.6231886166	0.1404126556	0.1347382198	-0.103733009	0
LRRK1	79705	0	2.2646624941	0.1475742145	0	-0.124122252	0
MAK	4117	0	2.0078318022	0.1523644514	0	-0.1309117181	0
MAP2K1IP1		0	1.5737926236	0.1444966778	0.1475532346	-0.0881662433	0
MAP2K2	407835	-0.6364224	0	0	0	0	0
MAP2K3	5606	0	0	0	0	-0.0819301918	0
MAP2K4	6416	0	0	0.1027432082	0	0	0
MAP3K14	9020	0	0	0	0	-0.0724596071	0
MAP3K6	9064	0	1.7366747557	0	0.1730559659	-0.1016838038	0
MAP3K7	6885	0	0	0	0.1246736635	0	0
MELK	9833	0	1.932617677	0.1322887231	0	-0.1321649334	0
MET	4233	-0.8306554118	1.5587167624	0	0.1152671985	0	0
MKNK1	8569	0	0	0	0.1177831324	-0.0701719605	0
MPP6	51678	-0.5960231875	0	0	0	0	0
MPP7	143098	0	0	0	0.1339683533	0	0
NME4	4833	0	2.1383455773	0.1451179753	0.1405009476	-0.1362921907	0
NME6	10201	0	0	0.1134694912	0	0	0
OSR1	130497	-0.5213102667	0	0	0	0	0
p24		-0.3435650385	0	0	0	-0.0791475374	0
PAK4	10298	-0.5306489583	0	0	0	0	0
PCM1	5108	0	2.1355771029	0.1228696339	0.1287111492	-0.1013362824	0
PCTK1		0	0	0.1185463341	0	-0.0796580007	0
PCTK3		0	0	0.1147505514	0	0	0
PDGFRA	5156	0	0	0	0.1415938911	0	0
PDK1	5163	0	0	0.116856564	0.1200262806	0	0
PDK4	5166	0	0	0	0	-0.0566570987	0
PDPK1	5170	0	0	0.1154572945	0	0	0
PFKP	5214	0	0	0	0	-0.0544526207	0
PIK3R4	30849	-0.97696	0	0	0	0	0
PIK4CA		-3.5110589091	2.9829469232	0.2030196698	0.227870097	-0.1603934059	0
PIK4CB		-0.6006052353	0	0	0	0	0
PKN2	5586	-0.6840730625	0	0	0	0	0
PRKCD	5580	0	0	0.1008470858	0.1182278733	-0.0623171886	0
PRKCDBP	112464	0	0	0	0	-0.0610411599	0
PRKCI	5584	-0.3953135	0	0	0	-0.0969059342	0.1312752438
PRKY	5616	0	0	0	0	-0.0561089	0
PRPS1	221823	0	2.2979784971	0.1523630494	0.1616320722	-0.1432502925	0.1916018688
PRPS2	5634	0	0	0	0	-0.0543121159	0
PSKH1	5681	0	2.8671762203	0.2149395892	0.2103765235	-0.1370060224	0
PTK9L	84084	0	0	0	0	-0.0624599794	0
RAB6A	5894	0	0	0.1221191189	0.1565963195	-0.0894259361	0
RAF1		0	0	0.1053950163	0	-0.0529807645	0
RPS6KB1	6198	-0.5494733333	0	0	0	0	0
SBK1	388228	0	0	0	0	-0.0682361747	0
SGKL		0	0	0.1357119648	0	0	0
SIK2	23235	0	0	0	0.1408657236	0	0
SLAMF6	114836	-0.4252741765	0	0	0	-0.0728310483	0
SRMS	6725	0	1.9378112685	0.1761706874	0.21965565	-0.1174763683	0
STK17A	9263	0	1.7576337841	0	0.1278444374	-0.0870925224	0
STK19	8859	0	0	0	0	-0.0710872644	0
STK24	8428	0	0	0	0	-0.0643778089	0
STK4	6789	0	0	0.1218911378	0	0	0
STK6		0	0	0	0.1193660567	0	0

Continued on next page

Table B.1 – continued from previous page

Gene	ENTREZ ID	RIPLEY	AVERAGE	CELL-BASED	GSEA-ONLY	MARS-ONLY	HIT DENV
TAF1	6872	0	1.7357011523	0.1163455498	0.1410310476	-0.0974076802	0
TEX14	56155	0	0	0	0	-0.0685526256	0
TGFBR2	7048	0	0	0	0	-0.0581756092	0
TIE		0	0	0	0	-0.0910392618	0
TLK1	9874	0	1.6030779342	0	0.1006021727	0	0
TXNDC3	51314	0	1.8099067532	0.1200612542	0.1414415843	-0.0891001732	0
TXNDC6	347736	0	1.5038896218	0	0.1222863847	-0.0665637725	0
ULK2	9706	0	0	0	0.1467522055	0	0
VRK3	51231	0	2.0428657479	0.1739949434	0.1652335831	-0.1170016487	0

B.2 Hit List of the DENV Screen

Table B.2 lists the hits of virus host dependency factors of the DENV screen after normalization and hit scoring using the CELL-BASED analysis method.

Table B.2: Gene hit list of the DENV screen after CELL-BASED analysis. Shown are *ES* after normalization.

Gene	ENTREZ ID	CELL-BASED
AAK1	22848	0.1209106731
ACVR2B	93	0.146984063
ADCK5	203054	0.2593055687
AKAP8L	26993	0.1131147699
C19orf35	374872	0.1674584813
C9orf96	169436	0.1838625724
CDC42SE2	56990	0.1337900039
CDK10	8558	0.1202148192
CDK4	1019	0.1190662535
CKI-alpha		0.1177560653
CKI-delta		0.1302087219
CKMT1		0.1208840862
COASY	80347	0.1643057047
COPB		0.1446878343
CSK	1445	0.1283304096
ERBB4	2066	0.1440386128
GK	2713	0.1314407357
HCV_138		0.1341174024
HCV_321		0.1547687568
IHPK3		0.1409453321
ITPK1	3705	0.1304454772
KDR	3791	0.1410455703
LCK	3932	0.1039074693
LEDGF-p75		0.133982579
LOC340371		0.1359490663
LRRK2	120892	0.1285685923
MAP2K7	5609	0.167248405
MAPK10	5602	0.1575260835
MAPK6	5597	0.1341785924
MAPK8	5599	0.1128646973
MAPKAPK2	9261	0.1319369824
MARK1	4139	0.1156667351
MAST1	22983	0.1313235059
MIDORI		0.1600694441

Continued on next page

Table B.2 – continued from previous page

Gene	ENTREZ ID	CELL-BASED
MINK		0.1329031123
NEK6	10783	0.1208132764
NTRK3	4916	0.102864854
PIK3R3	8503	0.1816203592
PKMYT1	9088	0.1093590518
PLXNA3	55558	0.1465929834
PLXNC1	10154	0.135352589
PRKACB	5567	0.1145926292
PRKCB1		0.1186531494
PRKCE	5581	0.1178679489
PRKCI	5584	0.1188402306
PRKCQ	5588	0.1209638557
PRKG2	5593	0.1376981852
PRKY	5616	0.1489939056
PRPS1	221823	0.1319423276
PTK2B	2185	0.1111078258
RAB1A	5861	0.1349586886
RIOK1	83732	0.1635074222
RIPK3	11035	0.1480503226
RPS6KL1	83694	0.2125937692
STK32C	282974	0.1301677447
TEC	7006	0.187453925
TRIB3	57761	0.1311717934

B.3 Pathway Analysis of HCV and DENV Hits

In Table B.3 and Table B.4 the results of the gene set enrichment analysis of the HCV screen for KEGG and Biocarta are listed and correspondingly Table B.5 and Table B.6 show the gene set enrichment analysis results of the DENV screen.

Table B.3: Results of gene set enrichment analysis of the HCV screen for KEGG and Biocarta pathways using the DAVID bioinformatics software on the hits derived from the CELL-BASED analysis. In the column “No.” results are enumerated to find the respective pathways in the continued Table B.4. Column “Category” shows whether the hit was found in the KEGG or Biocarta database. “Term” describes the name of the pathway. “Count” and “%” indicate how many genes and what percentage of the hit list are in the respective pathway. The column “Pvalue” shows the significance of the enrichment, “List” is the total number of genes in the corresponding pathway and “Fold Enrichment” expresses the enrichment score.

No.	Category	Term	Count	%	PValue	List Total	Fold Enrichment
1	KEGG PATHWAY	hsa05223:Non-small cell lung cancer	5	0.52	1.43E-4	27	17.44
2	KEGG PATHWAY	hsa04664:Fc epsilon RI signaling pathway	5	0.52	5.93E-4	27	12.07
3	KEGG PATHWAY	hsa05215:Prostate cancer	5	0.52	9.77E-4	27	10.58
4	KEGG PATHWAY	hsa05200:Pathways in cancer	8	0.84	9.84E-004	27	4.59
5	KEGG PATHWAY	hsa04666:Fc gamma R-mediated phagocytosis	5	0.52	1.25E-003	27	9.91
6	KEGG PATHWAY	hsa04010:MAPK signaling pathway	7	0.73	1.87E-003	27	4.94
7	KEGG PATHWAY	hsa04722:Neurotrophin signaling pathway	5	0.52	3.32E-003	27	7.59
8	KEGG PATHWAY	hsa05214:Glioma	4	0.42	3.85E-003	27	11.96
9	KEGG PATHWAY	hsa05212:Pancreatic cancer	4	0.42	0.01	27	10.46
10	KEGG PATHWAY	hsa05220:Chronic myeloid leukemia	4	0.42	0.01	27	10.04
11	KEGG PATHWAY	hsa04012:ErbB signaling pathway	4	0.42	0.01	27	8.66
12	KEGG PATHWAY	hsa04912:GnRH signaling pathway	4	0.42	0.01	27	7.69
13	BIOCARTA	h_RacCycDPathway:Influence of Ras and Rho proteins on G1 to S Transition Rho proteins on G1 to S Transition	3	0.31	0.02	13	13.82
14	KEGG PATHWAY	hsa04660:T cell receptor signaling pathway	4	0.42	0.02	27	6.98
15	BIOCARTA	h_pdgfPathway:PDGF Signaling Pathway	3	0.31	0.02	13	13.26
16	BIOCARTA	h_egfPathway:EGF Signaling Pathway	3	0.31	0.02	13	12.75
17	KEGG PATHWAY	hsa05213:Endometrial cancer	3	0.31	0.03	27	10.87
18	KEGG PATHWAY	hsa04530:Tight junction	4	0.42	0.03	27	5.62
19	KEGG PATHWAY	hsa00230:Purine metabolism	4	0.42	0.04	27	4.92
20	KEGG PATHWAY	hsa05218:Melanoma	3	0.31	0.05	27	7.96
21	KEGG PATHWAY	hsa04662:B cell receptor signaling pathway	3	0.31	0.06	27	7.53
22	KEGG PATHWAY	hsa04144:Endocytosis	4	0.42	0.07	27	4.09
23	KEGG PATHWAY	hsa05222:Small cell lung cancer	3	0.31	0.07	27	6.73
24	KEGG PATHWAY	hsa05210:Colorectal cancer	3	0.31	0.07	27	6.73
25	KEGG PATHWAY	hsa04062:Chemokine signaling pathway	4	0.42	0.07	27	4.03
26	KEGG PATHWAY	hsa04914:Progesterone-mediated oocyte maturation	3	0.31	0.07	27	6.57
27	KEGG PATHWAY	hsa04510:Focal adhesion	4	0.42	0.08	27	3.75
28	KEGG PATHWAY	hsa04810:Regulation of actin cytoskeleton	4	0.42	0.09	27	3.50

Continued on next page

Table B.3 – continued from previous page

No.	Category	Term	Count	%	PValue	List Total	Fold Enrichment
29	BIOCARTA	h_rbPathway:RB Tumor Suppressor/Checkpoint Signaling in response to DNA damage	2	0.21	0.10	13	18.42

Table B.4: Results of gene set enrichment analysis of the HCV screen for KEGG and Biocarta on the CELL-BASED hit list, continued from Table B.3 (see column "No." to find the corresponding pathways). The columns "Bonferroni", "Benjamini" and "FDR" show the p-value after the respective correction for multiple testing and column "Genes" lists the hit genes found in each pathway.

No.	Bonferroni	Benjamini	FDR	Genes
1	0.01	0.01	0.15	PDPK1, RAF1, CDK4, STK4, AKT3
2	0.04	0.02	0.60	PDK1, MAP2K4, RAF1, PRKCD, AKT3
3	0.06	0.02	0.99	FGFR2, PDPK1, RAF1, AKT3, CDK2
4	0.06	0.02	1.00	FGFR2, RAF1, CDK4, STK4, AKT3, CDK2, CSF1R, ACVR1C
5	0.08	0.02	1.26	LIMK2, CFL1, RAF1, PRKCD, AKT3
6	0.11	0.02	1.88	FGFR2, MAP3K6, MAP2K4, RAF1, STK4, AKT3, ACVR1C
7	0.19	0.03	3.32	PDK1, RAF1, CAMK2B, PRKCD, AKT3
8	0.22	0.03	3.84	RAF1, CAMK2B, CDK4, AKT3
9	0.30	0.04	5.55	RAF1, CDK4, AKT3, ACVR1C
10	0.33	0.04	6.20	RAF1, CDK4, AKT3, ACVR1C
11	0.46	0.05	9.21	MAP2K4, RAF1, CAMK2B, AKT3
12	0.57	0.07	12.53	MAP2K4, RAF1, CAMK2B, PRKCD
13	0.70	0.70	15.48	RAF1, CDK4, CDK2
14	0.67	0.08	15.97	PDK1, RAF1, CDK4, AKT3
15	0.72	0.47	16.64	CSNK2A1, MAP2K4, RAF1
16	0.75	0.37	17.83	CSNK2A1, MAP2K4, RAF1
17	0.84	0.12	25.46	PDPK1, RAF1, AKT3
18	0.86	0.12	26.53	CSNK2A1, CDK4, PRKCD, AKT3
19	0.94	0.16	35.23	NME4, ADK, PRPS1, NME6
20	0.96	0.18	40.84	RAF1, CDK4, AKT3
21	0.97	0.18	44.04	CD81, RAF1, AKT3
22	0.99	0.21	49.94	FGFR2, CXCR4, CSF1R, ACVR1C
23	0.99	0.20	51.03	CDK4, AKT3, CDK2
24	0.99	0.20	51.03	RAF1, AKT3, ACVR1C
25	0.99	0.19	51.34	CXCR4, RAF1, PRKCD, AKT3
26	0.99	0.19	52.53	RAF1, AKT3, CDK2
27	1.00	0.21	57.71	PDPK1, FLT4, RAF1, AKT3
28	1.00	0.23	63.71	FGFR2, LIMK2, CFL1, RAF1
29	1.00	0.85	65.26	CDK4, CDK2

Table B.5: Results of gene set enrichment analysis of the DENV screen for KEGG and Biocarta pathways using the DAVID bioinformatics software. The column "No." enumerates results like in the continued Table B.6. Column "Category" shows whether the KEGG or Biocarta database was used. "Term" describes the pathway name. "Count" and "%" indicate how many genes and what percentage of the hit list are in the respective pathway. The column "Pvalue" shows the significance of the enrichment, "List" is the total number of genes per pathway and "Fold Enrichment" expresses the enrichment score.

No.	Category	Term	Count	%	PValue	List Total	Fold Enrichment
1	KEGG PATHWAY	hsa04722:Neurotrophin signaling pathway	7	15.22	3.62E-5	28	10.25
2	KEGG PATHWAY	hsa04664:Fc epsilon RI signaling pathway	5	10.87	6.88E-4	28	11.64
3	KEGG PATHWAY	hsa04914:Progesterone-mediated oocyte maturation	5	10.87	9.95E-4	28	10.56
4	KEGG PATHWAY	hsa04660:T cell receptor signaling pathway	6	13.04	2.19E-4	28	10.09
5	KEGG PATHWAY	hsa04012:ErbB signaling pathway	5	10.87	1.04E-003	28	10.44
6	KEGG PATHWAY	hsa04912:GnRH signaling pathway	5	10.87	1.62E-003	28	9.27
7	KEGG PATHWAY	hsa04930:Type II diabetes mellitus	4	8.70	1.85E-003	28	15.46
8	KEGG PATHWAY	hsa04910:Insulin signaling pathway	5	10.87	5.19E-003	28	6.73
9	BIOCARTA	h_mapkPathway:MAPKinase Signaling Pathway	5	0.95	5.82E-003	15	5.99
10	KEGG PATHWAY	hsa05212:Pancreatic cancer	4	8.70	6.24E-003	28	10.09
11	KEGG PATHWAY	hsa04620:Toll-like receptor signaling pathway	4	8.70	1.57E-002	28	7.19
12	KEGG PATHWAY	hsa04530:Tight junction	4	8.70	3.30E-002	28	5.42
13	KEGG PATHWAY	hsa04920:Adipocytokine signaling pathway	3	6.52	4.86E-002	28	8.13
14	KEGG PATHWAY	hsa05120:Epithelial cell signaling in Helicobacter pylori infection	3	6.52	4.99E-002	28	8.01
15	KEGG PATHWAY	hsa04010:MAPK signaling pathway	5	10.87	5.03E-002	28	3.40
16	BIOCARTA	h_keratinocytePathway:Keratinocyte Differentiation	3	0.57	5.07E-002	15	7.56
17	KEGG PATHWAY	hsa04370:VEGF signaling pathway	3	6.52	5.94E-002	28	7.26
18	KEGG PATHWAY	hsa05210:Colorectal cancer	3	6.52	7.25E-002	28	6.49
19	KEGG PATHWAY	hsa04062:Chemokine signaling pathway	4	8.70	7.50E-002	28	3.88
20	KEGG PATHWAY	hsa04510:Focal adhesion	4	8.70	8.88E-002	28	3.61

Table B.6: Results of gene set enrichment analysis of the DENV screen for KEGG and Biocarta, continued from Table B.6 (see column "No." to find the corresponding pathways). The columns "Bonferroni", "Benjamini" and "FDR" show the p-value after the respective correction for multiple testing and column "Genes" lists the hit genes found in each pathway.

No.	Bonferroni	Benjamini	FDR	Genes
1	0.003	0.003	0.037	NTRK3, MAPK8, MAPK10, MAPKAPK2, PIK3R3, CSK, MAP2K7
2	0.047	0.016	0.710	MAPK8, MAPK10, PIK3R3, PRKCE, MAP2K7
3	0.067	0.017	1.025	PKMYT1, MAPK8, MAPK10, PRKACB, PIK3R3
4	0.015	0.008	0.226	PRKCQ, LCK, PIK3R3, CDK4, MAP2K7, TEC
5	0.070	0.014	1.070	ERBB4, MAPK8, MAPK10, PIK3R3, MAP2K7
6	0.107	0.019	1.664	PTK2B, MAPK8, MAPK10, PRKACB, MAP2K7
7	0.122	0.018	1.901	MAPK8, MAPK10, PIK3R3, PRKCE
8	0.305	0.044	5.241	PRKCI, MAPK8, MAPK10, PRKACB, PIK3R3
9	0.316	0.316	5.771	MAPK6, MAPK8, MAPK10, MAPKAPK2, MAP2K7
10	0.355	0.048	6.273	MAPK8, MAPK10, PIK3R3, CDK4
11	0.671	0.105	15.146	MAPK8, MAPK10, PIK3R3, MAP2K7
12	0.904	0.192	29.334	PRKCQ, PRKCI, CDK4, PRKCE
13	0.969	0.252	40.278	PRKCQ, MAPK8, MAPK10
14	0.972	0.241	41.125	MAPK8, MAPK10, CSK
15	0.973	0.228	41.399	MAPK8, MAPK10, MAPKAPK2, PRKACB, MAP2K7
16	0.966	0.816	41.180	PRKCQ, MAPK8, MAP2K7
17	0.986	0.249	46.959	MAPKAPK2, PIK3R3, KDR
18	0.995	0.281	54.125	MAPK8, MAPK10, PIK3R3
19	0.996	0.275	55.377	PTK2B, PRKACB, PIK3R3, CSK
20	0.999	0.304	61.814	MAPK8, MAPK10, PIK3R3, KDR

The Tables B.7 and B.8 show the enriched KEGG and Biocarta pathways of the analysis strategy AVERAGE.

Table B.7: Results of gene set enrichment analysis of the HCV screen for KEGG and Biocarta pathways using the DAVID bioinformatics software on the hits derived from the AVERAGE analysis. In the column "No." results are enumerated to find the respective pathways in the continued Table B.8. Column "Category" shows whether the hit was found in the KEGG or Biocarta database. "Term" describes the name of the pathway. "Count" and "%" indicate how many genes and what percentage of the hit list are in the respective pathway. The column "Pvalue" shows the significance of the enrichment, "List" is the total number of genes in the corresponding pathway and "Fold Enrichment" expresses the enrichment score.

No.	Category	Term	Count	%	PValue	List Total
1	KEGG PATHWAY	hsa04360:Axon guidance	3	0.345	0.061	17
2	KEGG PATHWAY	hsa00230:Purine metabolism	3	0.345	0.082	17

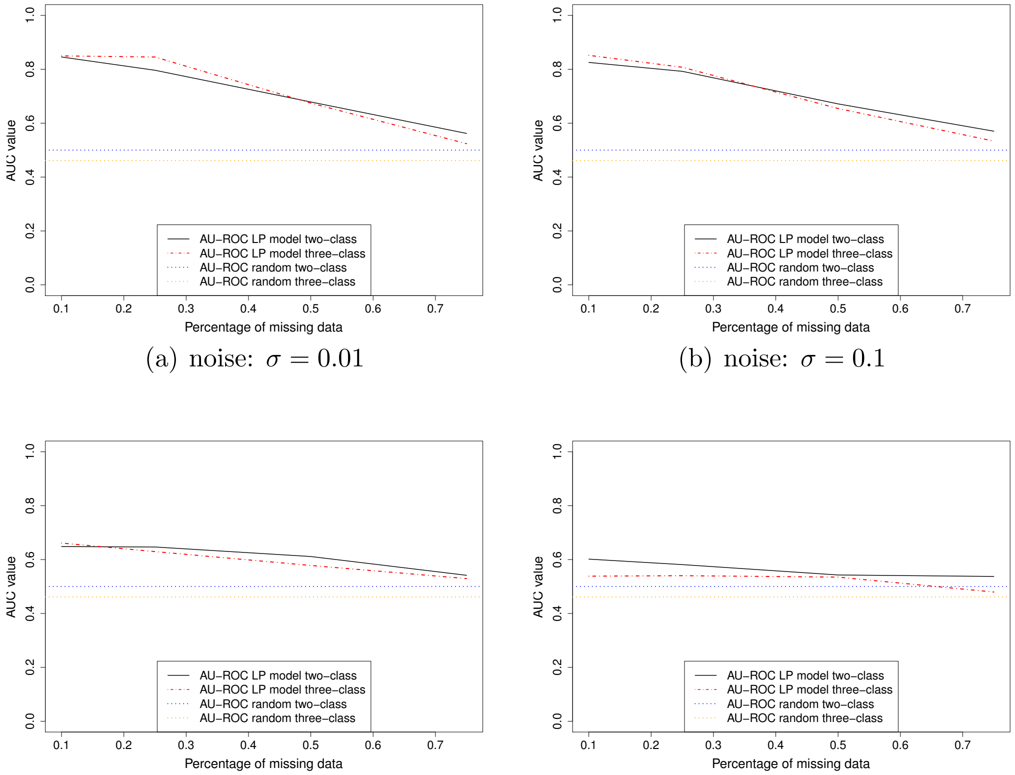
Table B.8: Results of gene set enrichment analysis of the HCV screen for KEGG and Biocarta on the AVERAGE hit list, continued from Table B.7 (see column "No." to find the corresponding pathways). The columns "Bonferroni", "Benjamini" and "FDR" show the p-value after the respective correction for multiple testing and column "Genes" lists the hit genes found in each pathway.

No.	Fold Enrichment	Bonferroni	Benjamini	FDR	Genes
1	6.956	0.913	0.913	43.456	LIMK2, MET, CFL1
2	5.865	0.964	0.811	54.046	NME4, ADK, PRPS1

Appendix C

Two-Class and Three-Class AUC Values

To assess, whether the LP model can infer the sign of an edge with a good performance, we use a three-class classification to compute AU-ROC and AU-PR values for the data simulated with missing data points. The results are represented in Figure C.1 and C.2. Clearly, the performance of the network inference with the LP model for the two-class classification is similar to the performance measured with a three-class classification.



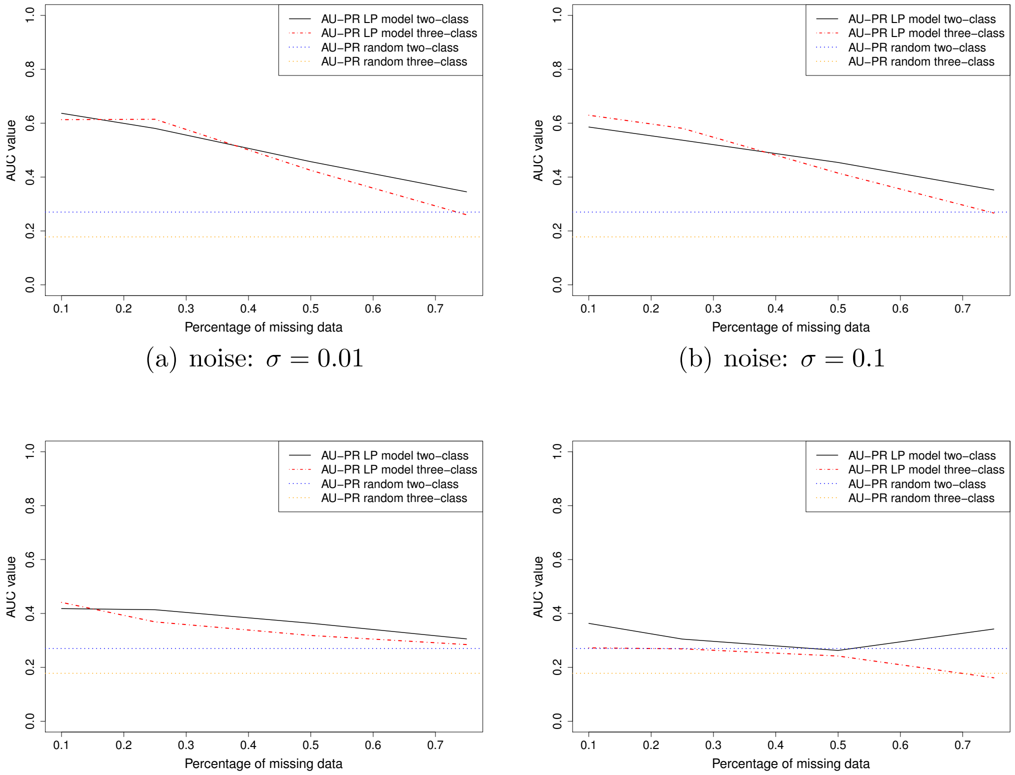
(a) noise: $\sigma = 0.01$

(b) noise: $\sigma = 0.1$

(c) noise: $\sigma = 0.27$

(d) noise: $\sigma = 0.5$

Figure C.1: Mean area under the two-class and the three-class ROC-curve for random guessing as well as for results of the network inference using the LP model on missing data over 100 repetitions of four different noise settings. (a) noise: $\sigma = 0.01$ (b) noise: $\sigma = 0.1$ (c) noise: $\sigma = 0.27$ (d) noise: $\sigma = 0.5$. The dotted lines represent random guessing values for the two- and three-class classification. The red (dash-dotted) lines, respectively, the straight (black) lines correspond to results of the three-class, respectively, the two-class classification analysis of the results derived with the LP model.



(a) noise: $\sigma = 0.01$

(b) noise: $\sigma = 0.1$

(c) noise: $\sigma = 0.27$

(d) noise: $\sigma = 0.5$

Figure C.2: Mean area under the two-class and the three-class PR-curve for random guessing as well as for results of the network inference using the LP model on missing data over 100 repetitions of four different noise settings. (a) noise: $\sigma = 0.01$ (b) noise: $\sigma = 0.1$ (c) noise: $\sigma = 0.27$ (d) noise: $\sigma = 0.5$. The dotted lines represent random guessing values for the two- and three-class classification. The red (dash-dotted) lines, respectively, the straight (black) lines correspond to results of the three-class, respectively, the two-class classification analysis of the results derived with the LP model.

Appendix D

ErbB Signaling Network Inferred with the LP model

In Table D.1 the median edge weights ± the median absolute deviation (MAD) of all LOOCV-steps for the inferred network topologies using the LP model on the ErbB signaling data are represented.

Table D.1: Median±MAD of inferred edge weights w_{ij} using the LP model on the ErbB signaling data. Here, i corresponds to genes of the ith row and j to genes of the jth column. If the MAD is not given explicitly, it is equal to zero.

	CDK2	CDK4	CDK6	Cyclin D1	Cyclin E1	ERalpha	ERBB1	ERBB2	ERBB3	IGF1R	MYC	p21	p27	pAKT1	pERK1/2	pRB1
CDK2	0	0	0	0	0	0	0	0	0	0	0	-0.14±0.21	0	0.25±0.37	0	0.33±0.27
CDK4	0	0	0	0	0	0	0.4±0.05	0.69	0	0	0	0	0	0	0	0.44±0.18
CDK6	0	0	0	0	0	0.95	0	0	0	0	0	0	0	0	0	0
Cyclin D1	-0.36±0.05	0	0	0	0	0	0	0	0	0	0	0.39±0.02	0.59±0.32	0.16±0.06	0.18±0.03	0.01±0.02
Cyclin E1	0	0	0	0	0	0	0	0	0	0.95	0	0	0	0	0	0
ERalpha	0	0	0	0	0	0	0	0	0	0	0.95	0	0	0	0	0
ERBB1	0	0	0	0	0	0	0	0.21	0	0	0	-0.2±0.07	0.36	0.03±0.05	0	-0.04±0.06
ERBB2	1±0.07	0.21±0.05	0	0	0	0	0.42±0.06	0	0	0	0	0	0	0	0	-0.02±0.03
ERBB3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IGF1R	0	0	0	0	0.95	0	0	0	0	0	0	0	0	0	0	0
MYC	0	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0
p21	-1.07±0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	1.62±0.03	0
p27	0	0.25	0	0.7	0	0	0	0	0	0	0	0	0	0	0	0
pAKT1	0	0	0	0.26	0	0	0	0	0	0	0	0	-0.2±0.3	0	0	0.31±0.15
pERK1.2	0±0.01	0	0	0	0	0	-0.02±0.02	0	0	0	0	0.72±0.14	0	0	0	0
pRB1	0.45±0.11	0.67±0.05	0	0.05	0	0	0.06	0	0	0	0	0	0.06±0.08	0.35±0.19	-0.6±0.06	0

Appendix E

Publications

The work presented in this thesis has been partially published in the following journals, conferences and workshops:

1. Knapp, B., Kaderali L. (2008). *Inferring Gene-Regulatory Networks from High-Throughput RNAi Screening Data.* Poster presentation at the 1st ViroQuant retreat, Asselheim/Grünstadt, April 21-22, 2008.

2. Rieber, N., Knapp, B., Kaderali L. (2008). *An automated Pipeline for the Statistical Analysis of High-Throughput RNAi Screens.* Poster presentation at the 1st ViroQuant retreat, Asselheim/Grünstadt, April 21-22, 2008.

3. Knapp, B., Rieber, N., Kumar, A., Matula, P., Erfle, H., Pepperkok, R., Rohr, K., Eils, R., Bartenschlager, R., Kaderali L. (2008). *A Statistics Pipeline for the Analysis of RNAi Knockout Data.* Statistical Computing 2008, June 1-4, Schloss Reisensburg (Günzburg).

4. Rieber, N., Knapp, B., Eils, R., Kaderali, L. (2009) *RNAither, an Automated Pipeline for the Statistical Analysis of High-Throughput RNAi Screens.* Bioinformatics, 25, 678-679.

5. Knapp, B., Mazur, J., Kaderali L. (2009). *Inferring Networks from High-Throughput Data.* Poster presentation at the 2nd ViroQuant retreat, Asselheim/Grünstadt, February 16-17, 2009.

6. Rieber, N., Knapp, B., Eils, R., Kaderali L. (2009). *RNAither, a Statistics Pipeline for the Automated Analysis of High-throughput RNAi Knockout Data.* Poster presentation at the 2nd ViroQuant retreat, Asselheim/Grünstadt, February 16-17, 2009.

7. Knapp, B., Dazert, E., Bartenschlager, R., Kaderali L. (2009). *Network inference using RNAi Knockdown Data and Biological Prior Knowledge.*

Poster presentation at the German Symposium on Systems Biology, Heidelberg, May 12-15, 2009.

8. Kiani, N. A., Knapp, B., Wörz, I., Bartenschlager, R., Kaderali L. (2010). *Analysis of RNA Interference Screens at the Level of Single Cell.* Poster presentation at the Conference on Systems Biology of Mammalian Cells, Freiburg, June 3-5, 2010.

9. Knapp, B., Kaderali, L. (2011). *Linear Model for Network Inference using RNA interference Data.* Fifth International Workshop on Machine Learning in Systems Biology, MSLB 2011, July 20-21, Vienna, Austria.

10. Knapp, B., Rebhan, I., Kumar, A., Matula, P., Kiani, N.A., Binder, M., Erfle, H., Rohr, K., Eils, R., Bartenschlager, R., Kaderali, L. (2011). *Normalization and Hit Scoring of High-Throughput, High-Content RNAi Data using Individual Cell Measurements.* Poster presentation at the International Conference on Systems Biology, Heidelberg, August 28 - September 1, 2011.

11. Knapp, B., Rebhan, I., Kumar, A., Matula, P., Kiani, N.A., Binder, M., Erfle, H., Rohr, K., Eils, R., Bartenschlager, R., Kaderali, L. (2011). *RNAi Screening Data Analysis using Individual Cell Measurements of High-Throughput, High-Content Screens.* Workshop statistical and dynamical models in biology and medicine, Göttingen, October 27- 28, 2011.

12. Knapp, B., Rebhan, I., Kumar, A., Matula, P., Kiani, N.A., Binder, M., Erfle, H., Rohr, K., Eils, R., Bartenschlager, R., Kaderali, L. (2011). *Normalizing for Individual Cell Population Context in the Analysis of High-Content Cellular Screens.* Submitted to BMC Bioinformatics and currently under revision.

13. Knapp, B., Kaderali, L. (2011). *Network Inference of Perturbation Data using a Linear Programming Approach.* In preparation for the sumbmission to Bioinformatics.

Notation Index and Abbreviations

μ	Mean, 18
σ	Standard Deviation, 18
I_m	$m \times m$ Identity Matrix, 91
AIC	Akaike information criterion, 77
ASGP-R	Asialoglycoprotein Receptor, 11
AUC	Area Under Curve, 59
CSNK2A1	Casein Kinase II Subunit Alpha, 57
CV	Coefficient of Variation, 18
D-NEM	Dynamic Nested Effects Model, 78
DAG	Directed Acyclic Graph, 71
DAPI	4',6-Diamidino-2-Phenylindole, 16
DAVID	Database for Annotation, Visualization and Integrated Discovery, 64
DDEPN	Dynamic Deterministic Effects Propagation Network, 84
DENV	Dengue Fever Virus, 10
DEPN	Deterministic Effects Propagation Network, 82
DNA	Deoxyribonucleic Acid, 8
DR	Dynamic Range, 17
dsRNA	Double-stranded RNA, 6
EGF	Epidermal Growth Factor, 116
EGFR	Epidermal Growth Receptor, 116
EM	Expectation-Maximization, 38
ER	Endoplasmic Reticulum, 10
ES	Enrichment Score, 53
esiRNAs	Endoribonuclease-prepared Short Interfering RNA, 13
FLT-4	Fms-Related Tyrosine Kinase 4, 57
GCV	Generalized Cross-Validation, 49
GFP	Green Fluorescence Protein, 14
GO	Gene Ontology, 72
GSEA	Gene Set Enrichment Analysis, 52
HCV	Hepatitis C Virus, 11

HDF	Host Dependency Factors, 51
JAK/STAT	Janus Kinases and Signal Transducers and Activators of Transcription, 81
kb	Kilobases, 10
KEGG	Kyoto Encyclopedia of Genes and Genomes, 73
LDL-R	Low-density Lipoprotein Receptor, 11
LOOCV	Leave-One-Out Cross-Validation, 97
Lowess	Locally Weighted Polynomial Regression, 20
LP	linear programming, 86
MAD	Median Absolute Deviation, 19
MARS	Multivariate Adaptive Regression Splines, 48
mRNA	Messenger RNA, 7
MSE	mean squared error, 97
NEM	Nested Effects Model, 74
nm	Nanometer, 9
nt	Nucleotides, 7
PCC	Pearson correlation coefficient, 31
PCR	Polymerase Chain Reaction, 8
PI4KA	Phosphatidylinositol 4-Kinase Alpha, 57
PPI	Protein-Protein Interaction, 72
PR	Precision to Recall, 100
RISC	RNA-induced Silencing Complex, 7
RNA	Ribonucleic Acid, 6
RNAi	RNA Interference, 1
ROC	Receiver Operator Characteristic, 59
RPPA	Reverse Phase Protein Array, 82
RPPA	Reverse Phase Protein Arrays, 117
RSS	Residual Sum of Squares, 49
RT-PCR	Reverse Transcriptase-polymerase Chain Reaction, 14
SA	Simulated Annealing, 78
shRNAs	Short Hairpin RNAs, 8
SID-1	Systemic RNA Interference Defective, 7
siRNAs	Short Interfering RNAs, 6
SR-BI	Scavenger Receptor B Type I, 11
TGN	*Trans*-Golgi Network, 10

Index

Bibliography

[6.711] DAVID Bioinformatics Resources 6.7. The database for annotation, visualization and integrated discovery (david) v6.7. `http://david.abcc.ncifcrf.gov/`, September 2011.

[AHP$^+$10] S. A. Angaji, S. S. Hedayati, R. H. Poor, S. Madani, S. S. Poor, and S. Panahi. Application of RNA interference in treating human diseases. *J. Genet.*, 89:527–537, Dec 2010.

[AKB04] D. J. Allocco, I. S. Kohane, and A. J. Butte. Quantifying the relationship between co-expression, co-regulation and gene function. *BMC Bioinformatics*, 5:18, Feb 2004.

[Alb05] R. Albert. Scale-free networks in cell biology. *J. Cell. Sci.*, 118:4947–4957, Nov 2005.

[AS09] M. Ackermann and K. Strimmer. A general modular framework for gene set enrichment analysis. *BMC Bioinformatics*, 10:47, 2009.

[Ash00] M. Ashburner. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[ASJ$^+$09] B. Anchang, M. J. Sadeh, J. Jacob, A. Tresch, M. O. Vlad, P. J. Oefner, and R. Spang. Modeling the temporal interplay of molecular signaling and gene expression by using dynamic nested effects models. *Proc. Natl. Acad. Sci. U.S.A.*, 106:6447–6452, Apr 2009.

[BA97] Adrian W. Bowman and Adelchi Azzalini. *Applied smoothing techniques for data analysis.* Number 18 in Oxford statistical science series. Clarendon Press, Oxford, 1997.

[BBH06] M. Boutros, L. P. Bras, and W. Huber. Analysis of cell-based RNAi screens. *Genome Biol.*, 7:R66, 2006.

[BBM$^+$94] S. Bates, L. Bonetta, D. MacAllan, D. Parry, A. Holder, C. Dickson, and G. Peters. CDK6 (PLSTIRE) and CDK4 (PSK-J3) are a distinct subset of the cyclin-dependent kinases that associate with cyclin D1. *Oncogene*, 9:71–79, Jan 1994.

[BCH$^+$09] K. L. Berger, J. D. Cooper, N. S. Heaton, R. Yoon, T. E. Oakland, T. X. Jordan, G. Mateu, A. Grakoui, and G. Randall. Roles for endocytic trafficking and phosphatidylinositol 4-kinase III alpha in hepatitis C virus replication. *Proc. Natl. Acad. Sci. U.S.A.*, 106:7577–7582, May 2009.

[BDB$^+$08] A. L. Brass, D. M. Dykxhoorn, Y. Benita, N. Yan, A. Engelman, R. J. Xavier, J. Lieberman, and S. J. Elledge. Identification of host proteins required for HIV infection through a functional genomic screen. *Science*, 319:921–926, Feb 2008.

[BEWS04] S. D. Buckingham, B. Esmaeili, M. Wood, and D. B. Sattelle. RNA interference: from model organisms towards therapy for neural and neuromuscular disorders. *Hum. Mol. Genet.*, 13 Spec No 2:R275–288, Oct 2004.

[BFP04] R. Bartenschlager, M. Frese, and T. Pietschmann. Novel insights into hepatitis C virus replication and persistence. *Adv. Virus Res.*, 63:71–180, 2004.

[BH10] M. Boettcher and J. D. Hoheisel. Pooled RNAi Screens - Technical and Biological Aspects. *Curr. Genomics*, 11:162–167, May 2010.

[BHF$^+$10] C. Bender, F. Henjes, H. Frohlich, S. Wiemann, U. Korf, and T. Beissbarth. Dynamic deterministic effects propagation networks: learning signalling pathways from longitudinal protein array data. *Bioinformatics*, 26:596–602, Sep 2010.

[BHS$^+$10] K. Börner, J. Hermle, C. Sommer, N. P. Brown, B. Knapp, B. Glass, J. Kunkel, G. Torralba, J. Reymann, N. Beil, J. Beneke, R. Pepperkok, R. Schneider, T. Ludwig, M. Hausmann, F. Hamprecht, H. Erfle, L. Kaderali, H. G. Kräusslich, and M. J. Lehmann. From experimental setup to bioinformatics: an RNAi screening platform to identify host factors involved in HIV-1 replication. *Biotechnol J*, 5:39–49, Jan 2010.

[Bio11a] BioCarta. Biocarta. `http://www.biocarta.com/`, September 2011.

[Bio11b] Bioconductor. Bioconductor open source software for bioinformatics 2.9. `http://www.bioconductor.org/`, October 2011.

[Bis07] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, New York, 2007.

[BMW00] A. J. Baddeley, J. Mller, and R. Waagepetersen. Non- and semiparametric estimation of interaction in inhomogeneous point patterns. *Statistica Neerlandica*, 54(3):329–350, 2000.

[Bo10] Michel Berkelaar and others. *lpSolve: Interface to Lp_solve v. 5.5 to solve linear/integer programs*, 2010. R package version 5.6.5.

[Bor82] K. H. Borgwardt. The average number of pivot steps required by the simplex-method is polynomial. *Mathematical Methods of Operations Research*, 26:157–177, 1982. 10.1007/BF01917108.

[BSF$^+$09] A. Birmingham, L. M. Selfors, T. Forster, D. Wrobel, C. J. Kennedy, E. Shanks, J. Santoyo-Lopez, D. J. Dunican, A. Long, D. Kelleher, Q. Smith, R. L. Beijersbergen, P. Ghazal, and C. E. Shamu. Statistical methods for analysis of high-throughput RNA interference screens. *Nat. Methods*, 6:569–575, Aug 2009.

[BTP$^+$09] J. Borawski, P. Troke, X. Puyang, V. Gibaja, S. Zhao, C. Mickanin, J. Leighton-Davies, C. J. Wilson, V. Myer, I. Cornellataracido, J. Baryza, J. Tallarico, G. Joberty, M. Bantscheff, M. Schirle, T. Bouwmeester, J. E. Mathy, K. Lin, T. Compton, M. Labow, B. Wiedmann, and L. A. Gaither. Class III phosphatidylinositol 4-kinase alpha and beta are novel host factor regulators of hepatitis C virus replication. *J. Virol.*, 83:10058–10074, Oct 2009.

[BTS$^+$00] A. J. Butte, P. Tamayo, D. Slonim, T. R. Golub, and I. S. Kohane. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc. Natl. Acad. Sci. U.S.A.*, 97:12182–12186, Oct 2000.

[BY07] Erez M Bublil and Yosef Yarden. The egf receptor family: spearheading a merger of signaling and therapeutics. *Current Opinion in Cell Biology*, 19(2):124 – 134, 2007. Cell regulation.

[CBGB04] S. L. Carter, C. M. Brechbuhler, M. Griffin, and A. T. Bond. Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics*, 20:2242–2250, Sep 2004.

[Che09] S. Cherry. What have RNAi screens taught us about viral-host interactions? *Curr. Opin. Microbiol.*, 12:446–452, Aug 2009.

[CHI10] H. Y. Chuang, M. Hofree, and T. Ideker. A decade of systems biology. *Annu. Rev. Cell Dev. Biol.*, 26:721–744, Nov 2010.

[Cle79] W. S. Cleveland. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[Cle81] W. S. Cleveland. LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *The American Statistician*, 35:54, 1981.

[CLH$^+$08] N. Chung, L. Locco, K. W. Huff, S. Bartz, P. S. Linsley, M. Ferrer, and B. Strulovici. An efficient and fully automated high-throughput transfection method for genome-scale siRNA screens. *J Biomol Screen*, 13:142–148, Feb 2008.

[CS09] R. W. Carthew and E. J. Sontheimer. Origins and Mechanisms of miRNAs and siRNAs. *Cell*, 136:642–655, Feb 2009.

[CSJ$^+$09] W. W. Chen, B. Schoeberl, P. J. Jasper, M. Niepel, U. B. Nielsen, D. A. Lauffenburger, and P. K. Sorger. Input-output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data. *Mol. Syst. Biol.*, 5:239, 2009.

[CTS$^+$02] L. Charboneau, H. Tory, H. Scott, T. Chen, M. Winters, E. F. Petricoin, L. A. Liotta, and C. P. Paweletz. Utility of reverse phase protein arrays: applications to signalling pathways and human body arrays. *Brief Funct Genomic Proteomic*, 1:305–315, Oct 2002.

[CY06] A. Citri and Y. Yarden. EGF-ERBB signalling: towards the systems level. *Nat. Rev. Mol. Cell Biol.*, 7:505–516, Jul 2006.

[CZK$^+$08] N. Chung, X. D. Zhang, A. Kreamer, L. Locco, P. F. Kuan, S. Bartz, P. S. Linsley, M. Ferrer, and B. Strulovici. Median absolute deviation to improve hit selection for genome-scale RNAi screens. *J Biomol Screen*, 13:149–158, Feb 2008.

[Dan51] G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. Wiley, 1951.

[DOW55] George B. Dantzig, Alex Orden, and Philip Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific J. Math.*, 5:183–195, 1955.

[DS09] T. F. Duchaine and F. J. Slack. rna interference and micro rna -oriented therapy in cancer: rationales, promises, and challenges. *Curr Oncol*, 16:61–66, Aug 2009.

[EHL$^+$01] S. M. Elbashir, J. Harborth, W. Lendeckel, A. Yalcin, K. Weber, and T. Tuschl. Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells. *Nature*, 411:494–498, May 2001.

[Eis06] M. Eisenstein. Microarrays: quality control. *Nature*, 442:1067–1070, Aug 2006.

[ENL$^+$07] H. Erfle, B. Neumann, U. Liebel, P. Rogers, M. Held, T. Walter, J. Ellenberg, and R. Pepperkok. Reverse transfection on cell arrays for high content screening microscopy. *Nat Protoc*, 2:392–399, 2007.

[ENR$^+$08] H. Erfle, B. Neumann, P. Rogers, J. Bulkescher, J. Ellenberg, and R. Pepperkok. Work flow for multiplexing siRNA assays by solid-phase reverse transfection in multiwell plates. *J Biomol Screen*, 13:575–580, Aug 2008.

[EP07] H. Erfle and R. Pepperkok. Production of siRNA- and cDNA-transfected cell arrays on noncoated chambered coverglass for high-content screening microscopy in living cells. *Methods Mol. Biol.*, 360:155–161, 2007.

[Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition.

[FFS$^+$07] H. Froehlich, M. Fellmann, H. Sueltmann, A. Poustka, and T. Beissbarth. Large scale statistical inference of signaling pathways from RNAi and microarray data. *BMC Bioinformatics*, 8:386, 2007.

[FH03] E. H. Feinberg and C. P. Hunter. Transport of dsRNA into cells by the transmembrane protein SID-1. *Science*, 301:1545–1547, Sep 2003.

[FMT$^+$] Holger Froehlich, Florian Markowetz, Achim Tresch, Christian Bender, Matthias Maneck, Claudio Lottaz, and Tim Beissbarth.

nem: Nested Effects Models to reconstruct phenotypic hierarchies. R package version 2.10.0.

[FPK+10] F. Fuchs, G. Pau, D. Kranz, O. Sklyar, C. Budjan, S. Steinbrink, T. Horn, A. Pedal, W. Huber, and M. Boutros. Clustering phenotype populations by genome-wide RNAi and multiparametric imaging. *Mol. Syst. Biol.*, 6:370, Jun 2010.

[FPT11] Holger Fröhlich, Paurush Praveen, and Achim Tresch. Fast and efficient dynamic nested effects models. *Bioinformatics*, 27(2):238–244, 2011.

[FR95] J. H. Friedman and C. B. Roosen. An introduction to multivariate adaptive regression splines. *Stat Methods Med Res*, 4:197–217, Sep 1995.

[FR02] Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[FR07] C. Fraley and A. Raftery. Model-based methods of classification: Using the mclust software in chemometrics. *Journal of Statistical Software*, 18(6):1–13, January 2007.

[Fri91] Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–141, 1991.

[FSA+09] H. Froehlich, O. Sahin, D. Arlt, C. Bender, and T. Beissbarth. Deterministic Effects Propagation Networks for reconstructing protein signaling networks from multiple interventions. *BMC Bioinformatics*, 10:322, 2009.

[FXM+98] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature*, 391:806–811, Feb 1998.

[GCB+04] Robert C Gentleman, Vincent J. Carey, Douglas M. Bates, et al. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[GCD+05] R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber. *Bioinformatics and Computational Biology Solutions using R and Bioconductor.* Springer, New York, 2005.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.

[Gof08] S. P. Goff. Knockdown screens to knockout HIV-1. *Cell*, 135:417–420, Oct 2008.

[Grö04] M Grötschel. Vorlesungsskript algorithmische diskrete mathematik ii: Lineare optimierung. `http://www.zib.de/groetschel/teaching/skriptADMII.pdf`, WS 2003/2004.

[Hau03] P. Hauck. Skript zur vorlesung: Diskrete optimierung. `http://www-dm.informatik.uni-tuebingen.de/skripte/DiskreteOptimierung/Diskrete_Optimierung_Skript.pdf`, SS 2003.

[HC77] Andrew C. Harvey and Patrick Collier. Testing for functional misspecification in regression analysis. *Journal of Econometrics*, 6(1):103 – 119, 1977.

[Hec96] David Heckerman. A tutorial on learning with bayesian networks. Technical report, Learning in Graphical Models, 1996.

[HEK$^+$95] J. W. Harper, S. J. Elledge, K. Keyomarsi, B. Dynlacht, L. H. Tsai, P. Zhang, S. Dobrowolski, C. Bai, L. Connell-Crowley, and E. Swindell. Inhibition of cyclin-dependent kinases by p21. *Mol. Biol. Cell*, 6:387–400, Apr 1995.

[HP73] F. Harary and E. M. Palmer. *Graphical Enumeration.* Academic Press, 1973.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, New York, 2001.

[JBS$^+$03] A. L. Jackson, S. R. Bartz, J. Schelter, S. V. Kobayashi, J. Burchard, M. Mao, B. Li, G. Cavet, and P. S. Linsley. Expression profiling reveals off-target gene regulation by RNAi. *Nat. Biotechnol.*, 21:635–637, Jun 2003.

[JCL$^+$09] T. R. Jones, A. E. Carpenter, M. R. Lamprecht, J. Moffat, S. J. Silver, J. K. Grenier, A. B. Castoreno, U. S. Eggert, D. E. Root, P. Golland, and D. M. Sabatini. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proc. Natl. Acad. Sci. U.S.A.*, 106:1826–1831, Feb 2009.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC '84, pages 302–311, New York, NY, USA, 1984. ACM.

[KCT$^+$07] R. Konig, C. Y. Chiang, B. P. Tu, S. F. Yan, P. D. DeJesus, A. Romero, T. Bergauer, A. Orth, U. Krueger, Y. Zhou, and S. K. Chanda. A probability-based approach for the analysis of large-scale RNAi screens. *Nat. Methods*, 4:847–849, Oct 2007.

[KDZ$^+$09] L. Kaderali, E. Dazert, U. Zeuge, M. Frese, and R. Bartenschlager. Reconstructing signaling pathways from RNAi data using probabilistic Boolean threshold networks. *Bioinformatics*, 25:2229–2235, Sep 2009.

[KG00] M. Kanehisa and S. Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28:27–30, Jan 2000.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[KLC99] J. Kim, D. Lee, and J. Choe. Hepatitis C virus NS5A protein is phosphorylated by casein kinase II. *Biochem. Biophys. Res. Commun.*, 257:777–781, Apr 1999.

[KM72] Victor Klee and George J. Minty. How good is the simplex algorithm? In *Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin)*, pages 159–175. Academic Press, New York, 1972.

[KM06] J. T. Kittler and S. J. Moss. *The Dynamic Synapse: Molecular Methods in Ionotropic Receptor Biology.* CRC Press, Boca Raton (FL), 2006.

[KPH$^+$07] R. Kittler, L. Pelletier, A. K. Heninger, M. Slabicki, M. Theis, L. Miroslaw, I. Poser, S. Lawo, H. Grabner, K. Kozak, J. Wagner, V. Surendranath, C. Richter, W. Bowen, A. L. Jackson, B. Habermann, A. A. Hyman, and F. Buchholz. Genome-scale RNAi profiling of cell division in human tissue culture cells. *Nat. Cell Biol.*, 9:1401–1412, Dec 2007.

[KRK$^+$11] B. Knapp, I. Rebhan, A. Kumar, P. Matula, N. A. Kiani, M. Binder, H. Erfle, K. Rohr, R. Eils, R. Bartenschlager, and L. Kaderali. Normalization for Individual Cell Population Context in the Analysis of High-Content Cellular Screens. *BMC Bioinformatics*, Submitted in 2011.

[Kum10] A. Kumar. *Molecular Studies on Dengue Virus-Host Interaction.* PhD thesis, University of Heidelberg, June 2010. `http://archiv.ub.uni-heidelberg.de/volltextserver/frontdoor.php?source_opus=10750`.

[LBM$^+$00] H. A. Lane, I. Beuvink, A. B. Motoyama, J. M. Daly, R. M. Neve, and N. E. Hynes. ErbB2 potentiates breast tumor proliferation through modulation of p27(Kip1)-Cdk2 complex formation: receptor overexpression does not determine growth dependency. *Mol. Cell. Biol.*, 20:3210–3223, May 2000.

[LBN$^+$09] Q. Li, A. L. Brass, A. Ng, Z. Hu, R. J. Xavier, T. J. Liang, and S. J. Elledge. A genome-wide genetic screen for host factors required for hepatitis C virus propagation. *Proc. Natl. Acad. Sci. U.S.A.*, 106:16410–16415, Sep 2009.

[LSR$^+$11] A. Lan, I. Y. Smoly, G. Rapaport, S. Lindquist, E. Fraenkel, and E. Yeger-Lotem. ResponseNet: revealing signaling and regulatory networks linking genetic and transcriptomic screening data. *Nucleic Acids Res*, May 2011.

[LVHWN10] L. S. Lambeth, N. J. Van Hateren, S. A. Wilson, and V. Nair. A direct comparison of strategies for combinatorial RNA interference. *BMC Mol. Biol.*, 11:77, 2010.

[Mar06] F. Markowetz. Probabilistic Models for Gene Silencing Data. *PhD thesis, Free University Berlin*, 2006.

[MBS05] F. Markowetz, J. Bloch, and R. Spang. Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics*, 21:4026–4032, Nov 2005.

[MHC$^+$06] N. Malo, J. A. Hanley, S. Cerquozzi, J. Pelletier, and R. Nadon. Statistical practice in high-throughput screening data analysis. *Nat. Biotechnol.*, 24:167–175, Feb 2006.

[MKG$^+$05] P. Muller, D. Kuttenkeuler, V. Gesellchen, M. P. Zeidler, and M. Boutros. Identification of JAK/STAT signalling components by genome-wide RNA interference. *Nature*, 436:871–875, Aug 2005.

[MKR05] S. Mukhopadhyay, R. J. Kuhn, and M. G. Rossmann. A structural perspective of the flavivirus life cycle. *Nat. Rev. Microbiol.*, 3:13–22, Jan 2005.

[MKTS07] F. Markowetz, D. Kostka, O. G. Troyanskaya, and R. Spang. Nested effects models for high-dimensional phenotyping screens. *Bioinformatics*, 23:i305–312, Jul 2007.

[MKW$^+$09] P. Matula, A. Kumar, I. Worz, H. Erfle, R. Bartenschlager, R. Eils, and K. Rohr. Single-cell-based image analysis of high-throughput cell array screens for quantification of viral infection. *Cytometry A*, 75:309–318, Apr 2009.

[MLE$^+$03] V. K. Mootha, C. M. Lindgren, K. F. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. Ridderstrale, E. Laurila, N. Houstis, M. J. Daly, N. Patterson, J. P. Mesirov, T. R. Golub, P. Tamayo, B. Spiegelman, E. S. Lander, J. N. Hirschhorn, D. Altshuler, and L. C. Groop. PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, 34:267–273, Jul 2003.

[MRRK09] J. Mazur, D. Ritter, G. Reinelt, and L. Kaderali. Reconstructing nonlinear dynamic models of gene regulation using stochastic sampling. *BMC Bioinformatics*, 10:448, 2009.

[MS02] M. T. McManus and P. A. Sharp. Gene silencing in mammals by small interfering RNAs. *Nat. Rev. Genet.*, 3:737–747, Oct 2002.

[MS06] J. Moffat and D. M. Sabatini. Building mammalian signalling pathways with RNAi screens. *Nat. Rev. Mol. Cell Biol.*, 7:177–187, Mar 2006.

[MT77] F. Mosteller and J. W. Tukey. *Data Analysis and Regression: A Second Course in Statistics.* Addison-Wesley, 1977.

[MT04] G. Meister and T. Tuschl. Mechanisms of gene silencing by double-stranded RNA. *Nature*, 431:343–349, Sep 2004.

[MW47] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Statist.*, 18:50–60, 1947.

[NBBW07] C. J. Needham, J. R. Bradford, A. J. Bulpitt, and D. R. Westhead. A primer on learning in Bayesian networks for computational biology. *PLoS Comput. Biol.*, 3:e129, Aug 2007.

[NWH$^+$10] B. Neumann, T. Walter, J. K. Heriche, J. Bulkescher, H. Erfle, C. Conrad, P. Rogers, I. Poser, M. Held, U. Liebel, C. Cetin, F. Sieckmann, G. Pau, R. Kabbe, A. Wunsche, V. Satagopam,

M. H. Schmitz, C. Chapuis, D. W. Gerlich, R. Schneider, R. Eils, W. Huber, J. M. Peters, A. A. Hyman, R. Durbin, R. Pepperkok, and J. Ellenberg. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, 464:721–727, Apr 2010.

[OSI$^+$07] O. Ourfali, T. Shlomi, T. Ideker, E. Ruppin, and R. Sharan. SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, 23:i359–366, Jul 2007.

[PCL$^+$09] X. Peng, E. Y. Chan, Y. Li, D. L. Diamond, M. J. Korth, and M. G. Katze. Virus-host interactions: from systems biology to translational research. *Curr. Opin. Microbiol.*, 12:432–438, Aug 2009.

[PGB10] O. Pelz, M. Gilsdorf, and M. Boutros. web cellHTS2: a web-application for the analysis of high-throughput screening data. *BMC Bioinformatics*, 11:185, 2010.

[Pla05] L. C. Platanias. Mechanisms of type-I- and type-II-interferon-mediated signalling. *Nat. Rev. Immunol.*, 5:375–386, May 2005.

[PUC$^+$98] P. Pileri, Y. Uematsu, S. Campagnoli, G. Galli, F. Falugi, R. Petracca, A. J. Weiner, M. Houghton, D. Rosa, G. Grandi, and S. Abrignani. Binding of hepatitis C virus to CD81. *Science*, 282:938–941, Oct 1998.

[R D09] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2009. v2.12.1.

[RBB$^+$09] J. Reymann, N. Beil, J. Beneke, P. P. Kaletta, K. Burkert, and H. Erfle. Next-generation 9216-microwell cell arrays for high-content screening microscopy. *BioTechniques*, 47:877–878, Oct 2009.

[Rip81] Brian D. Ripley. *Spatial statistics / Brian D. Ripley.* Wiley, New York :, 1981.

[RKEK09a] N. Rieber, B. Knapp, R. Eils, and L. Kaderali. RNAither, an automated pipeline for the statistical analysis of high-throughput RNAi screens. *Bioinformatics*, 25:678–679, Mar 2009.

[RKEK09b] Nora Rieber, Bettina Knapp, Roland Eils, and Lars Kaderali. Rnaither, an automated pipeline for the statistical analysis of high-throughput rnai screens. *Bioinformatics*, 25:678–679, March 2009.

[RRB+11] S. Reiss, I. Rebhan, P. Backes, I. Romero-Brey, H. Erfle, P. Matula, L. Kaderali, M. Poenisch, H. Blankenburg, M. S. Hiet, T. Longerich, S. Diehl, F. Ramirez, T. Balla, K. Rohr, A. Kaul, S. Buhler, R. Pepperkok, T. Lengauer, M. Albrecht, R. Eils, P. Schirmacher, V. Lohmann, and R. Bartenschlager. Recruitment and activation of a lipid kinase by hepatitis C virus NS5A is essential for integrity of the membranous replication compartment. *Cell Host Microbe*, 9:32–45, Jan 2011.

[Sch99] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1999.

[SCMS+05] A. Sweet-Cordero, S. Mukherjee, A. Subramanian, H. You, J. J. Roix, C. Ladd-Acosta, J. Mesirov, T. R. Golub, and T. Jacks. An oncogenic KRAS2 expression signature identified by cross-species gene-expression analysis. *Nat. Genet.*, 37:48–55, Jan 2005.

[SFL+09] O. Sahin, H. Frohlich, C. Lobke, U. Korf, S. Burmester, M. Majety, J. Mattern, I. Schupp, C. Chaouiya, D. Thieffry, A. Poustka, S. Wiemann, T. Beissbarth, and D. Arlt. Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC Syst Biol*, 3:1, 2009.

[SH06] L. Sachs and J. Hedderich. *Angewandte Statistik: Methodensammlung mir R*. Springer, 2006.

[SPWM98] R. L. Sutherland, O. W. Prall, C. K. Watts, and E. A. Musgrove. Estrogen and progestin regulation of cell cycle progression. *J Mammary Gland Biol Neoplasia*, 3:63–72, Jan 1998.

[SRM+10] A. Suratanee, I. Rebhan, P. Matula, A. Kumar, L. Kaderali, K. Rohr, R. Bartenschlager, R. Eils, and R. Konig. Detecting host factors involved in virus infection by observing the clustering of infected cells in siRNA screening images. *Bioinformatics*, 26:i653–658, Sep 2010.

[SSR+09] B. Snijder, R. Sacher, P. Ramo, E. M. Damm, P. Liberali, and L. Pelkmans. Population context determines cell-to-cell variability in endocytosis and virus infection. *Nature*, 461:520–523, Sep 2009.

[Str11] String. String (search tool for the retrieval of interacting genes/proteins); version 9.0. `http://string-db.org/`, October 2011.

[SWL$^+$05] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F. H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, J. Timm, S. Mintzlaff, C. Abraham, N. Bock, S. Kietzmann, A. Goedde, E. Toksoz, A. Droege, S. Krobitsch, B. Korn, W. Birchmeier, H. Lehrach, and E. E. Wanker. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122:957–968, Sep 2005.

[Tan06] S. L. Tan. *Hepatitis C Viruses: Genomes and Molecular Biology.* Horizon Bioscience, Norfolk (UK), 2006.

[TB10] M. Theis and F. Buchholz. High-throughput RNAi screening in mammalian cells with esiRNAs. *Methods*, Dec 2010.

[TBP$^+$09] A. W. Tai, Y. Benita, L. F. Peng, S. S. Kim, N. Sakamoto, R. J. Xavier, and R. T. Chung. A functional genomic screen identifies cellular cofactors of hepatitis C virus replication. *Cell Host Microbe*, 5:298–307, Mar 2009.

[TLDR$^+$09] M. Trotard, C. Lepere-Douard, M. Regeard, C. Piquet-Pellorce, D. Lavillette, F. L. Cosset, P. Gripon, and J. Le Seyec. Kinases required in hepatitis C virus entry and replication highlighted by small interference RNA screening. *FASEB J.*, 23:3780–3789, Nov 2009.

[TM08] A. Tresch and F. Markowetz. Structure learning in Nested Effects Models. *Stat Appl Genet Mol Biol*, 7:Article9, 2008.

[TOR$^+$01] G. C. Tseng, M. K. Oh, L. Rohlin, J. C. Liao, and W. H. Wong. Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects. *Nucleic Acids Res.*, 29:2549–2557, Jun 2001.

[TQL$^+$06] R. Tibes, Y. Qiu, Y. Lu, B. Hennessy, M. Andreeff, G. B. Mills, and S. M. Kornblau. Reverse phase protein array: validation of a novel proteomic technology and utility for analysis of primary leukemia specimens and hematopoietic stem cells. *Mol. Cancer Ther.*, 5:2512–2521, Oct 2006.

[UIPT$^+$10] S. Urcuqui-Inchima, C. Patino, S. Torres, A. L. Haenni, and F. J. Diaz. Recent developments in understanding dengue virus replication. *Adv. Virus Res.*, 77:1–39, 2010.

[vGvK11] Michael van Ginkel and Geert van Kempen. Dipimage toolbox. `http://www.diplib.org/`, September 2011.

[VPC+09] F. H. Vaillancourt, L. Pilote, M. Cartier, J. Lippens, M. Liuzzi, R. C. Bethell, M. G. Cordingley, and G. Kukolj. Identification of a lipid kinase as a host factor involved in hepatitis C virus RNA replication. *Virology*, 387:5–10, Apr 2009.

[Wil46] F. Wilcoxon. Individual comparisons of grouped data by ranking methods. *J. Econ. Entomol.*, 39:269, Apr 1946.

[WKIKG09] S. E. Winograd-Katz, S. Itzkovitz, Z. Kam, and B. Geiger. Multiparametric analysis of focal adhesion formation by RNAi-mediated gene knockdown. *J. Cell Biol.*, 186:423–436, Aug 2009.

[WM09] M. D. White and G. R. Mallucci. Therapy for prion diseases: Insights from the use of RNA interference. *Prion*, 3:121–128, Jul 2009.

[Wor09] World Health Organization. Dengue and dengue haemorrhagic fever. `http://www.who.int/mediacentre/factsheets/fs117/en/`, March 2009.

[Wor11] World Health Organization. Hepatitis C. `http://www.who.int/csr/disease/hepatitis/whocdscsrlyo2003/en/index4.html`, Jan 2011.

[WRBB08] A. M. Wiles, D. Ravi, S. Bhavani, and A. J. Bishop. An analysis of normalization methods for Drosophila RNAi genomic screens and development of a robust validation scheme. *J Biomol Screen*, 13:777–784, Sep 2008.

[WWW+07] R. S. Wang, Y. Wang, L. Y. Wu, X. S. Zhang, and L. Chen. Analysis on multi-domain cooperation for predicting protein-protein interactions. *BMC Bioinformatics*, 8:391, 2007.

[YGvV+95] Ian T. Young, Jan J. Gerbrands, Lucas J. van Vliet, Cip data Koninklijke Bibliotheek, Den Haag, Young Ian Theodore, Gerbrands Jan Jacob, Van Vliet, and Lucas Jozef. Fundamentals of image processing, 1995.

[ZBB05] A. Zaczek, B. Brandt, and K. P. Bielawski. The diverse signaling network of EGFR, HER2, HER3 and HER4 tyrosine kinase receptors and the consequences for therapeutic approaches. *Histol. Histopathol.*, 20:1005–1015, Jul 2005.

[ZCO99] J. H. Zhang, T. D. Chung, and K. R. Oldenburg. A Simple Statistical Parameter for Use in Evaluation and Validation of High Throughput Screening Assays. *J Biomol Screen*, 4:67–73, 1999.

[ZH05] B. Zhang and S. Horvath. A general framework for weighted gene co-expression network analysis. *Stat Appl Genet Mol Biol*, 4:Article17, 2005.

[ZZC08] S. Zhang, X. S. Zhang, and L. Chen. Biomolecular network querying: a promising approach in systems biology. *BMC Syst Biol*, 2:5, 2008.