Network Propagation with Node Core for Genotype-Phenotype Associations and Module Identification

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Gal Barel

Berlin , 2020

Erstgutachter: Prof. Dr. Martin Vingron
Zweitgutachter: Prof. Dr. Ulrich Stelzl
Tag der Disputation: 21 Januar 2021

Preface

Publications and contributions

This thesis is built on a project that was developed upon extensive experimentation and explorations of network propagation applications for studying genotype-phenotype associations, and in particular disease genes and disease modules. The idea of modifying the commonly used network propagation formulation grew out of fruitful discussions with my adviser Ralf Herwig. This approach was recently published in *Nucleic Acids Research (NAR)* [30] and is available at https://doi.org/10.1093/nar/gkaa639. In addition to the published version, this thesis includes further implementations, evaluations and applications, in order to further expand upon aspects which were not previously demonstrated and discussed. One of which is the application of the approach to study drug-toxicity. Prior to the development of the approach I developed a workflow for the analysis of toxicogenomics data using pathway and network information, which was previously published in *Frontiers in genetics* [29] and is only briefly mentioned in this thesis. Furthermore, I participated in the analysis of drug-toxicity data which was collected, in collaboration with others, under the 7th framework project "HeCaToS". Parts of the results have recently been accepted for publication in *Communications biology* [263], and some of the data was re-analyzed by me for the purpose of this thesis.

Throughout the chapters of this thesis I will use the personal pronoun *we* to indicate work done by myself, with the supervision and support from my adviser Ralf Herwig. Any results which are based on previous works by others or which were done in collaboration will be mentioned accordingly in the text.

Acknowledgments

First and foremost, I would like to profoundly thank my adviser Ralf Herwig, for providing me with the opportunity to work as a PhD candidate in his group at the Max Planck Institute (MPI) for Molecular Genetics. I am very grateful for his supervision and scientific support, as well as for the chance to collaborate on multiple projects. I am also thankful to my supervisor Martin Vingron, for accepting me into the Computational Biology department, advising my work and reviewing this thesis. I would also like to thank Annalisa Marsico for participating during my thesis advisory committee in fruitful discussions. Finally, I am grateful to Ulrich Stelzl who agreed to serve as my external supervisor and for reviewing the work in this thesis.

I would further like to thank my group members (and office mates) Matthias Lienhard and Kristina Thedinga, who also supported me by proof-reading this work and

provided useful comments. I am very grateful to be part of the Computational Biology department, and would like to thank all of its members, for interesting scientific discussions during seminars and retreats, as well as for great times spent together. I want to especially thank Philip Kleinert and Edgar Steiger, who became dear friends, and also supported me by commenting on this work. I am also fortunate enough to be part of The International Max Planck Research School for Biology AND Computation (IMPRS-BAC), which provided me with the opportunity to get to know fellow students and participate in scientific and soft skill seminars. I am forever grateful to Kirsten Kelleher for her tremendous help on all PhD related issues, for organizing so many wonderful events and workshops, and for her never-ending will to help, always with a smile on her face. I also want to thank her for proof-reading this thesis and for her assistance with translating to German.

I would like to especially thank my friends Tobias Zehnder and Roman Schulte-Sasse, for our great times together during the course of our PhD work at the MPI for Molecular Genetics. I am thankful for their amazing support all along, and especially in the months of writing this work. I also want to thank them for proof-reading of and commenting on this thesis. I consider myself very lucky to have met them, and I am sure my work benefited greatly from their opinions.

Finally, my deepest gratitude goes to my family and friends, who supported me throughout the years of working on my PhD. I especially thank my parents, my brother and my sister for their unconditional trust in me, while working on my PhD away from them, yet with so much encouragement. Last but not least, I deeply thank Felix Sorantin, my beloved partner, who is always by my side, even through challenging times, with his never-ending patience, optimism and inspiration.

Contents

List of Figures

List of Tables

List of Abbreviations

AP-MS	affinity purification coupled to mass-spectrometry
AUROC	area under the ROC
BC	betweenness centrality
CC	clustering coefficient
cDNA	complementary DNA
CNV	copy number variation
CPDB	ConsensusPathDB
DAU	daunorubicin
DEG	differentially expressed gene
DMSO	dimethyl sulfoxide
DNA	deoxyribonucleic acid
DOX	doxorubicin
EPI	epirubicin
FC	fold change
FDR	false discovery rate
GO	gene ontology
GRN	gene regulatory network
GWAS	genome wide association studies
IDA	idarubicin
KEGG	Kyoto Encyclopedia of Genes and Genomes
mRNA	messenger RNA
NAGA	Network Assisted Genomic Association
NCG	Network of Cancer Genes
NGS	next generation sequencing
OMIM	Online Mendelian Inheritance in Man
PPI	protein-protein interaction
RDPN	random degree-preserving networks

RNA	ribonucleic acid
RNA-seq	RNA sequencing
ROC	receiver operating characteristic
RWR	random walk with restart
SCC	strongly connected component
SNP	single nucleotide polymorphism
SNV	single nucleotide variant
TCGA	The Cancer Genome Atlas
TF	transcription factor
Y2H	yeast two hybrid

Part I

INTRODUCTION AND PRELIMINARIES

1 Introduction

One of the main goals in molecular biology, and even more so in molecular genetics, is to be able to associate genomic elements with some (disease) phenotype. To that end, researchers attempt to decipher the role of each molecule and its contribution to the concert of events at the cell which lead to the emerging phenotype. The key molecules, namely DNA, RNA and proteins, drive most of the cellular functions by constantly interacting with each other. Among these interactions, the ones between two and more proteins play an important role by regulating and executing many processes in the cell. These interactions are often depicted as network, which serves as a powerful scaffold for analyzing and investigating molecular data. One such possible analysis involves the diffusion of molecular evidence throughout the entire network in a process termed network propagation. Since genes that carry similar functions are believed to be connected in the network, the signal is thus amplified, enabling researches to draw informed conclusions about the mechanisms that lead to the phenotype under investigation. Several approached based on this concept have already been successfully applied to genomic measurements, mostly for the detection of disease genes and modules.

1.1 Research objective

This thesis presents a network propagation framework with the aim of identifying novel genotype-phenotype associations as well as relevant network modules. The developed method provides two main novel contributions that address drawbacks which arose from inspecting previous network propagation approaches in molecular biology. The first is a modification to the mathematical formulation of network propagation which provides an improved ranking of the genes at the end of the propagation. The second is a semi-supervised module identification approach that incorporates prior knowledge and allows for the detection of network modules which connect well-known genes with novel ones. The method was initially developed with the goal of improving disease-gene predictions and disease-module detection, and as such was applied to two complex diseases. Additionally, the method is also suitable for analyzing other types of genotype-phenotype associations, and was therefore also demonstrated on drug-toxicity data.

1.2 Thesis outline

This thesis is divided into two parts. The first is aimed at providing relevant background for the work which is presented in the second part. This Chapter serves as a brief introduction for the scope of the thesis. Chapter 2 will provide an introduction to

molecular biology, focusing mainly on molecular interactions, pathways and networks. Chapter 3 will include an introduction to experimental and computational techniques for the purpose of associating phenotypes, mainly disease ones, with genomic elements. The experimental background will be centered around genomic measurements that are produced using sequencing technologies, and how those can be used to identify disease-genes. The computational background will focus on computational methods for identifying disease genes, and mainly on network-based methods, with a particular emphasis on network propagation methods. The last Chapter in this section, Chapter 4, will contain fundamental concepts that are relevant to the thesis, namely definitions from graph theory and the mathematics of network propagation.

The second part of this thesis is comprised of four chapters and contains a comprehensive report of the work that was done. Chapter 5 will describe the developed method, which consists of a modified version of network propagation for genotype-phenotype associations. The first part is concerned with the adjustment of the mathematical formulation, and the second one with a semi-supervised module identification procedure. In Chapter 6 the performance of the method will be evaluated using a set of well-known disease-associated genes. In addition, the influence of the various parameters of the method on its performance will be presented. In Chapter 7 the method will be applied to three different data sets in order to demonstrate its benefits in identifying genes and modules that are relevant to the experimental evidence. The results will be compared with other state-of-the-art methods which were previously developed for similar purposes. Finally, the method, its performance, application and results will be discussed in detail in Chapter 8. The work will be concluded, and future applications and improvements will be devised.

2 Molecular Biology Background

This chapter is aimed at introducing the field of molecular biology in general, and in particular of molecular genetics. The key players in molecular biology are presented, in addition to models for representing their interactions, such as pathways and networks. Finally, we review briefly how disruptions of the genome can lead to changes in molecular interactions and therefore cause disease phenotypes.

2.1 Key players in molecular biology

Living organisms are composed of one or more cells. The cell, which consists of a cytoplasm surrounded by a membrane, contains different kinds of biomolecules in various cell compartments (Figure 2.1). The nucleus contains the deoxyribonucleic acid (DNA), a long double stranded molecule, comprised of four nucleotides (also called bases), that are connected to one another in a helical shape. The DNA contains the genetic information of the cell, that codes for genes and other genetic elements. Sections of the DNA can be transcribed to ribonucleic acid (RNA), a single stranded molecule which is also comprised of four nucleotides. There are several types of RNA molecules in the cell, which are involved in different molecular processes. A messenger RNA (mRNA) molecule holds the coding information of a gene, i.e. the exons, which can be translated into a protein by the Ribosome machinery. The introns are the non-coding sequences of a gene which are removed prior to translation. Proteins are long chains of connected amino acids, that consist of one or more polypeptides, which perform many of the cell's functions. They help carry out chemical reactions on other molecules which are present in the cell, such as lipids and carbohydrates, as well as other types of small metabolites. Protein can be classified into families, which usually reflect their similarity in sequence and function. For example, enzymes are a family of proteins which act as biocatalysts, and are able to accelerate biochemical reactions. Another example are signaling proteins, which bind to a ligand, and activate a chain of transmission that results in a cellular response.

2.2 Molecular interactions

The many components of a cell interact with each other in order to execute all the necessary processes for the cell to function. These processes vary from the most vital ones that are present in every cell, such as metabolism and energy conversion, to specific ones that can determine the cell's identity and role. It is crucial for the interactions to successfully occur in order to carry out all processes, otherwise numerous damages could arise, that might eventually lead to disease. The types of interactions

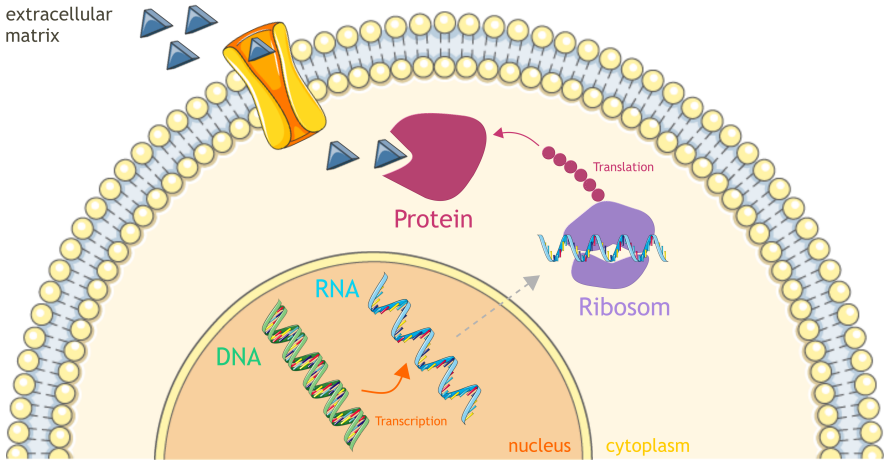


Figure 2.1: Key players in molecular biology: The cell is separated from the extracellular matrix by a membrane. The nucleus, one of the cell organelles, contains the DNA, which is transcribed into RNA molecules. The RNA is transported outside of the nucleus and into the cytoplasm, where it is translated by the Ribosome into a protein. The protein binds to a signaling ligand, which is imported into the cell via a receptor protein which is bound to the membrane. (Original illustrations taken from SMART Servier Medical Art [https://smart.servier.com/])

depend on the molecules involved. There are metabolic interactions, which are physical-chemical reactions within the cell, for example an enzyme catalyzing the conversion of one molecule to another. A signaling interaction usually involves two proteins, where one chemically modifies some property of the amino acids of the other, for example in a phosphorylation process. Proteins can also chemically interact via the side chains of their amino-acids and form protein complexes, which serve as a functional machinery to perform a specific process. Transcription factors (TFs) are a special kind of proteins that can bind to regions in the DNA via genetic interactions and by that regulate the expression levels of a gene. Naturally, many biological processes are highly complex and include multiple types of molecules and interactions. For example insulin is secreted in response to glucose, however the process is tightly regulated and involves the integration of many signals from both metabolites and hormones. Therefore, it is usually very useful to summarize and overview biological processes in pathways and networks, which will be elaborated in the next Sections.

2.3 Molecular pathways

Many biological processes can be represented in pathways, which describe a series of reactions between genes, proteins and other metabolites. The main idea behind a pathway is to compile the set of participating players and their interactions which occur together in order to execute some process. The most common pathways are associated with metabolic reactions, gene regulatory mechanisms or signal transduction processes. In addition, pathways can also represent a group of reactions which are associated with the same outcome, for instance developmental pathways or complex disease progression, such as cancer. Pathways can be visually represented as graphs, where nodes correspond to biomolecules and reactions are represented by arrows or

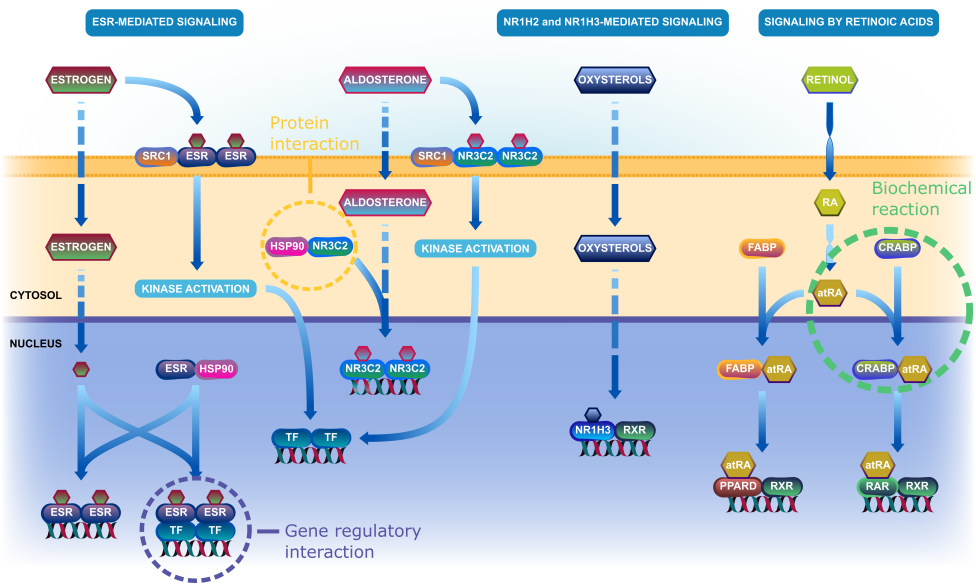


Figure 2.2: Signaling by Nuclear Receptors: An example of different signaling pathways, invoked by the signaling molecules estrogen and aldosterone or by oxysterol and retinol (also known as vitamin A1). These molecules trigger a chain of signaling reactions that eventually effect gene expression in the nucleus and by that some cellular function. These pathways include different molecular interactions such as biochemical, protein-protein and gene regulatory ones. For example, the estrogen hormone binds to estrogen receptors, which can then act as TFs and regulate the expression of genes which control cellular processes like proliferation and differentiation. Adapted from Reactome [139].

lines connecting them. Figure 2.2 illustrates three different signaling pathways which involve metabolic, protein-protein and gene regulatory interactions. Pathway concepts are very useful when analyzing molecular data. They help to aggregate knowledge regarding complex mechanisms and represent the interdependencies between many cellular processes.

Molecular pathways can be defined in different ways, and are usually collected in databases that are publicly available. However, each database is curated differently, with interactions being collected from the literature and manual data submissions, and are therefore not always comprehensive or consistent with others [33, 272]. Pathguide [21], an online pathway resource list, currently provides information about more than 700 such pathway resources. Many of the resources span over multiple types of interactions, for example the Kyoto Encyclopedia of Genes and Genomes (KEGG) [146] database, which collects knowledge of metabolic reactions, signal transduction, genetic regulation and more. Some resources are more specific and include only one type of interactions, for example DrugBank [319] which collects information about drug-target interactions and drug metabolism pathways. Due to the large number of resources, and in order to concentrate pathway information for all kinds of interactions, there have been efforts to integrate several resources into one. Pathway Commons [247] currently integrates pathway information from 22 databases, while ConsensusPathDB (CPDB)

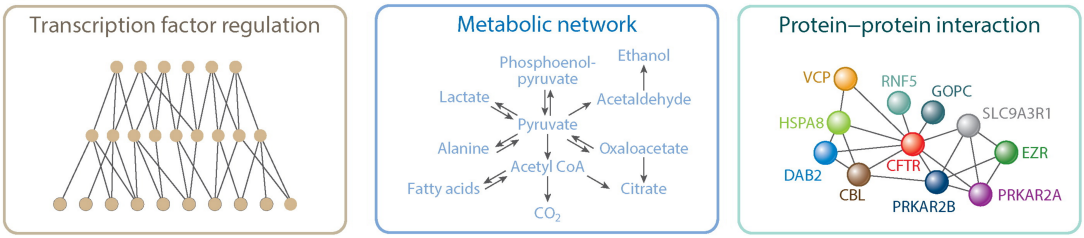


Figure 2.3: Molecular interaction networks: Visualization of three example interaction networks in a cell. On the left is a TF interaction network, which represents a gene regulatory network (GRN). The nodes are TFs and the edges suggest to a regulative interaction. In the middle is a metabolic network, describing biochemical reactions in the cell. The nodes are metabolites and the edges represent the biochemical reaction which converts one to another. On the right is a PPI network, where nodes are proteins and edges represent a direct or complex interaction between them. Adapted from [188].

[118] provides information for over 5,000 pathway concepts collected from 32 different resources. These types of meta-databases agglomerate most of the information that is relevant for each pathway, and therefore enable a comprehensive overview of the players and their interactions.

2.4 Protein-protein interaction networks

Many living systems can be described using networks. In social networks, connections describe interactions between individuals or organizations [311]. The world wide web can be represented as a vast network, connecting millions of HTML pages [8]. A cell can also be described as a network, whose players are molecules such as genes and proteins, and the connections describe biochemical reactions [315]. This network is sometimes also referred to as the interactome, the set of all molecular interactions in a cell. Most of these huge complex networks are highly structured and follow a common topology [313]. The aggregation of information into networks enables their analysis and can generate useful observations that have applications in many disciplines. Network theory is largely based on graph theory, where networks are mathematically defined using graph models. The relevant definitions are given in Section 4.1.

In molecular biology, different systems can be described using networks (For examples see Figure 2.3). The chemical interactions between proteins are usually described by protein-protein interaction (PPI) networks. Genetic interactions, which represent gene-gene interactions, are portrayed as gene regulatory networks (GRNs). Metabolic interactions, which usually represent enzymatic reactions, can also be described as a network, connecting enzymes with metabolites. Each one of these networks not only serves as a fundamental source of information, but can also serve as a tool for analyzing and understanding all sorts of biological questions [188]. Many metrics and concepts that have been developed in graph theory are applicable to molecular interaction networks, as well as a large variety of algorithms which can be applied to them. At the center of this work are PPI interactions and networks. We will describe how interactions between proteins can be experimentally measured, focusing on one of the most com-

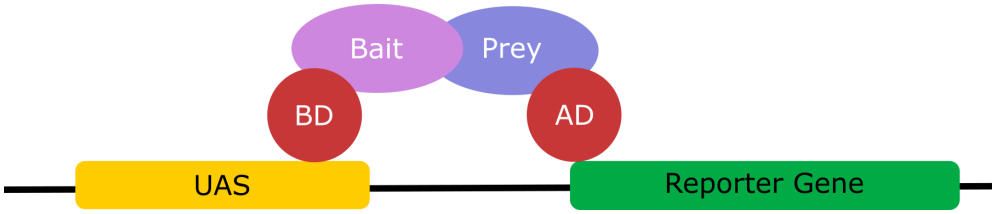


Figure 2.4: The yeast two hybrid system: The 'bait' protein is attached to a binding domain (BD) and the 'prey' protein is attached to the activating domain (AD). Once the 'bait' protein interacts with the 'prey' protein, the BD is attached to the upstream activating sequence (UAS) and the AD to the reporter gene, such that it can be transcribed. Thus, if the reporter gene is detected, then it serves as a measure for the interaction between the 'bait' and the 'prey' proteins.

mon techniques. Furthermore, we will review how PPI networks have been constructed over the years, and mention some of the recent efforts to expand the knowledge in the field.

Proteins are characterized by domains, a region in the protein's sequence that is associated with some function. Thus, two proteins can interact with each other via one of their domains, such that a chemical bond is created between the amino acids. Proteins can also organize themselves into a complex, where multiple bonds are formed between two or more proteins, resulting in an energetically favorable structure, where the proteins stay connected to one another [184]. These protein complexes are able to perform various molecular functions that are essential for the cell. In fact, many fundamental processes are carried out by complex machineries of multiple proteins, for example the Mediator complex, which helps activate transcription of genes [10].

There are two main experimental techniques to study protein interactions: the yeast two hybrid (Y2H) system [88] and affinity purification coupled to mass-spectrometry (AP-MS) methods [179]. Y2H is designed to capture binary interactions between two proteins, while the AP-MS methods are able to measure protein complex interactions. The latter can only detect which proteins come together within a complex, but does not necessarily determine the binary interactions within the complex [327]. PPI networks usually represent binary interactions, and therefore generally include interactions that were captured via Y2H experiments [221]. This work mainly focuses on the Y2H system, as even though it is widely used for measuring PPIs, it also suffers from some technical biases, which are described in Section 3.6.1.

Y2H is a genetic system, based on the *GAL4* protein of the yeast *Saccharomyces cerevisiae*, to study interactions between two proteins. Figure 2.4 visualizes the main components of the system. *GAL4* is a TF that activates expression that is necessary for the degradation of galactose. The protein has two main domains, the first binds to the DNA, and the second contains a region that is necessary for transcription activation. The Y2H system contains two fused proteins: a 'bait' protein, which is fused to the DNA binding domain of *GAL4*, and a 'prey' protein, which is fused to the activating domain. Only when the fused proteins interact via the 'bait' and 'prey' domains is the transcription activated and can be measured. The basic Y2H system was later expanded to a high-throughput screen, and thus almost all 6,000 of the yeast proteins were tested for

binary interactions [137, 138, 293, 305]. The same method could easily be utilized for studying protein interactions in other model organisms, for example *C. elegans* [303, 304], and was later also adapted to humans [250, 277].

Interactions between proteins are most commonly described using network models. Every protein is represented as a node in the network, and an interaction between two proteins is represented by an edge. The interactions can also be weighted, such that the weight describes the confidence level in the interaction. Typically such networks follow the same model of other complex systems, such as the internet, and can be represented by scale-free graphs [27, 140, 325]. Such graphs are characterized by a power law distribution of the degree, i.e. most of the nodes in the network have a low degree, and only a small sub-set has an extremely high degree (definitions and further details are given in Section 4.1.3). This creates a 'small-world' effect, where any two nodes are connected via a path of a few edges only [65, 313].

Over the years there have been tremendous efforts to collect protein interactions, summarize them in networks and make them available via online databases and resources. Starting from PPI networks of yeast proteins [260, 322] and quickly expanding to networks of human proteins [20, 218, 253, 330]. The first human PPI networks included a rather small number of interactions [74], which in turn grew larger and larger, as the high-throughput experimental technologies advanced [254]. Furthermore, interactions could be computationally predicted for more organisms based on sequence homology of the proteins and potential conservation of the interactions between them [165, 186]. In addition to experimental and computational techniques for identifying new interactions, those can also be extracted via automated text mining of abstracts and publications [80, 181].

As the number of PPIs databases grew, there was a need to collect all of the information and centralize it into one resource [54]. This way, all the experimental and curated evidence for an interaction would become available under one place. Some databases focus only on protein interactions, such as STRING [283], where others also provide additional interactions. CPDB [145] is an example for such meta-database, summarizing more than 600,000 unique interactions from 32 public resources, out of which more than 400,000 are interactions between proteins. Since these meta-databases combine information from multiple resources, they can also provide a confidence score for every interaction, to suggest how many times it was reported and on which type of evidence it is based. CPDB uses `IntScore` [143], an interaction confidence scoring tool, that combines both topology based and annotation based methods, and generates a final integrated score. STRING uses a different approach [301], where the scores are derived according to a trusted data set that is defined using KEGG pathways [146].

2.5 Molecular genetics - from genotype to phenotype

The field of genetics, and molecular genetics in particular, has made a long way since Gregor Mendel established the rules of inheritance, in what is nowadays termed

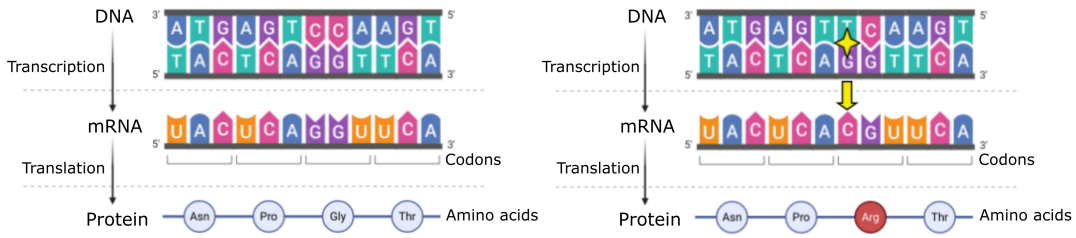


Figure 2.5: The central dogma of molecular biology: On the left, the double stranded DNA is transcribed into mRNA which is then translated into the amino acid sequence that makes up a protein. When the DNA is mutated, for example by a point mutation which changes one **C** base to **T**, as depicted on the left, then the mutation is also carried on to the mRNA. Due to the genetic code, the change in one base changes the matching amino acid from Glychin (Gly) to Argenin (Arg). This change in amino acid could affect for example the 3D structure of the protein and by that its usual function. (Figure created in bioRENDER [https://biorender.com/])

Mendelian inheritance. Already in the early 1900s it was clear that some genetic material controls certain traits, or phenotypes, and that it is inherited and not just acquired. Although it was already known then that genes existed on chromosomes, only in the 1940s was it made clear that the DNA is the molecule which carries the genetic information. In 1953 James Watson and Francis Crick famously determined the three-dimensional double-helix structure of the DNA, based on crystallographic work of Rosalind Franklin and Maurice Wilkins [312]. The double-helix model, containing two strands of DNA, which are complementary to each other, provided the concept of replication, where a new strand can be reconstructed based on the sequence of the old one. This also explains the mode of inheritance, where every part of the genome exists in two copies, one maternal and one paternal. Eventually it became clear that the four bases of the DNA contain the genetic code, and that the genetic information is converted into proteins via mRNA, in what is now known as the central dogma [73]. Since a protein is responsible for carrying out some cellular process, any change to its function could lead to a change in the cell's phenotype. And as the protein is determined by the genetic code, this means that even a single change to one base of the DNA could substantially affect the protein, and cause a genetic disorder or disease. Figure 2.5 illustrates how a single mutation in the DNA sequence can eventually alter the protein that it is translated to.

2.5.1 Genetic disorders

Broadly speaking, a genetic disorder or disease is caused by a variation or an alteration of the DNA. For the disease to be inherited, this alteration must be present in the germline cells, such that it is carried on to the next generation. The alteration, also called allele, can range from a change in a single nucleotide to larger changes such as insertion (additional new nucleotides), deletion (missing nucleotides), duplication (additional existing nucleotides) or even aberration to an entire chromosome. In the simple case, an alteration of one gene only can be directly associated with the disease phenotype, in what is termed Mendelian diseases. The mode of inheritance in such

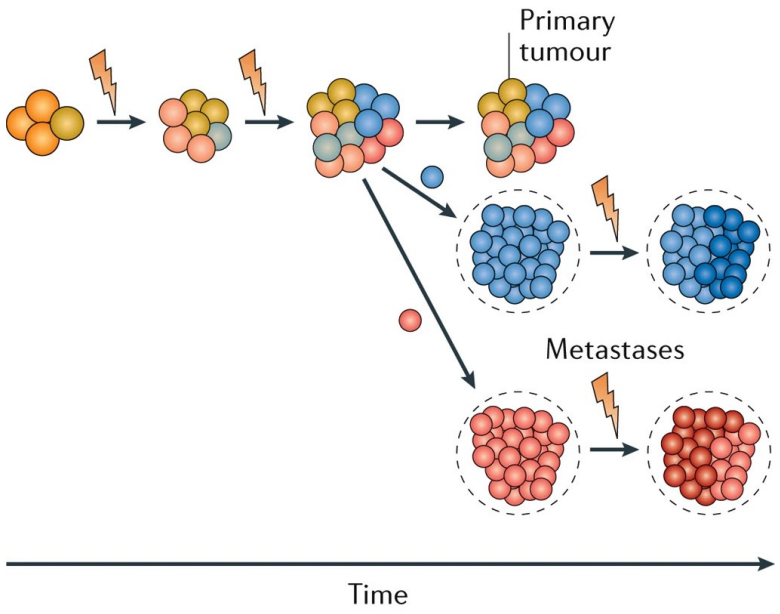


Figure 2.6: Mutations in cancer progression: Cancer cells accumulate mutations over time, such that the primary tumor is comprised of multiple cell populations. Some sub-clonal cells might also form metastases in other tissues, where they continue to accumulate more mutations. Adapted from [134].

cases is either dominant, when a mutation in one of the copies of the gene is enough to cause the disease phenotype, or recessive, where both alleles must be mutated for the disease to manifest. However, genetic disorders can also be more complex, such that the disease phenotype is affected by many genes, sometimes also in combination with environmental factors. Online Mendelian Inheritance in Man (OMIM) [12] currently lists more than 6,000 disease phenotypes for which the molecular basis is known[1], the majority of them are single gene disorders and traits. However, most common diseases are complex ones, and therefore it is more challenging to identify all the genetic factors that contribute to the disease phenotype [40]. For this purpose it can be very useful to analyze such complex diseases using pathways and networks, as those allow researchers to accumulate the experimental evidence and review the process as a whole.

2.5.2 Cancer

Contrary to genetic diseases, cancer is a complex disease that arises by changes in the DNA of somatic cells, i.e. any cell that is not of the germline. Early genetic changes, such as single nucleotide variants (SNVs), copy number variations (CNVs) or even whole chromosomal rearrangements, grant the cancerous cells a selective advantage, which promotes abnormal cell growth and forms a tumor [111]. A mutation in a tumor suppressor gene, which is usually involved in the regulation of cell division and replication, that also results in its loss of function could promote cancer formation. On the other hand, if a gene that is involved in cell growth and proliferation is mutated such that it is

1 `https://omim.org/statistics/geneMap`

expressed in higher levels then that gene can become an oncogene, and therefore also promote cancer formation. As the cells proliferate, they accumulate more and more genetic changes, and can eventually spread from their primary location to other tissues (Figure 2.6). More than 100 types of cancers, spanning over most tissues, are known to affect humans. However, the mutational landscape varies greatly between different types of cancers, as well as between patients within the same type. Some genes are known to be highly mutated, whereas many genes are only mutated in several patients [98]. Therefore it is particularly challenging to identify those genes that promote cancer development and progression. The "genomic era" has certainly made it possible to advance cancer research, as sequencing technologies allowed to study the cancer genome in high depth [317]. The first census of human cancer genes [95], which reported an initial list of 291 genes in 2004, has since expanded to include over 700 genes [273] and the efforts are still ongoing. On top of that, cancer progression could be a consequence of the disruption of some cellular process, and so the observed mutational landscape in patients is different [112]. Hence, it is essential to also examine the mutations in the context of pathways and networks, which give a more comprehensive overview of the entire process and the effects that one gene has on the others.

3 Experimental and Computational Techniques in Genotype-Phenotype Associations

In this chapter we guide the reader through the process of identifying genomic variations that cause diseases, in particular complex diseases and cancer. We review the experimental techniques for sequencing the genome, detecting mutations and measuring gene expression levels. We then describe the computational procedures for identifying disease genes, and focus on methods which utilize molecular interaction networks for this task. Finally, we introduce the concept of network propagation and its application for disease genes and disease modules identification. We give an overview of propagation-based methods, discuss existing challenges and motivate for a revised model.

3.1 High-throughput quantification of the genome via next generation sequencing

Studying the genome requires the detection of the nucleic acids, which the DNA is composed of, and the identification of their order. Major developments have taken place in the field of DNA-sequencing since the first generation was developed by Fred Sanger [255] and later termed Sanger sequencing. While the first technologies were used for many years, they were limited to short fragments of less than one kilobase (kb) [117]. This meant that in order to identify the entire human genome it first had to be fragmented into shorter sequences. After those were resolved, then the fragments needed to be assembled together to reconstruct the whole sequence. With time, further technological improvements finally allowed researchers to determine the first draft of the human genome in the early 2000s [160, 298]. The microarray platform [225, 266] first enabled to study the genome in a highly parallel assay for measuring both the DNA and RNA molecules [210]. The second generation of sequencing, nowadays usually referred to as NGS, allowed for mass parallelisation, substantially increasing the amount of DNA that could be sequenced at the same time [182], and allowing to complete the sequencing of the human genome in a much shorter time and at a considerably lower cost [316]. Upon further advancements came the third generation of sequencing which enabled the sequencing of a single molecule [256], and allowed to produce much longer reads, i.e. longer fragments of DNA [76]. By now there are dozens of different sequencing platforms, which differ mainly in the number of reads they produce and their lengths [105].

All of the above contributed to the genomics revolution and will no doubt continue

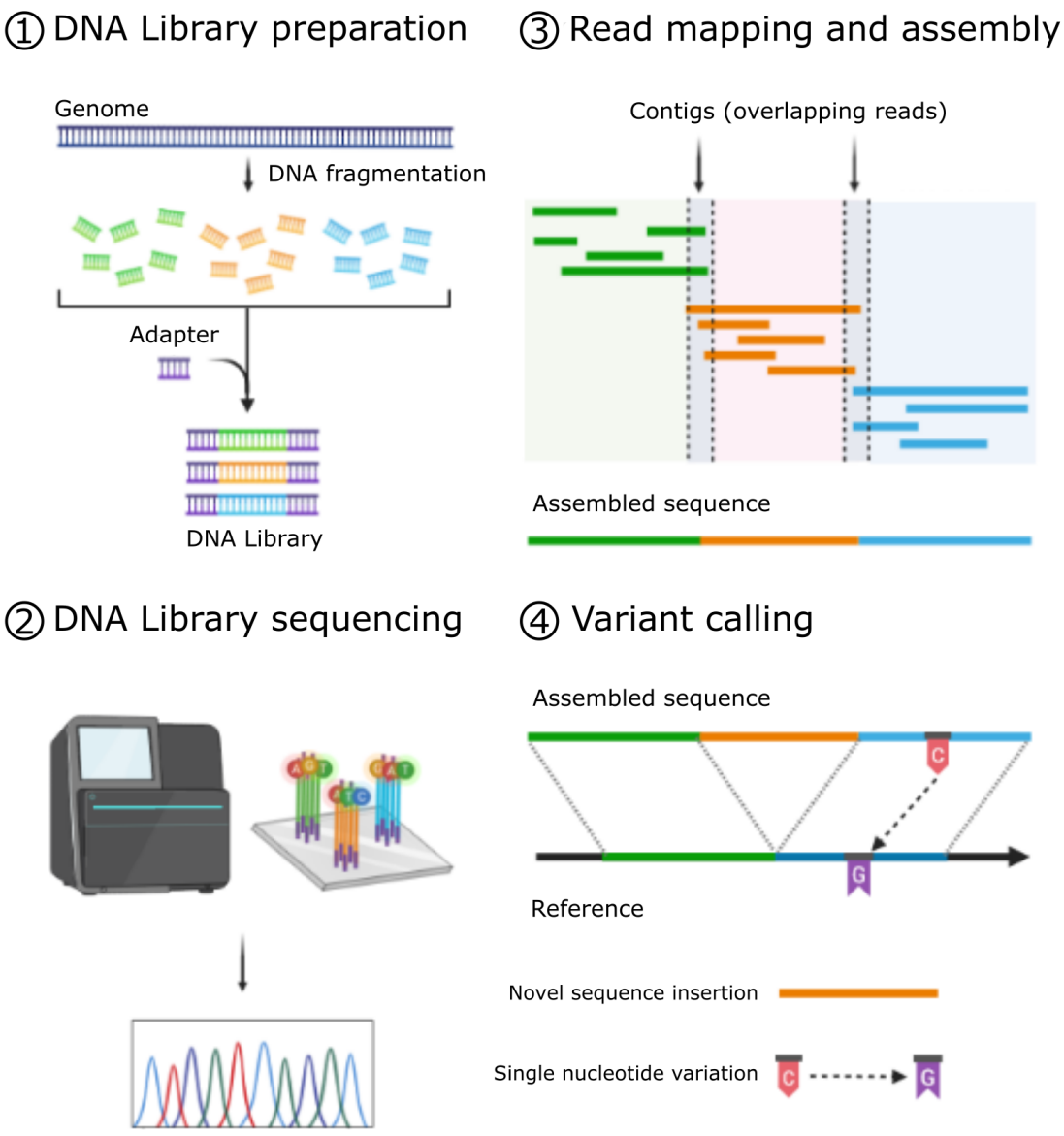


Figure 3.1: DNA sequencing and variant detection: Four stages in the process of DNA sequencing and variation detection. (1) The genome is fragmented into short fragments and a DNA library is prepared for sequencing by adding short unique adapters to each sequence. (2) The DNA library is sequenced in parallel using next generation sequencing (NGS) technologies and for each fragment the linear sequence of nucleotides is detected. (3) The short fragments, referred to as reads, are assembled into longer fragments (contigs) based on overlapping reads such that the whole sequence can be resolved. (4) The sequence is compared to a reference genome to detect variations. In this example the assembled sequence contains an inserted fragment (in orange) which does not appear in the reference genome, as well as a point mutation in one position, where **C** is changed to **G**. (Figure created in bioRENDER [https://biorender.com/])

to do so in the next years. The power of genome sequencing has enabled to retrieve information not only regarding the genome's sequence, but also allowed researches to develop many read counting applications, for example to measure the transcriptome (the set of RNA transcripts) [309] and the epigenome (the set of chemical modifications of the DNA) [220]. Due to the depth of the field it is infeasible to comprehensively review its influence on molecular biology. Here we will only focus on whole-exome and whole-genome sequencing, and in particular on the identification of variations in the DNA (Figure 3.1). In the next Sections we will briefly explain the workflow that is required for processing DNA sequencing data, and further elaborate on the procedure of mutation calling. In addition, we will shortly review the process of RNA sequencing (RNA-seq) as a central technology for measuring the expression levels of the transcriptome.

3.1.1 Whole-exome and whole-genome sequencing

While whole-genome sequencing allows for the determination of the complete DNA sequence, whole-exome sequencing identifies only the exonic regions, i.e. only the fragments of the genome that are known to be coding for proteins [125]. Consequently, whole-exome sequencing is much cheaper and produces lower amount of data per sample, which allows for the sequencing of more individuals, and by that increase the depth of a genomic study. Nevertheless, the strategies for processing the data remain the same. Most commonly the DNA is fragmented into overlapping short sequences, and the majority of sequencing platforms generate millions of reads, which must be aligned to a known reference genome or assembled *de novo*. The very first sequences, before a reference was completed, had to be assembled *de novo* [316] using complex algorithms which were able to aggregate the reads and produce the correct order [258]. Once a complete and accurate reference genome was built, new sequencing data could be directly mapped to it using sophisticated alignment algorithms [207].

Since the human genome is identical in more than 99.9% across individuals, it is feasible to detect differences between a reference and a sequenced sample. In comparison to a reference genome, an individual typically harbors 4–5 million single nucleotide polymorphisms (SNPs) and several hundred thousand short indels (insertions or deletions) [68]. Longer structural variants are more complex and therefore harder to detect [9], especially using short read technologies. They remain an on-going challenge, rapidly advancing due to improvements in long read technologies [261]. To date, a large number of individuals have been sequenced, and the cohort of sequences revealed common genetic polymorphisms in the human population [67]. These variations can help distinguish if a detected SNP is common or rare, and therefore determine if it could be associated with a disease phenotype.

3.1.2 RNA sequencing

Even though most cells of an organism contain the same DNA sequence, they usually vary greatly in the genes that they express, as those help control the specific function of the cell. Gene expression levels are primarily measured via the quantification of mRNA

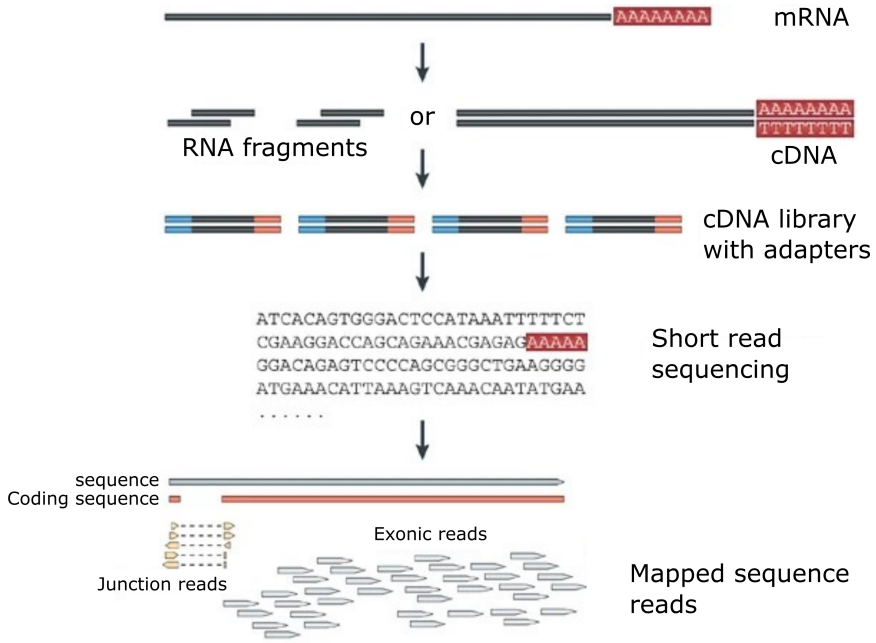


Figure 3.2: RNA sequencing steps: mRNA molecules are converted into complementary DNA (cDNA), and unique sequence adapters are added to each one. The molecules can be first fragmented and then converted or vice versa, as preparation for high-throughput sequencing. The resulting sequencing reads are then mapped to the DNA sequence, where they are identified as exonic regions (i.e. coding sequences) or junction reads (bordering intronic regions). Adapted from [309].

molecules, whose sequences correspond to the DNA sequence of the gene. The mRNA molecules are the processed transcripts of the genes that are also translated into the proteins that the genes are coding for. Thereby, measuring the gene expression levels via mRNA serves as a proxy for the level of the corresponding protein.

Microarray technologies first allowed to quantify the transcriptome [210] and nowadays high-throughput NGS is most commonly applied [197]. The process is described for example in Figure 3.2. In short, the mRNA molecules are first reverse transcribed, i.e. converted into a cDNA molecule which is comprised of the exonic regions of the gene's sequence. This cDNA molecule can then be sequenced via most of the NGS technologies, as previously described in Section 3.1 and Section 3.1.1. Since the mRNA contains only exonic regions, the sequence of the cDNA must be aligned to the reference genome using an alignment algorithm which is able to identify the entire region of the gene, even though the genome also contains the introns [309]. As genes can be transcribed into multiple mRNA molecules it is also possible to quantify the number of copies that were present at the particular time of the sequencing. To that end, each cDNA molecule is attached with unique adapters, e.g. short random sequences, which help to distinguish between different mRNA molecules of the same gene, and therefore facilitate the quantification of each molecule once the sequences are resolved.

This quantification is very useful, for example when comparing two different conditions based on the expression of the transcriptome, where one gene can be up- or down-regulated in one of the conditions in comparison with the other. This type of analysis, commonly referred to as differential expression analysis, can be applied us-

ing a statistical test which compares the expression of the genes under two conditions, where the null hypothesis is that there is no difference between them [183]. In addition, directly comparing the expression levels from different conditions can require a normalization step to account for technical effects, which arise due to differences in experimental environments as well as variations in sequencing platforms [48]. Once the expression levels are normalized the data is modeled, usually using a Poisson or a negative binomial distribution [326], and an appropriate test is applied to calculate the significance of change in expression between the two conditions. By now there are many methods which were developed for this purpose (see for instance the reviews in [70, 237]). Some popular examples are `limma` [244], `edgeR` [246] and `DESeq2` [171]. The results of those allow researchers to associate each one of the genes with the order of magnitude of the change, measured by fold change (FC), and the significance level, given by the P-value calculated using the statistical test.

3.2 Identification of disease genes from NGS data

While disease gene discovery was already possible prior to the sequencing revolution, there is no doubt that exome- and whole-genome sequencing capabilities advanced the field tremendously [23]. Older gene discovery strategies were mostly limited in their ability to study only a few individuals and to explain only a small fraction of the heritability of the disease using the detected gene [122]. Once a catalog of the common human SNPs was available [67], researchers could identify potential disease-causing SNPs by comparing the exome of an affected individual to the catalog [206]. In addition, the functional effect and possible impact of a variant could be predicted according to the type of mutation, i.e. whether it changes the protein or not, and the known role of the gene, such as participation in a relevant pathway [3, 205]. This strategy proved very powerful for many Mendelian disorders, which are usually governed by one gene only, however was not suitable for more complex diseases [43]. Larger scale whole-genome sequencing studies facilitated the detection of rare variants, which are more prevalent in common diseases [64] that are generally more complex and affected by multiple genes. On top of that, with the establishment of RNA-seq, the detected transcripts could also be used to identify variations in the protein-coding regions [59].

3.2.1 Genome Wide Association Studies

Once the technologies for sequencing the entire genome were widely available, genome wide association studies (GWAS) could be established. The main difference to previous experimental methods is the ability to sequence the whole genome, or at least the whole exome, and not being limited to specific regions only. The goal however remains the same: to identify regions in the genome that could be correlated with some phenotype, such that an association between a gene and the phenotype could be made. As these associations are only based on correlations they certainly need to be experimentally validated to demonstrate causation. For a (disease) phenotype in question, the experimental design usually requires a cohort of cases and matched controls, which can be

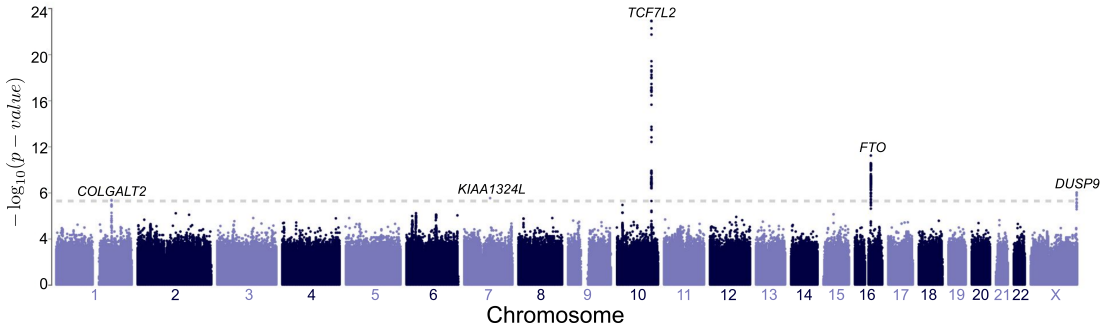


Figure 3.3: GWAS Manhattan plot: An example Manhattan-plot for the GWAS results from a study on type 2 diabetes, based on 8,126 cases and 30,917 controls. The X-axis denotes genomic coordinates along the 22 human chromosomes, and the Y-axis indicates the level of significance (measured by the negative logarithm of the P-value) of the association between one coordinate and the disease. The horizontal dashed line marks the significance threshold equivalent to a P-value of $5e-8$. The five regions which contain genomic positions that have a P-value above the marked threshold are labeled with their nearest genes. The figure was extracted from the PheWeb web interface [284] which is populated with the UK Biobank summary statistics [51].

compared to each other. The aim is to detect SNPs among the cases only, which are not present in the controls, and to find a correlation between those detected SNPs and the phenotype. A detected SNP can be within a gene's sequence, or in the vicinity of it, such that it is close enough to be immediately related to the gene. As a result, the gene is predicted to be associated with the phenotype. However, the SNP can also reside in a non-coding area of the genome, and therefore it is more complicated to relate it to a specific gene.

In order to detect a statistically significant correlation between a SNP and the phenotype, a statistical test must be applied. The test depends mostly on the measured phenotypic trait. For a review on the different tests see [50]. As the hypothesis of correlation is tested on millions of SNPs, the cohort under study must be large enough, and a multiple testing correction procedure must be applied. Eventually, every genomic position is associated with the phenotype at some statistical significance level, and only those which are strong enough serve as candidates for further studies. The results are usually visualized as a Manhattan plot, for example in Figure 3.3, where each genomic position is depicted with its significance level.

Over the last years there have been multiple GWAS studies which successfully associated many different diseases with genomic regions [285]. Since 2008 the GWAS catalog has been collecting results and variant-trait associations from publications, and to date it includes summary statistics from more than 5,000 studies [49]. However, there still remain several challenges in identifying genotype-phenotype associations. GWAS studies are only able to explain a small proportion of the heritability, i.e. the genetic components that govern the disease phenotype [83, 91, 180]. Moreover, the GWAS signals are not always tracked to the causal variants, as those might be in regulatory sequences which modify levels of expression or other regulatory mechanisms downstream of the identified variants [265]. And finally, despite the power of exome sequencing, the genetic effect is still only identified for 25% of the patients when di-

agnosing a Mendelian disorder [324]. This could be due to large heterogeneity, with many rare variants, which are less common, affecting only a small fraction of the individuals. All of the above highlight the need for incorporating more data and conducting further analyses in order to identify the causal mechanisms of a disease.

3.2.2 Identifying cancer genes

Since cancer is mostly driven by genomic changes in somatic cells, such as SNVs and CNVs, the predominant technique for characterizing cancer mutations is via sequencing of the DNA. Even though to date there have been many studies which attempted to elucidate cancer mechanisms via other cellular measurements, such as gene expression levels and DNA methylation quantification, our main focus here will be on the characterization of cancer genes via DNA sequencing. While the very first cancer mutation studies were able to identify a small group of mutations via older sequencing techniques [279], advancements in sequencing technologies have profoundly increased the amount of data and facilitated the identification of many cancer mutations and cancer genes [192]. The volume of different cancer sequencing projects to this day is truly immense. At first the projects were smaller and helped to characterize the mutational landscapes of specific types of cancer [69]. Very quickly, the projects evolved into larger pan-cancer projects [314], integrating data from multiple types of cancer. So far most of the projects generated only exome-sequencing data [200], which is typically restricted to coding regions only, however just recently whole-genome sequencing of 38 tumor types have been made available by the Pan-Cancer Analysis of Whole Genomes (PCAWG) consortium [286].

There are many challenges to consider when trying to identify cancer mutations based on sequencing data and the research on the topic has been extensive. For the purpose of this work we will only provide a very primary summary. We will focus on The Cancer Genome Atlas (TCGA), one of the most comprehensive cancer projects in the last years, and exemplify how cancer mutation data is often collected and analyzed. TCGA is one of the largest efforts to collect and analyze cancer genomics data from more than 10,000 patients spanning over 33 types of cancer [135]. With the aim of characterizing cancer on all molecular levels, data was mainly collected for SNVs, CNVs, mRNA expression, and DNA methylation [123, 124]. For each data type an extensive analysis workflow had to be established. Here, we will only focus on the details that are relevant for the identification of somatic mutations. The data was generated using whole-exome sequencing, and was eventually analyzed using one centralized mutation calling pipeline [84]. This analysis enabled the identification of 3.5 million somatic variants, which are based on seven mutation-calling algorithms. The establishment of such a centralized pipeline also allowed researches to analyze the data on a pan-cancer level, i.e. summarizing evidence from all 33 cancer types, in order to identify cancer genes [22].

The first step in identifying somatic mutations from sequencing data is calling for variants, i.e. detecting mostly SNVs and small insertions/deletions in the tumor samples, and comparing them to matched normal samples. On top of that, it is also possible to identify other changes such as CNVs [329]. To date, there exist many algorithms that

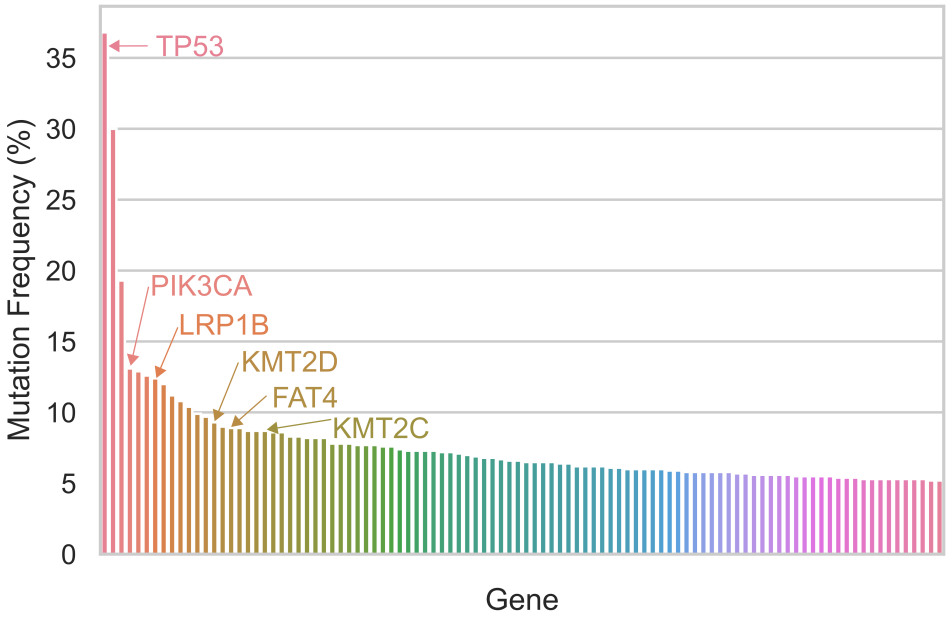


Figure 3.4: Long-tail distribution of cancer mutations: The mutation frequency for the 100 most frequently mutated genes among 10,437 samples across 32 pan-cancer TCGA studies (data downloaded from the cBioPortal for cancer genomics [55, 97]). Among the top 20 genes are six well-known cancer genes (according to the cosmic cancer census list [273]). The mutation frequencies for all genes (n=18,470) range between 36.8 and less than 0.1, with a median of 0.7.

were specifically developed for the detection of somatic mutations [87, 307]. However, despite major advancements, there are still challenges in detecting alterations, and the results of the different methods can vary greatly [150]. This is due to remaining technical problems with sequencing techniques and alignments, as well as the nature of somatic mutations, which are hard to detect due to unique properties, such as low frequencies, local CNVs and tumor subclonality [87].

Once a final set of variants is compiled, further analysis can be applied to identify cancer driver genes, i.e. genes that are mutated and have a mechanistic effect on cancer development and progression [288]. This requires to distinguish driver mutations from passenger ones, which occur incidentally and do not drive development or maintenance of tumor cells [106]. This step is necessary as in cancer the majority of genes is mutated in at least five samples [161], a phenomenon which has been termed 'long-tail' [98], due to the long-tail distribution of mutation frequencies, and is illustrated in Figure 3.4. Thus, assessing the mutation frequencies and identifying those that occur significantly more than expected is key, and there are many statistical methods which attempt to do so (see for example the review in [287]).

Mutation frequencies can vary greatly, both among patients with the same tumor type, and also across tumor types, hence the variation must be accounted for when identifying driver genes [162]. However, due to the high differences in mutation frequencies, it is still challenging to identify those cancer genes which are very rarely mutated, as they are not detectable after statistical adjustments [161]. In addition to

assessing the mutation frequencies, it is also useful to estimate the functional effect of the detected variants based on existing annotations and predictions related to the corresponding protein [104]. On top of that, many methods that integrate prior knowledge, such as from molecular pathways and networks [72], and/or other types of data [6, 224, 281] have been developed in aim of characterizing driver genes. Since the collection of algorithms is very large we refer the reader to some reviews on the topic [57, 77, 238].

3.3 Computational methods for identifying disease genes

With the advancements in experimental methods for disease-gene identifications arose also many computational methods for analyzing the data, mostly with the aim of generating *in silico* predictions of disease-gene associations. These tools vary in both the type of data that is being analyzed, as well as the computational approach for prioritizing or ranking the novel predictions. The results of such tools help focus the experimental studies on promising candidates and allow for further explorations and functional validations.

Initially, the computational tools were developed to help target the search for genes, given a larger genomic region that was identified via early methods such as linkage analysis or Homozygosity mapping [82, 262]. Later, by using existing information about genes that were already discovered to be associated with some disease, researchers could compare disease genes with non-disease genes, based on sequence properties or functional annotations, such as gene ontology (GO) terms [18], and generate novel predictions [2, 132, 174, 185, 271].

Once high-throughput sequencing data became widely available, more and more methods, that also integrate multiple types of data, could be developed. The data is usually highly heterogeneous and constitutes measurements of sequences and mutations, gene expression levels, protein interactions as well as functional annotations. Each type of information can help elucidate different aspects of the disease-gene association, and therefore there is a great variation of methods that contribute different insights about the mechanisms of a disease. Integrating together multiple types of data is helpful in accumulating evidence and providing more comprehensive insights.

By now there is an abundance of tools, and there has been already considerable efforts to summarize and review them (see for example [147, 196, 232]). Here we will mention a few of the tools that are relevant for the scope of this work, with a particular focus on methods that utilize PPI networks.

The first group of methods is mostly based on results from text mining of scientific journals, which can be broadly used for retrieving evidence for genes, diseases and cellular processes [156]. In order to identify new disease gene candidates, a search that is based on the disease characteristics, or a set of already identified disease genes, is executed and a statistical assessment is applied to the retrieved associations [294, 328].

The second group of methods attempts to identify novel genes based on varying features of other genes which have already been associated with the disease. The charac-

terization can be based on sequence similarities [101], gene expression profiles [215], functional annotations [99] and more. Some methods also integrate the evidence from different types of data and combine the results for generating a final prioritization [4].

The last group of methods is based on the analysis of interaction networks, which enable new predictions based on the connections between genes. The interactions can be experimentally measured, such that a network is constructed from the data, and different network measures can be estimated [90]. Alternatively, existing interaction information can be represented using a network, which can then be analyzed in combination with the experimental data [216].

3.4 Network based methods

Molecular networks are commonly used to help solve many different questions [188]. One of the main concepts in network analysis is the guilt-by-association principle [211]. Originally, it was exploited to predict the function of a protein, by looking at the function of other proteins it interacts with [260]. In the same manner, if a protein is associated with a disease, it is likely that its interaction partners will also be associated with the same disease [163]. Thereby, networks can be a very useful tool for predicting novel disease-gene associations. Furthermore, if a gene is known to be mutated, and found associated with a disease, it can be very useful to view it in the context of an interactome, since other genes, which are not necessarily mutated, might also be affected, and therefore contribute to the disease phenotype. This is particularly useful in the case of cancer, where some genes are mutated at very low frequencies and therefore could only be detected as relevant when examined in the context of an interaction network [58].

The most basic methods explore only the direct neighborhood of disease genes [216]. Other methods utilize the topological measures in the network to predict novel disease genes [159, 320, 323]. In order to exploit information from the entire network different methods that are based on the concept of diffusion or network propagation were also explored. Köhler et al. [158] first introduced random walk and diffusion-kernel for the identification of disease genes. Later, Vanunu et al. [297] used a propagation-based algorithm to infer associations over the entire network, and, using prior knowledge of disease similarities, generated novel predictions. As it is an essential part of this work, we will further elaborate on propagation-based methods in Section 3.5.

Another approach for identifying disease genes is by first identifying network modules, which can then be associated with a disease [214]. It has been previously shown that disease genes are not randomly spread in the network, but rather tend to be close to each other [26, 191]. The connections between the genes essentially form sub-graphs within the network, which are usually referred to as disease modules (Figure 3.5). Such disease modules might include genes that were not experimentally detected, yet are affecting the disease phenotype. The identification of network modules in general, and disease modules particularly, is extensively studied, with many suggested solutions (see for example the review in [203]).

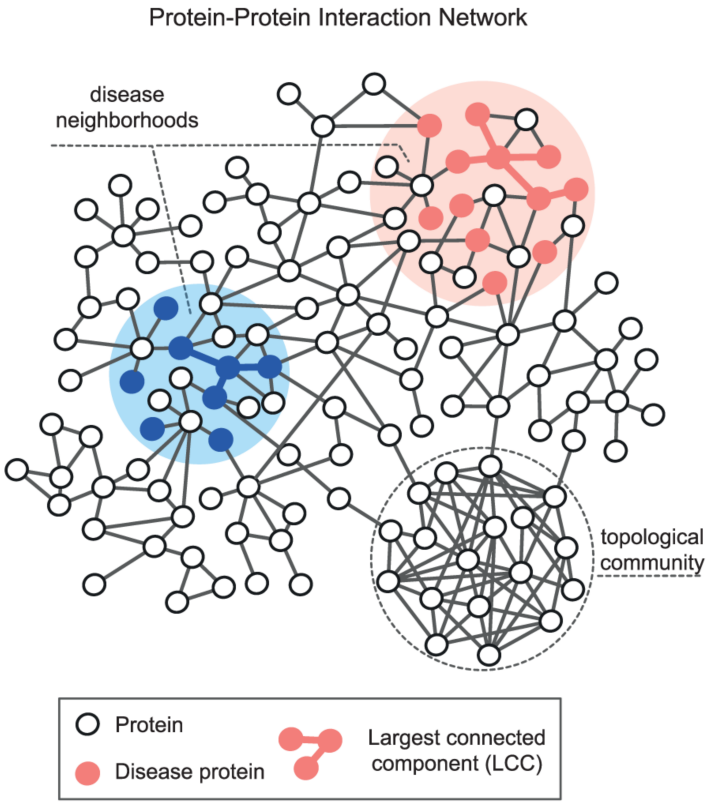


Figure 3.5: Network modules in disease: Proteins that are associated with some disease tend to localize within the same network neighborhoods, i.e. in close proximity in the network. The proteins can be connected via intermediate nodes to form disease modules. In contrast, some proteins are organized in topological communities, i.e. regions in the network that are densely connected, and might be related to some function or phenotype. Adapted from [102].

3.4.1 Identification of network modules

As molecular networks are usually very large, it is impractical to perceive or visualize the entire network, and therefore identifying smaller parts of the network that can then be associated with some function or disease can be very beneficial. Generally, molecular networks tend to be very modular [110, 115, 245], which means they can be partitioned into smaller modules, that usually represent some functionality. Thus, a very common problem when analyzing molecular interaction networks is the identification of modules. These modules (see definitions in Section 4.1.1) are sub-graphs of the network that most commonly represent biological processes and pathways. Since many genes can be involved in one phenotype or disease, yet also one gene can be related to more than one function, the identified modules can be overlapping and related to many biological processes. The modules can be purely topology based, such that they include nodes that are densely connected among each other, but less connected to the rest of the network. A functional module will include nodes that are both in proximity as well as share some functionality. A disease module will include nodes that share similar functionalities such that a disruption of them, or their interactions, results in a

disease phenotype (Figure 3.5).

Here we will focus on methods which aim to identify functional and disease modules, and so usually combine both network information together with experimental data. The problem of identifying modules solely based on the topology of the network, sometimes also called community detection or graph clustering, was recently assessed in the Disease Module Identification DREAM Challenge [62], and serves as a comprehensive overview on this problem.

The identification of functional modules and disease modules by integration of experimental data together with a molecular interaction network, aims to outline regions in the network which are significantly enriched in experimental evidence. The problem was first formulated in the development of the `jActiveModules` tool [136], where the goal was to identify significantly high scoring sub-networks in molecular interaction networks. Accordingly, methods that aim to find such modules must first summarize the experimental evidence into scores or weights, which can then be assigned to each node in the network, such that some optimized search for modules can be applied.

To date, there are many different methods that try to address the problem, and they vary both in the way they score nodes, as well as in the computational approach for extracting modules [321]. However, identifying the correct modules is a computationally hard problem [136], and thus most methods apply some heuristic solutions, which might be sub-optimal but still relevant in practice [194]. Nevertheless, there are some methods that aim to provide exact solutions, and despite the computational costs can be executed in relatively short times. Some examples are the `BioNet` algorithm [34, 78], the branch-and-cut approach [19] and recently also `NetMix` [241]. Another set of methods is focused on the diffusion or flow of data in the network, rather than the optimization of the problem. Propagating the data through the network can essentially "smooth" the information and help identify modules that accumulate most of the flow. As network propagation is at the heart of this work, these methods are further discussed in the next Section.

3.5 Network propagation for genotype-phenotype associations and module identification

Network propagation commonly refers to the diffusion of information or flow in an interaction network. The mathematical foundations of network propagation, which are based on random walk processes or diffusion kernels, are provided in Section 4.2. The concept of propagating information through a network is widely used in many fields, for instance to model the flow of electricity [81] or to rank web pages, as in Google's PageRank algorithm [217]. In molecular biology, it was initially applied to identify unknown members of a protein complex in PPI networks [52], to detect protein homologs [208] and to predict protein function [199]. Later, the same approach was also applied to the problem of prioritizing and identifying disease genes [158, 297]. These methods, that were based on random walks in PPI networks, were later surveyed and compared with other computational methods for disease-gene predictions, and were found to outperform other clustering and linkage-based tools [201]. Since then, various approaches

that are based on network propagation were developed for associating genes with different phenotypes, as well as for identifying gene modules in interaction networks (see the review in [71]). The main advantage of network propagation methods is their ability to use the topological information that is encoded in the network and combine it with prior knowledge or experimental evidence. In this way, both resources are simultaneously exploited and accurate predictions can be made.

3.5.1 Overview of network propagation methods

Recent methods that are based on network propagation generally differ in their main aim and the data they analyze. The aim can be the prioritization and prediction of disease genes or the identification of disease modules. Figure 3.6 illustrates the general framework and the various types of analyses. The data is usually focused on a disease phenotype, and can be extracted from different types of experiments. Most of the methods use a PPI network for modelling the interactions, although other kinds of networks are applicable too. Due to the abundance of PPI resources, the chosen network usually varies between the methods. Several methods have previously suggested to use multiple interaction networks, by applying the propagation on each network separately, and integrating the results to increase the confidence in the predictions [61, 226].

In general, the methods can be divided into two groups. The first group of methods aims to prioritize disease genes and generate novel predictions of disease-gene associations. Initial approaches extracted genotype-phenotype associations from curated resources, for instance OMIM [12], and smoothed the prior knowledge in the network using random walks [168, 269]. Once whole-exome sequencing data became available, it could also be incorporated for the identification of mutated genes that could be associated with a disease, as applied in `ExomeWalker` [270]. Similarly, gene expression data could also be incorporated in the propagation process. In `NetWalk` [152] the expression levels were used to set the transition probabilities of the random walk, whereas in `RegMode` [236] they are used as the initial scores or weights for the propagation.

The second group of methods is aimed at identifying network modules, in addition to generating new predictions. In order to identify such modules, it is usually required to apply another computational procedure or to incorporate functional information. Those will be described in Section 3.5.3.

3.5.2 Random walk with restart - formulations and applications

Even though all propagation methods are based on the same concept, the implementations of the procedure differ, as well as the computational approaches for identifying modules. In this work we will focus on the random walk with restart (RWR) implementation of network propagation. This implementation differs from the random walk one as it allows the walk to be restarted, i.e. reverted to the initial state and only then continued. As a result, the final score for each node reflects a combination of the input information and the topology of the network. The implementations and their differences are discussed in Sections 4.2.1-4.2.2. Here, we will further review in detail

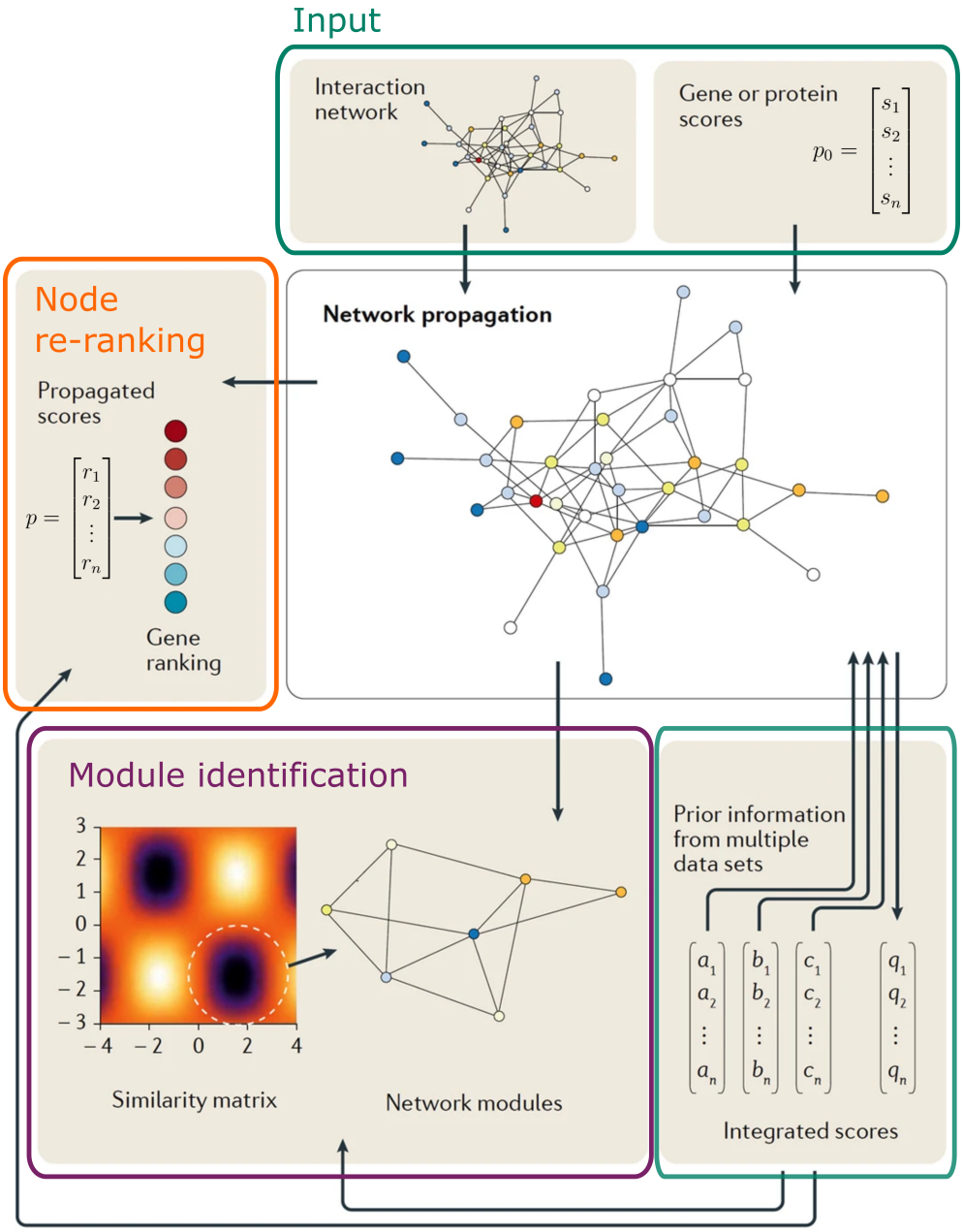


Figure 3.6: Network propagation framework: Methods which are based on network propagation are typically applied on two sources of input: an interaction network and a vector with scores for each node (gene or protein) in the network. Some approaches might integrate multiple scores and apply the propagation more than once. After the propagation two types of output are possible. The first is the re-ranking of the nodes, which is a vector with a propagation score for each node in the network. The second is network modules, which can be identified based on a similarity matrix that is extracted at the end of the propagation. Adapted from [71].

methods that apply the RWR implementation, discuss the different variations and refer to remaining challenges.

To initialize the RWR process, genes in the network must be assigned with input weights. The weights can be derived from prior knowledge, such that only genes that are already known to be associated with the phenotype are scored. In the basic implementations binary weights are assigned, i.e. well-known disease genes are scored with 1 and the rest of the genes in the network with 0. Other implementations calculate a different score based on the prior knowledge. For instance, Köhler et al. [158] extracted disease-gene associations from OMIM [12] and assigned equal probabilities to all the nodes in the same disease, such that the sum of their weights was 1. Vanunu et al. [296] also used OMIM disease associations, but scored the nodes according to a similarity metric between diseases, i.e. based on the fact that a node might already be associated with another similar disease.

Alternatively, the weights of the nodes can also be initialized based on experimental evidence, for example gene expression levels or mutation frequencies. The data must be summarized such that each node is assigned one weight, which usually requires some pre-processing and statistical analysis of the experimental values. One might choose to initialize weights based on the entire data set, i.e. for all the nodes in the network, or only for a sub-set of nodes, for example only those that passed a minimum significance level. It is also possible to use the P-values themselves, instead of the raw or normalized data, for the initialization of the weights. Carlin et al. [53] scored genes with their significance level that was calculated based on GWAS data. Cancer mutation data is also often statistically summarized into weights and used in many RWR applications [166, 242, 295]. Ruffalo et al. [251] used both raw mutation counts and differential gene expression levels, applied RWR twice, for each data set separately, and used the propagated information to generate features for a logistic regression model, in order to predict if a gene is causal in breast cancer.

Several modifications to the formulation of the RWR have already been suggested. `RWRH` [168] was developed as a RWR process for a heterogeneous network, i.e. the walk can jump between multiple interaction networks. Zhang et al. [332] replaced the interaction matrix with a weighted matrix, which reflects different types of evidence about the interaction. Recently, Lee et al. [164] created an interaction network that combines both PPIs and pathway information and applied network propagation in order to link transcription factors to cancer pathways. In `NetWalk` [152] the experimental values are used not only for assigning weights, but also for setting the restart probability separately for each node, such that it is proportional to the evidence. In a similar way, Jin et al. [141] developed Random Walk with Extended Restart (RWER), which also allows for the calculation of a distinct restart probability for each node in the network, using a supervised algorithm to learn the restart probabilities, given the interaction network. Ahmed et al. [5] developed `MEXCOwalk`, where the random walk probabilities for every pair of genes were defined based on their mutual exclusivity and coverage in cancer mutation data. Very recently, Hristov et al. [130] suggested a guided random walk, where the probability of walking to a neighbor depends on its network proximity to a pre-defined set of known disease genes.

3.5.3 Module identification based on network propagation

In the case of network propagation, and specifically for the RWR implementation, there are two general ways to identify modules post propagation. The first approach is by overlaying the weights after the propagation back onto the nodes in the network, and applying some other computational method to extract modules. For example, in `PRINCE` [296] protein complexes were identified in an iterative procedure which was based on the prioritization scores that were calculated using a propagation-based algorithm. Another example is `TieDIE` [223] where propagation is applied twice, for two different data types, and sub-networks are identified by extracting minimal paths from highly scoring genes in one set to the other. In the second approach, the propagation is computed such that instead of extracting weights once the random walk is converged, a similarity matrix for all the nodes in the network is calculated using the RWR formulation and the initial weights. This allows for the computation of a similarity score for every pair of nodes in the network, which can then be used to identify network modules. For example, both `HotNet` [295] and `HotNet2` [166] compute a similarity matrix, create a directed graph from those node pairs which are highly similar, and search for connected components in this graph. `Hierarchical HotNet` [242] applies an hierarchical clustering algorithm on the computed similarity matrix to identify sub-networks. Similarly, in network-based stratification (NBS) [126] consensus clustering is achieved by using the unsupervised technique of non-negative matrix factorization (NMF).

3.6 Motivation for a revised computational model

By now, network propagation is a popular method of choice for identifying disease genes and modules. There are at least 17 software tools which implement various propagation variations for different purposes [71]. While many methods have been successful in identifying both novel disease genes and modules, some fundamental shortcomings still remain. Most methods use node degree in the propagation formulation to generate the walking probabilities. However, in PPI networks node degree suffers from some biases, which will be elaborated in the following Section. Consequently, the node re-raking after the propagation will also be biased, which could hinder the results and might even generate false predictions. Therefore, there is a need to address this bias and modify the propagation model appropriately.

In addition, there are still some issues when identifying network modules based on the re-ranking of the nodes. Some methods, for instance Network Assisted Genomic Association (NAGA) [53], produce modules simply by taking the top nodes after the re-ranking. This is problematic since a) the nodes are not necessarily directly connected in the PPI, and b) the new ranking is not statistically tested. Other methods (e.g. [38, 86]) might apply a statistical test to evaluate the re-raking, however, they do not provide a process to identify network modules which is based on the significance levels. Furthermore, methods which compute a similarity matrix, like `HotNet2` [166] and `Hierarchical HotNet` [242], produce modules where nodes might be directly connected, even though such connections do not exist in the PPI network. Finally, so far most methods propagated information either from well-known disease genes or from

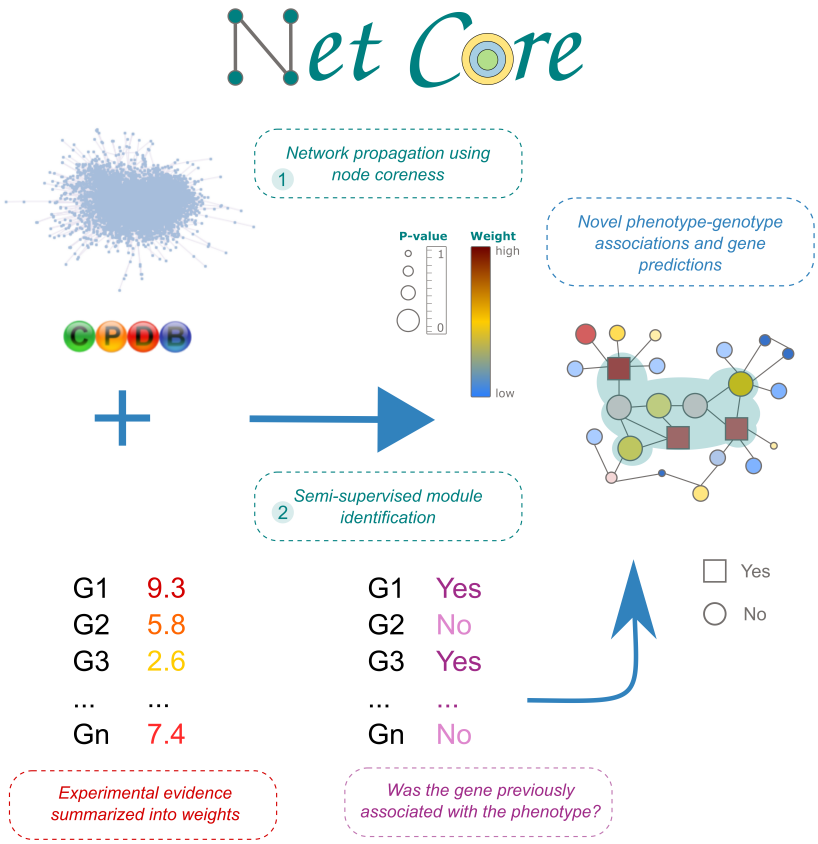


Figure 3.7: NetCore - a revised network propagation model: Our revised network propagation model NetCore provides two novel contributions: (1) it applies network propagation using node core and (2) it implements a semi-supervised module identification procedure. Both facilitate the prediction of novel genotype-phenotype associations and the detection of relevant network modules.

experimental evidence. Yet, combining both could facilitate the identification of disease-relevant modules.

The next Section will provide further details on these enumerated shortcomings of network propagation and the identification of disease modules. These also serve as the main motivation for the development of our method NetCore, illustrated in Figure 3.7. NetCore, which aims to address these shortcomings, will be described in detail in the next part of this thesis.

3.6.1 Study bias in PPI networks affects network propagation

In Section 2.4 we described the yeast two hybrid system for measuring PPIs. This system proved to be very useful in identifying many binary interactions between proteins, however it also introduced some study biases. The experimental design may result in some systematic technical biases. Interactions involving the 'bait' proteins are more likely to be measured than interactions involving they 'prey' proteins [278]. This means that 'bait' proteins will have more interactions in the PPI network, and therefore

a higher degree. On top of that, the number of interactions when using a protein as 'bait' is not always the same as when using it as 'prey' [11]. This is especially problematic as there is usually also a selection bias concerning the 'bait' proteins, i.e. proteins that are more commonly studied are more often selected as 'bait' [198]. In fact, there is a positive correlation between the degree of a protein and the number of times it has been screened for interaction partners [109, 257]. Moreover, proteins which are mentioned more often in publications are highly interconnected, whereas less studied proteins are also less connected [248]. It is especially prevalent in proteins that are associated with diseases, and particularly true for well-known cancer genes [257].

The aforementioned study bias problem in PPI networks is highly relevant for the formulations of network propagation (see Section 4.2). In these formulations, most commonly, the degrees of the nodes determine the probabilities of walking from one node to another. Therefore, high degree nodes will be visited more often, which will result in a larger weights for them after the propagation. Indeed, many network propagation methods will report mostly high degree nodes and rank them higher than nodes with a lower degree [119]. Recently, Picart-Armada et al. [229] examined the effects of statistical normalization on network propagation results and found that unnormalized scores are indeed more biased than different versions of normalized ones.

To address this study bias in network propagation results, two general methods for statistical adjustments have been previously proposed. Erten et al. [86] first developed `DADA`, a suite of degree aware statistical adjustments that can be applied to network propagation results. Given a set of seed nodes, they proposed three different reference models for adjusting the scores of the seed nodes after the propagation. Later, Biran et al. [38] developed a normalization based on random degree-preserving networks (RDPN). This approach, as opposed to the ones developed in `DADA`, is based on randomizations of the input network, rather than of the input seed nodes, and was shown to outperform `DADA`. Other propagation-based methods, such as `HotNet2` and `Hierarchical HotNet`, have applied similar approaches in their statistical significance test, however those were directly incorporated to evaluate the significance of their reported results, and do not provide a general solution.

Even though it is already possible to address the study bias via statistical normalizations, there is still no proposal for addressing it directly within the propagation formulation. In the current formulation, most commonly, the degree is used for adjusting the adjacency matrix (see Section 4.2.1). We postulated that a different modification, based on a less biased node metric, could further reduce the bias in the propagation results. Once a different adjustment to the adjacency matrix is available, the propagation results could still be statistically tested, as previously suggested by others. Subsequently, based on these unbiased propagation weights, a new prioritization of the genes would be achieved. Furthermore, network modules could then be identified based on the propagation weights and their significance level.

3.6.2 Drawbacks in module identification for network propagation

Addressing the study bias would help generate more accurate predictions, however there still remains the challenge to identify network modules. The RWR based methods that attempt to identify modules still have some drawbacks. First, most of the methods that search for modules based on the prioritization after the propagation apply some threshold when choosing the nodes for the modules. However, this threshold is usually only set to select the top ranking genes, and is not statistically significant. Second, the methods that identify modules based on a similarity matrix might create false interactions that do not exist in the PPI network. Since they calculate a similarity score for every pair of nodes in the network, they artificially connect all the nodes before applying some computational approach to extract modules according to the calculated similarity matrix. Although all the nodes in the network are connected through at least one path, they are not all directly connected, and therefore the modules might include false direct connections between nodes. Hence, there is a need to establish an approach that could generate network modules based on propagation results, such that the included nodes are weighted with a statistically significant value and the connections between the nodes are only based on the existing interactions in the input network.

Since increasing amount of information is accumulated for many diseases and phenotypes, we proposed to apply a semi-supervised approach for identifying network modules. So far, most of the propagation-based methods generated new predictions and identified modules either based on known disease genes, or experimental evidence. However, there has not yet been an attempt to integrate both within the network propagation framework. Only very recently Hristov et al. [130] proposed a guided network propagation scheme, such that data is propagated towards known disease genes, and by that prior knowledge is incorporated within the propagation to improve disease-gene predictions. Our aim is to combine the predictive power of network propagation together with prior knowledge in order to detect not only novel disease genes, but also disease modules, which connect well-known disease genes with novel predictions in one sub-network. Due to the connectivity in these disease modules, they would be more comprehensive and enriched in functions and pathways that could be relevant to the disease mechanisms.

4 Fundamental Concepts

This chapter includes some fundamental concepts which are used throughout this work. The basic definitions in graph theory are provided, as well as the mathematics of network propagation. There are many more concepts in graph theory that can be applied to molecular interaction networks, as described, for example, in the book *Networks: an introduction* by M. Newman [202].

4.1 Graph theory

Graphs are mathematical models that are used to describe pairwise interactions between objects, for example genes or proteins. A graph G is defined by a set of vertices (also called nodes) V which can be connected by edges $E \subseteq \{V \times V\}$, such that $G = \{V, E\}$. The graph can be **directed**, where edges are asymmetric, i.e. $(v_i, v_j) \neq (v_j, v_i)$. In an **undirected** graph, these two edges are equivalent $(v_i, v_j) = (v_j, v_i)$. A **weighted** graph is a graph where each edge is also assigned with one value $\{\forall(v_i, v_j) \in E|(v_i, v_j) = w\}$. Figure 4.1 shows examples of (un-)directed and (un-)weighted graphs. A **connected** graph, in the undirected case, is a graph where there exists a path from any node v_i to any other node v_j. In the directed case, the graph is **strongly connected** if there is a directed path between any pair of ordered nodes, or **weakly connected** if there is only an undirected path between them. A **complete** graph is a full graph, i.e. a graph where there is an edge between every two nodes in the graph. If the number of nodes in the graph is n, then the maximum number of edges is $n(n-1)$ in the directed case, and $\frac{1}{2}n(n-1)$ in the undirected case.

Graphs are usually represented in an **adjacency list** or **adjacency matrix**. An **adjacency list** stores for each node in the graph a list of its adjacent (also called neighboring) nodes. An **adjacency matrix** is a two dimensional matrix, where each dimension represents a list of all nodes. Binary values imply whether there exists an edge between two nodes. Alternatively, the values can represent the edge weights.

4.1.1 Sub-graphs and modules

A sub-graph F is a graph that is generated using another graph G, by taking a sub-set of nodes and edges from G. If a sub-set of edges $\{e_1, .., e_m\}$ from G is chosen for F, then all of the nodes that connect these edges in G must also be included in F, in addition to other nodes from G that might also be added to F. An **induced** sub-graph, given a group of nodes $\{v_1, .., v_n\}$, will include all of the edges that connect the chosen sub-set of nodes. A **spanning** sub-graph includes all of the nodes from G, but not necessarily

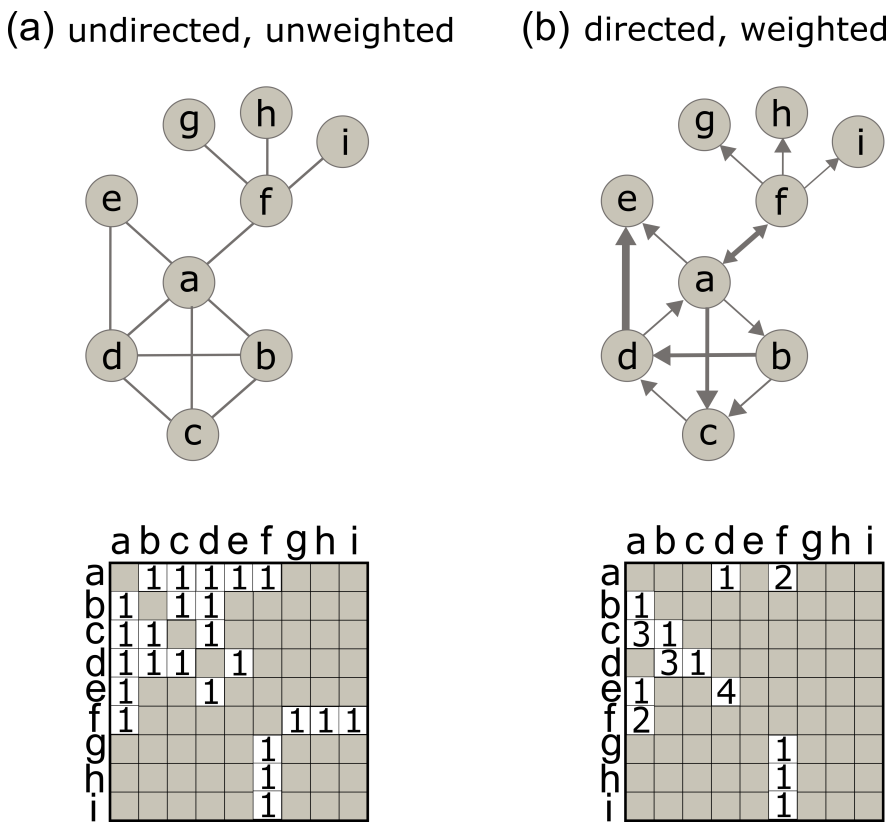


Figure 4.1: Graph models: An example graph with nine nodes and 12 edges. (a) is undirected and unweighted whereas (b) is directed and weighted. The adjacency matrix for each version indicates both the directionality and the weights of the edges.

all of the edges. A well-studied example of sub-graphs is a **clique**, which is a sub-set of nodes that are all connected to each other.

Sub-graphs are commonly used in many graph algorithms, as well as for defining different graph and node properties. An important definition that emerges from sub-graphs is that of graph components. The distinction between **directed** and **undirected** graphs here is imperative. In an **undirected** graph G, a **connected component** is a sub-graph of G that includes a sub-set of nodes such that every two nodes are connected in a path, and nodes outside of the connected component are not connected by a path to nodes within. When the graph G is fully connected, all of its nodes belong to the same connected component. A node that is not connected to any other node in the graph defines itself one component of size one. Figure 4.2 shows an example of an undirected graph which has two connected components. In a **directed** graph G, a **strongly connected component** is a sub-graph of G that includes a sub-set of nodes such that there exists a directed path between every two nodes. A directed graph that is fully connected includes one strongly connected component.

In addition to sub-graphs, a graph G can also be divided into sub-sets of nodes called communities or **modules**, which usually represent a group of nodes which are similar to each other according to some measure. Whereas the components of a graph are non

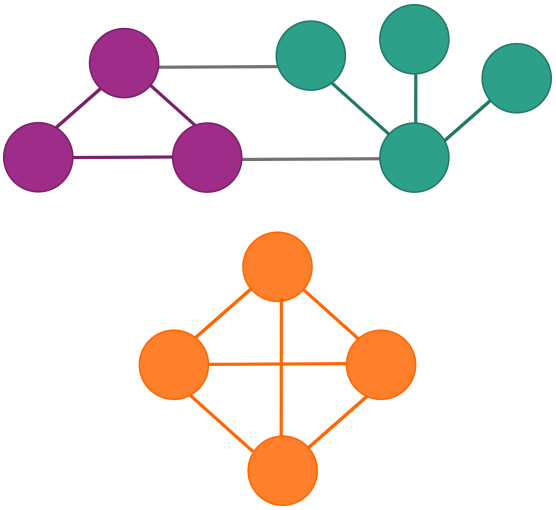


Figure 4.2: Sub-graphs and modules: An example of an undirected graph with 11 nodes and 14 edges. The graph contains two connected components. The first component (orange nodes) is also a clique, as all the nodes are connected to each other. The second component (purple and green nodes) can be divided into two sub-graphs, according to color. The purple sub-graph may also define a network module, as there are more connections between purple nodes than between the green nodes.

overlapping, and essentially represent a partition of the graph, modules are sub-graphs that can possibly overlap. The modules can be derived solely based on the topology of the network, but also based on labels or weights that are associated with the nodes or the edges. The nodes in the module are typically close to each other, i.e. there should be a short path between every pair of nodes. Furthermore, usually the density of edges among members of the module should be higher than to non-members.

The problem of identifying graph modules, sometimes also referred to as **community detection** or **graph clustering** (not to be confused with graph clustering coefficient), is in fact not possible to solve in a feasible time [44]. Even if one would like to identify only two non-overlapping modules, the number of possible partitions would be $\frac{n!}{n_1!n_2!}$ given n nodes, and modules of sizes n_1 and n_2 such that $n = n_1 + n_2$. Therefore, most algorithms must apply some heuristics in order to detect network modules, which then result in different modules depending on the algorithm that was executed. Consequently, it is also difficult to compare and asses the performance of such algorithms. One way to asses the so-called quality of the modules is by calculating their modularity, i.e. compare the fraction of edges within modules to the expected fraction given that the edges were random. Graph **modularity** [204] is therefore designed to measure how well the graph can be divided into modules. A graph with high modularity is a graph that can be partitioned into modules such that the probability of two nodes from the same module to be connected is higher than two nodes form different modules. Further details on the definitions of graph modules and the approaches for detecting them are available in the review by Fortunato and Hric [89].

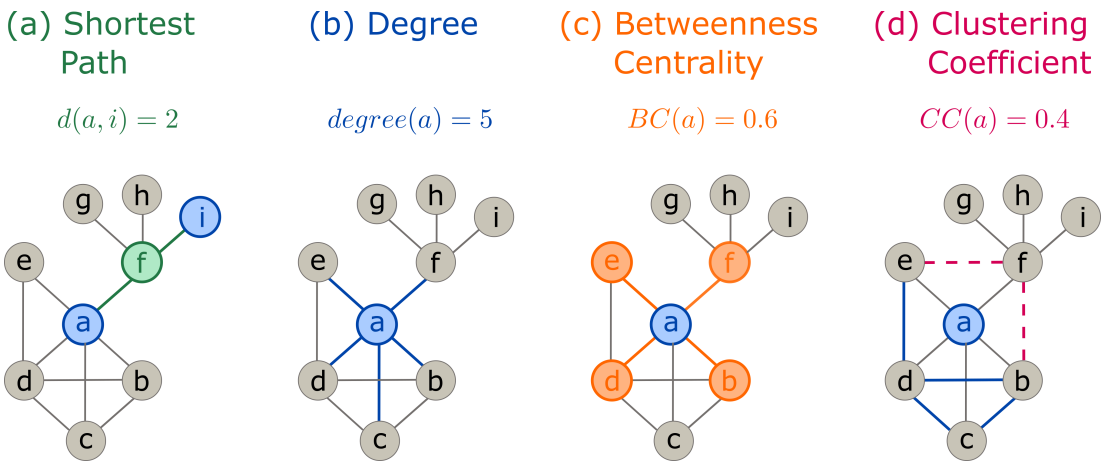


Figure 4.3: An example graph with 9 nodes and 12 edges. (a) The shortest path from **a** to **i**, which goes through **f** only, is of length 2. **Node metrics and measures:** (b) The node degree of **a** is 5 as it is connected to five other nodes. (c) The betweenness centrality (BC) measures how many short paths go through **a**. For example, the shortest paths between **b** and **e** and between **f** and **d** go through **a**, respectively. (c) The clustering coefficient (CC) measures the number of connections between the neighbors of **a**, relative to the number of possible connections. For example, **e**, **f**, and **b** are neighbors of **a**, however the connections (**e**,**f**) and (**b**,**f**) are not in the graph.

4.1.2 Node metrics and measures

Both elements of the graph, the nodes and the edges, can be described and summarized using different metrics and measures. For example, the distance between two nodes in a graph can be calculated as the length of the shortest path between them, i.e. the number of edges that are needed in order to get from one to the other (Figure 4.3(a)). These measures can also be used to characterize the entire structure of a graph. Different node metrics have been defined to quantify and emphasize different aspects of the topology of the graph. Some of these metrics are used to indicate how central a node is, i.e. how important it is (depending on what the graph represents). Here we list several of the commonly used metrics, albeit there are many more. We provide only the definitions in the case of undirected graphs.

- **Node degree**: the degree of a node is the number of its adjacent nodes, which is also referred to as the number of neighbors. Figure 4.3(b) shows for example that the degree of node a is 5, since it is connected to five other nodes in the graph.

- **Betweenness centrality (BC)**: the BC is a measure that indicates how many shortest paths in the graph pass through the node v [92]. It is given by the following definition:

$$\mathrm{BC}(v) = \sum_{x,y \in V} \frac{\mathrm{SP}(x,y|v)}{\mathrm{SP}(x,y)} \tag{4.1}$$

  where $\mathrm{SP}(x,y)$ is the number of shortest paths between x and y, and $\mathrm{SP}(x,y|v)$ is the number of shortest paths between x and y that go through v. In Figure 4.3(c) the BC of node a is 0.6 as 60% of the shortest paths in the graph pass through it.

- **Clustering coefficient (CC)**: the CC measures for a node v the proportion of connections between its neighbors (relative to the number of connections that can exist) [313]. It is given by the following definition:

$$CC(v) = \frac{2|T(v)|}{degree(v)(degree(v)-1)} \tag{4.2}$$

  where $T(v) = \{(x,y)|(v,x) \in E \wedge (v,y) \in E\}$, i.e. all the edges that connect all the nodes that interact with v. If v has d neighbors, then its degree is d and therefore the number of edges that could exist between all its neighbors is $\frac{d(d-1)}{2}$. In Figure 4.3(d) the CC of node a is 0.4 since only 40% of the connection between its neighbors exist in the graph.

- **Node core**: the core, sometimes also referred to as k-shell, of a node v is defined by the largest value k of a k-core graph that contains v [41]. A k-core graph is a maximal sub-graph of a graph G that contains only nodes with a degree of at least k. To find the k-core number of a node v a k-shell decomposition algorithm [31] can be used. The algorithm works iteratively, by removing at each step the nodes with degree smaller than k and creating the next k-core sub-graph. An example is provided in Figure 4.4. In the first step, all nodes with degree 1 are removed from the graph. Subsequently, more nodes with a degree of 1 appear (in the newly created sub-graph) and those are removed too. Eventually, all these removed nodes are assigned a core value of $k = 1$. In the next step, in a similar way, all nodes with degree of 2 are removed, until only nodes with degree of 3 or more are left. The process is continued until no further nodes can be removed, and the highest k-core is reached.

- **Node H-index**: Similar to the H-index for citations [121], the H-index of a node is defined by the maximum value h, such that the node has at least h neighbors with degree of h or higher [155]. The H operator [172] can be defined to produce the H-index of node i given the degrees of its neighbors:

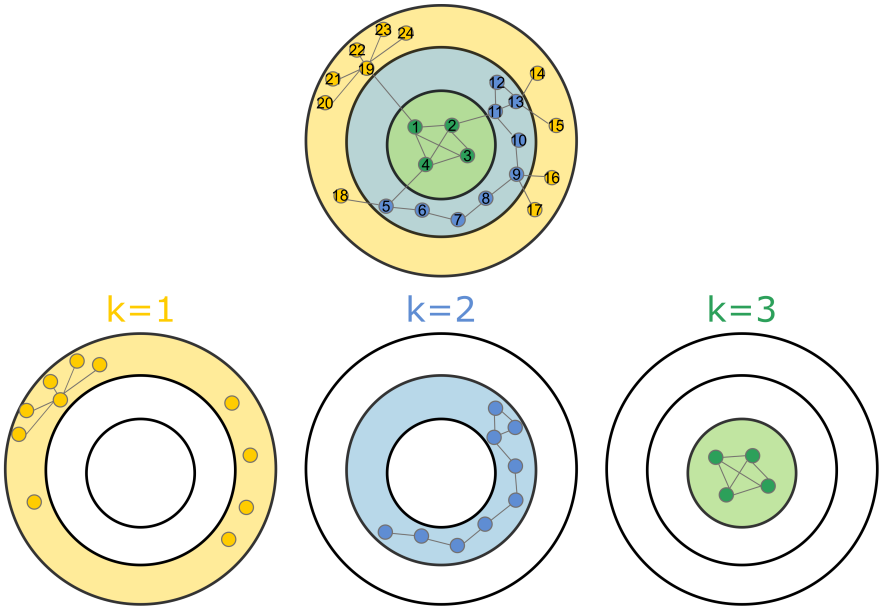
$$h_i = H(d_{1i}, ..., d_{ji}) \tag{4.3}$$

  where node i has j neighbors, and d_{ji} is the degree of the j-th neighbor.

  Lü et al. [172] first described the relation between the degree, core and H-index. They defined a series of H-indices using the H operator, such that for a node v the initial value is its degree, the second value is its first order H-index and the series converges to the final value which is equivalent to its k-core.

4.1.3 Random graph generation

Certain global and local properties of graphs are more common than others, and therefore several definitions of graph models have emerged over the years. In **small-world** graphs most nodes are not directly connected to the rest of the nodes in the network, however, the probability of the neighbors of a node being connected to each



Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Degree	4	4	3	4	3	2	2	2	4	2	3	2	4	1	1	1	1	1	6	1	1	1	1	1
h(1)-index (H-index)	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
h(2)-index (core)	3	3	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1

Figure 4.4: k-shell decomposition and H-index: An example network with 25 nodes and 29 edges. In this example the network is decomposed into three layers, each one representing a different k-core sub-graph. The first layer (k-core = 1) and the nodes in it are colored in yellow, the second in blue (k-core = 2) and the third in green (k-core = 3). The table details the different H-indices, starting from degree and ending in core. In this example, the convergence to core is already achieved at h(2)-index.

other is high. Therefore, these graphs are characterized by a short average distance between nodes, despite having a relatively high average CC. Many real networks follow the properties of small-world graphs, including social networks, biological networks, and the internet. **Scale-free** [7] graphs are characterized by a power-law degree distribution. These graphs typically include **hubs** which are high degree nodes, and their degree greatly exceeded the average degree in the graph. The distances in these graphs are also relatively short, however, the CC also follows a power-law distribution, as it decreases as the node degree increases. Many complex networks that hold small-world graph characteristics also follow scale-free ones, for example, PPI networks.

A graph that follows a certain graph model can be generated via a random process or using some probability distribution. The main idea is, given a set of V nodes, to randomly generate a number of edges E such that a certain property of the graph is achieved. The most common random graph model is the Erdös–Rényi one [85]. According to this model, each edge, out of all possible edges, has a uniform probability to be present in the graph. These types of graphs are characterized by a Poisson de-

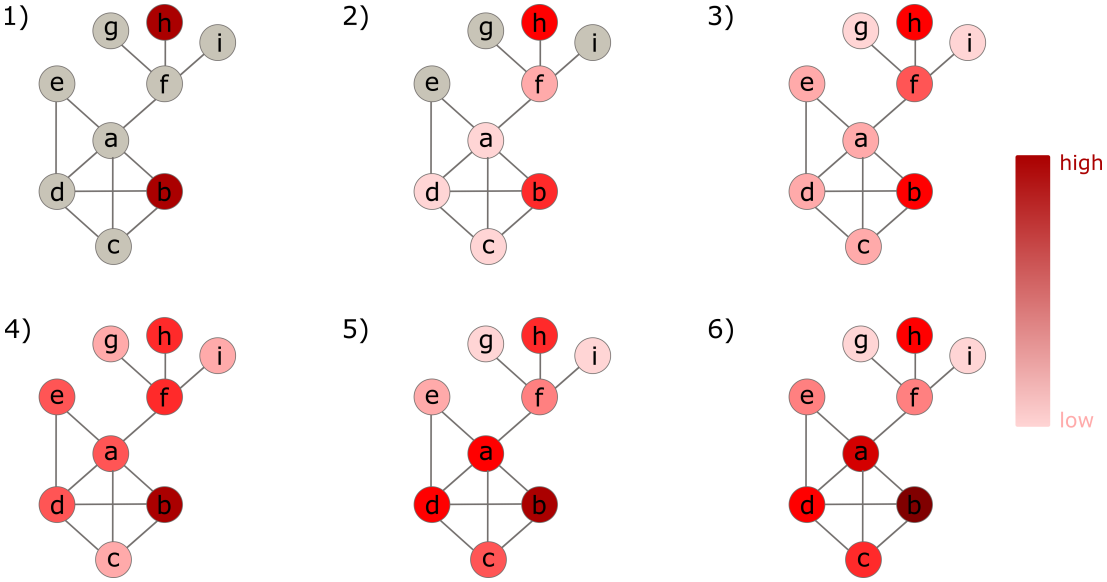


Figure 4.5: Network propagation: A step-by-step demonstration of the propagation process on a network of 9 nodes and 12 edges. The propagation process is depicted from time step 1, where only nodes **b** and **h** are scored with a weight higher than 0, and until time step 6, which describes here the steady-state. The nodes are colored according to the amount of weight that is propagated to them. The weights are propagated in a step-wise manner, such that each node gives and receives weight to and from its neighbors.

gree distribution. However, many complex networks are scale-free and therefore follow a power-law degree distribution. Hence, other models for generating random graphs have been suggested, in order to reproduce different characteristics of complex networks. For instance, the Watts–Strogatz model [313] does not follow a power-law degree distribution, but does preserve a short average path length and high clustering coefficients. The Barabási–Albert model [25] has the same characteristics, and also follows a power-law degree distribution. The hierarchical model [209] is similar, but follows a different CC distribution. There are many more algorithms that generate different types of graphs, and they are reviewed for example in the book by Bollobás [42].

4.2 Network propagation

Network propagation is the process of propagating, or diffusing, information throughout a graph in an iterative fashion through a series of steps, or until a steady state is reached (Figure 4.5). As described in Section 3.5, it is used in a wide variety of fields, including molecular biology. There are several mathematical formulations that can be applied to execute the propagation process, based on random walks or diffusion models. Here, we will provide the definitions for the case of an **undirected** and **unweighted** graph only. To incorporate edge weights only minor changes are necessary. However, the mathematical formulations in the **directed** case are more complex and further summarized by Malliaros and Vazirgiannis [178].

The input for all formulations always includes two main components: a graph G and an initial weight vector $p_0 = \{s_1, \ldots, s_n\}$ with weights for all the nodes $V = \{v_1, \ldots, v_n\}$ in the graph. At every step t the current amount of weight for node v, which is given by $p_t(v)$, is calculated based on the weights of its neighbors in the previous step, where p_{t-1} holds the weights for all nodes at step $t-1$. Thus, the weight on node v at step t is calculated according to:

$$p_t(v) = \sum_{u \in N(v)} p_{t-1}(u) w(u,v) \tag{4.4}$$

such that $N(v)$ are the neighboring nodes of v and $w(u,v)$ is the probability of propagating from u to v. If G is a weighted graph, then $w(u,v)$ might reflect the weight of the edge (u,v).

Network propagation can be viewed as a random walk on the graph [170], which is a sequence of nodes $v_0, \ldots, v_t$ where v_{t+1} is randomly chosen among the neighbors of v_t. In the walk, the probability of being at node v_{t+1} at time step $t+1$ depends only on the probability of being at node v_t at time step t and the walking probability from v_t to v_{t+1}. The sequence is thus equivalent to a Markov chain, with a random variable X and the process $X_0, X_1, \ldots, X_t$. The state of the Markov chain at time step t is equivalent to the node v_t that was visited at that time step. The Markov chain is thus defined by the Markov property:

$$\mathbb{P}(X_{t+1} = v_{t+1} | X_t = v_t, X_{t-1} = v_{t-1}, \ldots, X_0 = v_0) = \mathbb{P}(X_{t+1} = v_{t+1} | X_t = v_t) \tag{4.5}$$

for all $t = 1, 2, \ldots$ and all possible states (nodes) $v_1, \ldots, v_n$.

The walking probabilities can be described in a matrix form, where W is derived in some way from the adjacency matrix A of the graph (see Sections 4.2.1-4.2.3). And so Equation 4.4 can be defined as:

$$p_t = W p_{t-1} \tag{4.6}$$

In the same way, W defines the transition probability matrix in the Markov chain, which describes the probability of going from one state to another: $W_{i,j} = \mathbb{P}(X_{t+1} = v_i | X_t = v_j)$. In a Markov chain W is by definition stochastic, i.e. the columns sum up to 1 (here the probabilities of going from v_j to v_i are given by the columns of W).

In the random walk, at every time step, the weights are calculated using W and the weights from the previous time step, such that $p_0 \rightarrow p_1 = W p_0 \rightarrow p_2 = W p_1 = W^2 p_0 \rightarrow p_3 = W p_2 = W^3 p_0$ and so on. This means that after t steps p_t can be described using the initial weight vector p_0 and the matrix W:

$$p_t = W^t p_0 \tag{4.7}$$

The Perron-Frobenius theorem guarantees that the Markov Chain converges if it follows three conditions: 1) time-homogeneous, 2) irreducible and 3) aperiodic. Under these conditions, there exists a unique equilibrium distribution (also called stationary distribution) π such that $\pi = W\pi$. π is the largest non-negative eigenvector and since

W is stochastic the eigenvalue with π is 1. Once converged, the equilibrium distribution π does not change anymore.

Correspondingly, the random walk on the graph converges to a steady-state only when the graph is fully connected and W is stochastic and does not change over time. Then, calculating $W^t p_0$ is possible by power iteration, which results in a multiple of the eigenvector (to the largest eigenvalue). As such, the final scores $p = \{r_1, \ldots, r_n\}$ can be directly calculated from p_t by taking the limit (Equation 4.8). Under these conditions the Perron-Frobenius theorem guarantees that the limit exists and that it is unique. Thus, p is equivalent to the equilibrium distribution π.

$$p = \lim_{t \to \infty} p_t \tag{4.8}$$

While in Markov chains the transition probabilities are defined depending on the states, in network propagation they are defined using the adjacency matrix A of the graph. The different implementations of network propagation also differ in the definition of W. Here we will explore three main variants: random walk, random walk with restart and diffusion kernel.

4.2.1 Random walk

In a random walk process on an interaction network, at each time step t, the walk randomly moves towards one of the neighbors of the current node v_t. Thus, W can be set using the adjacency matrix A such that $W = AD^{-1}$, where D is a diagonal matrix with the degree of the nodes. Then the random walk can be described by:

$$p_t = (AD^{-1})^t p_0 \tag{4.9}$$

The higher t is, i.e. the more steps of the walk were executed, the more of the initial information in p_0 was diffused through the graph, and p_t can be used to represent the graph's topology. Since the definition for W is stochastic, i.e. the columns sum up to 1, and as the graph is connected, the random walk converges and the steady-state distribution is reached. At the steady-state the final weights p depend only on the topology of the graph, regardless of the initial weights p_0. p thus defines the probability of being at each one of the nodes, which does not change anymore.

In order for the final weights to indicate how connected are the nodes in the network to the nodes which have a non-zero initial weight in p_0 one can modify the random walk and define the random walk with restart (RWR).

4.2.2 Random walk with restart

An alternative version to the random walk is the RWR, which is also referred to as insulated diffusion or personalized PageRank [217]. Here at each time step the walk can either continue to one of the neighbors, or restart from the initial position with a restart probability equal to α. α controls the trade off between the initial weights

p_0 and the topology of the network which is represented by W. The RWR process is described by:

$$p_t = \alpha p_0 + (1-\alpha) W p_{t-1} \tag{4.10}$$

Plugging Equation 4.7 into Equation 4.10 results in:

$$p_t = (\alpha I + (1-\alpha) W^t) p_0 \tag{4.11}$$

where $M = \alpha I + (1-\alpha) W^t$ is the transition matrix, which remains stochastic (as I is the identity matrix). Hence, this process also converges at the steady-state with the direct solution:

$$p = \alpha (I - (1-\alpha) W)^{-1} p_0 \tag{4.12}$$

where $F = \alpha (I - (1-\alpha) W)^{-1}$ is sometimes also called the diffusion matrix.

W is usually set to AD^{-1}, as for the random walk, or also sometimes to $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. Both of which are stochastic and thus satisfy the necessary convergence condition.

The computation of the diffusion matrix F requires to invert the matrix $I - (1-\alpha) W$ which can be computationally intensive (cubic running time with respect to the number of nodes) for very large graphs. However, it is important to note that F can be computed once, allowing to execute RWR several times on the same network, using different initial weights. On the other hand, this computation requires to store F, which can take a large amount of space (quadratic space with respect to the number of nodes). It is possible to reduce the computation time and space, for example by exploiting some graph properties which are embedded in the adjacency matrix [289]. Such solutions are faster and reduce the required space, yet the approximation might slightly deviate from the correct result.

4.2.3 Diffusion kernel

Diffusion kernels are exponential graph kernels which are based on the heat equation [153] and are also called heat kernels. These kernels are the continuous-time analogues of the RWR. F defines a kernel if it is symmetric and positive semi-definite, e.g. for the diffusion kernel F can be defined by $F = e^{-\alpha L}$ where L is the network's Laplacian matrix with $L = D - A$.

Part II

METHOD, EVALUATION AND APPLICATION

5 NetCore: Network Propagation with Core Normalization

NetCore is a network propagation approach which was designed to attend to the two main issues that were presented in Section 3.6. The innovation of the proposed method are two-fold: 1) it addresses the study bias problem in PPI networks by incorporating an alternative centrality measure in the RWR formulation, and 2) it combines prior knowledge with experimental evidence in a network propagation framework to identify network modules in a semi-supervised fashion. The following chapter includes a description of the two main parts of the NetCore approach and its implementations. We first describe the different steps of the NetCore approach and then provide details regarding all of its components.

5.1 Overview of the NetCore approach

NetCore's workflow, presented in Figure 5.1, consists of three main steps: (1) data initialization, (2) node re-ranking and (3) module identification:

1. **Data initialization**: for the network propagation procedure both a PPI network and initial node weights are necessary. For NetCore, we extracted a high quality PPI network from CPDB, which is described in Section 5.2. However, it is also possible to use other interaction networks from different resources. The input weights must be provided such that for each gene (node) one input weight is computed in advance, which best reflects its experimental evidence. Yet, not all the nodes in the network must be weighted in order to execute the propagation, and therefore unweighted nodes will be assigned with 0. As the weights are computed prior to the application of NetCore, they depend entirely on the phenotype in question and could vary according to the type of data. In Chapter 7 we provide a few examples on how to derive weights from different genomics data for applying NetCore. Alternatively, NetCore can also be applied for a given list of genes of interest, using a binary scoring scheme, i.e. the propagation will be executed only from these genes.

2. **Node re-ranking**: In this step the weights are propagated in the network via a modified RWR formulation that is implementing node core, which is detailed in Section 5.3. The weights after the propagation are assigned with a significance level based on a network randomization procedure, which is described in detail in Sections 5.4-5.5.

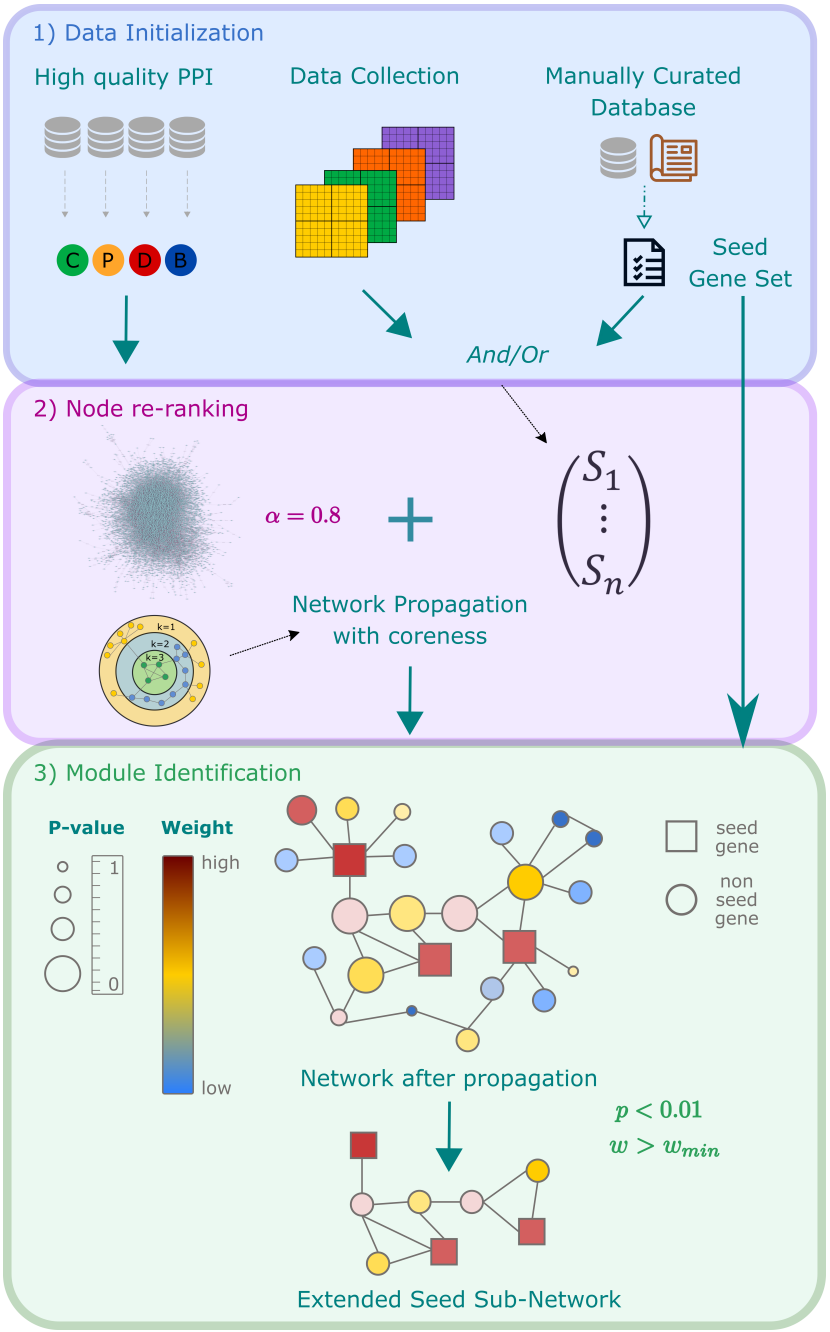


Figure 5.1: NetCore workflow: The approach is depicted in a three step workflow: (1) **Data initialization**: includes the extraction of a scaffold PPI network, experimental data and extraction of a seed gene list representing prior knowledge. (2) **Node re-ranking**: network propagation using node core which involves initialization of node weights with experimental data or alternatively with a weighted list of seed genes, random walk with restart propagation specifying the restart parameter α (default $\alpha = 0.8$), and assigning a propagated final weight and a P-value (through permutation analysis) to each node. (3) **Module identification**: in a semi-supervised fashion combining both network propagation results and the seed gene list. The seed genes are connected by PPIs in a sub-network and neighbor nodes that have a significant P-value and a sufficient weight after re-ranking are added to create an extended seed sub-network.

3. **Module identification**: the final step allows for the identification of network modules based on the ranking after the propagation in a semi-supervised procedure, which is described in Section 5.6. Given a pre-defined list of seed genes, and in combination with genes which achieved a significant weight after the propagation, sub-networks (modules) are extracted. These sub-networks expand on prior knowledge and interconnect well-known seed genes with novel predictions.

5.2 High confidence PPI network from ConsensusPathDB

ConsensusPathDB (CPDB) [145] is a meta-database for molecular interactions and pathways, available for human, yeast and mouse. The current human release (version 34 on 15.01.2019) integrates 32 public resources. It is composed of more than 600,000 unique interactions including: binary and complex protein-protein, genetic, metabolic, signaling, gene regulatory and drug-target. In addition, it holds more than 5,000 pathway concepts. By now, CPDB includes more than 300,000 binary protein-protein interaction from 19 different resources, which can be represented in a PPI network. This network serves as a comprehensive model of the human protein interactome. Each interaction is associated with a confidence score that was calculated using the `IntScore` [143] method. This confidence score is a meta-score, i.e. it is based on a mixture of multiple topology-based and annotation-based measurements. The final score aims to indicate how plausible an interaction is. This is highly relevant in the case of PPIs due to technical issues with the experimental methods that produce such interactions, which might result in false observations [227, 248].

For NetCore's workflow we needed to chose an appropriate interaction network. CPDB, which was previously developed in-house [118, 142, 144, 145], integrates PPIs from multiple resources and provides confidence scores for them. Thus, we found it to be useful for the purpose of applying network propagation in NetCore. In addition, the PPI network from CPDB was also recently reported in an independent study as one of the top performing networks for identifying disease genes via network propagation [133]. However, this network includes a large number of interactions, some of which might not be accurate due to experimental deficiencies [227, 248]. Therefore, there is a need to construct a smaller, more accurate PPI network, which is sufficiently comprehensive and contains mostly true-positive interactions. To this end, we previously constructed a high confidence PPI network from CPDB (version 32 on 11.01.2017) [29]. We used the confidence scores from `IntScore` [143] and selected only interactions with a score equal and above 0.95. These scores are only available for binary interactions, and not complex ones, and therefore the network consists only of binary PPIs.

The distribution of the scores, shown in Figure 5.2(a), is rather bimodal, that is most of the interactions have either a very low score, equal or below 0.1, or a very high score, mostly above 0.8. Therefore, choosing a high cut-off allows us to keep more than a third of the interactions, while still maintaining only high quality ones. This network is much smaller, with $10,707$ proteins and $114,516$ unique interactions (compared with $16,526$ proteins and $264,493$ interactions in total), yet it preserves the power law distribution

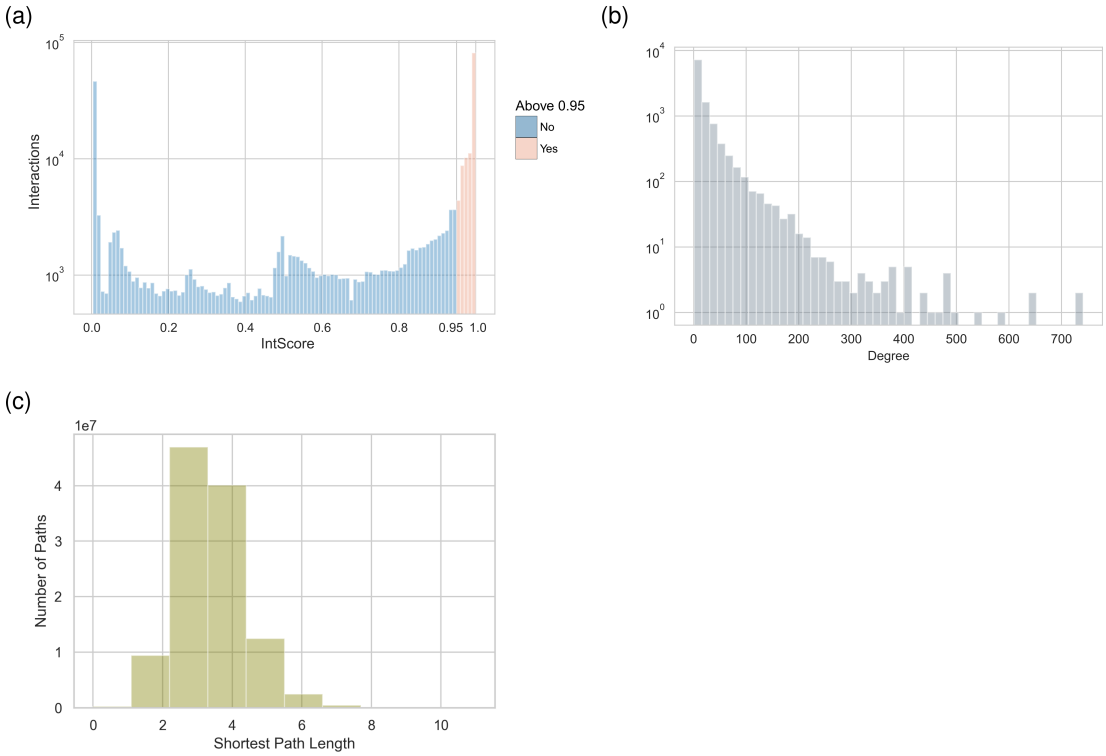


Figure 5.2: Metrics and measures in the CPDB PPI network: a) Distribution of `IntScore` values for the entire CPDB network. b) Degree distribution for the high confidence CPDB PPI network. c) Shortest path lengths in the high confidence CPDB PPI network.

of the node degree (Figure 5.2(b)), which is one of the main characteristic of biological networks [27, 140, 325]. In addition, it is also characterized by short distances (Figure 5.2(c)), since the maximal length of a shortest path in the network is 11, and the median is only 3. Therefore, this version of the network is used in NetCore and throughout the evaluations and applications that are described in Chapter 6. A comparison between different versions of the CPDB PPI network is provided in Section 6.3.1. From here on, unless specified otherwise, when we refer to the CPDB PPI network, we refer to the high quality version.

In Section 3.6.1 we elaborated on the study bias that is inherent within PPI networks. We described how some nodes are more connected than others, creating a bias towards high degree. So far, the proposed solutions for addressing the study bias in network propagation were focused on statistically adjusting the results (see for example [86] and [38]). Here we suggest a modification which can be applied directly to the mathematical formulation of network propagation in the form of RWR. The idea is to normalize the adjacency matrix using other node metrics than degree in order to reduce the study bias. To that end, we explored three alternative node metrics (see definitions in Section 4.1.2): clustering coefficient, betweenness centrality and core. In the following sections we describe the characteristics of these metrics in the CPDB PPI network and their relation to node degree. Then, we demonstrate how node core can be used for normalizing the adjacency matrix, and provide various versions of core-based normalizations which are suitable for the RWR form of network propagation.

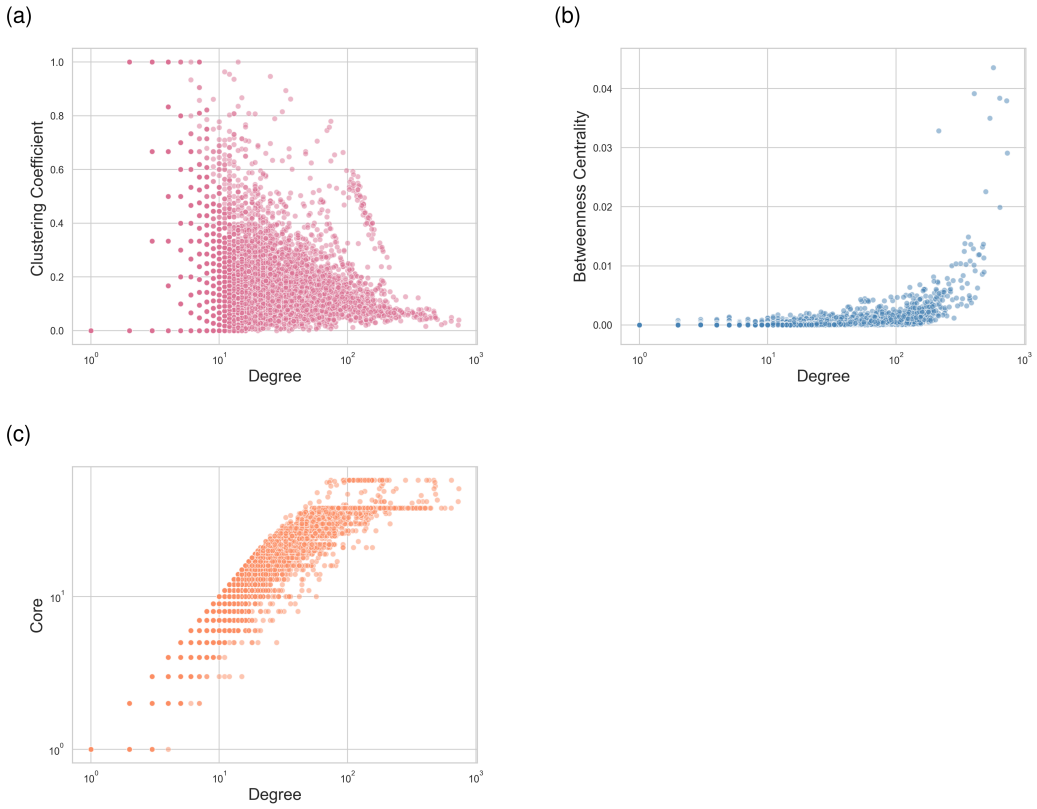


Figure 5.3: Node metrics in the CPDB PPI network in relation to degree: a) clustering coefficient, b) betweeness centrality, c) core.

5.2.1 Clustering coefficient in CPDB PPI

The clustering coefficient (CC) is used as a measure for how well connected are the neighbors of a given node. The values range from 0 to 1, where larger values indicate better connectivity of the node's neighbors. In the PPI network from CPDB most of the nodes in the network have a CC equal or below 0.2, with an average of 0.16 for all nodes. Figure 5.3(a) illustrates for every node in the network its degree and CC. Overall, there is hardly any correlation between CC and degree (Pearson correlation coefficient of -0.01). Yet, only nodes with a degree lower than 100 have a CC which is higher than 0.6. Nodes with the highest degrees always display a very low CC, i.e their neighbors are less connected among themselves. Thus, this measure also reflects the study bias, as low degree nodes will have a high CC, and high degree nodes a lower one. The fact that the neighbors of high degree nodes are not well connected implies that those were probably only used as 'prey' proteins, while the high degree nodes as 'bait' proteins.

5.2.2 Betweenness centrality in CPDB PPI

Betweenness centrality (BC) was devised to measure how central a node is in a network. It calculates how many shortest paths in the network pass through the node. The values range from 0 to 1, where higher values imply higher centrality. In the CPDB PPI network the values are at most 0.04 for all nodes. Figure 5.3(b) illustrates for every node in the network its degree and BC. There is a positive correlation to node degree (Pearson correlation coefficient of 0.73), where only the nodes with the highest degrees have a BC value above 0.02. The majority of nodes in the network have a BC value below 0.01. Hence, the study bias is also reflected by the BC, since the most central nodes by this measure are also the nodes with the highest degree.

5.2.3 Core in CPDB PPI

In Section 4.1.2 we introduced the core metric for nodes. In short, node core determines to which k-core layer of the network a node belongs to. Due to the nature of the k-shell decomposition algorithm (Figure 4.4), the core is always equal or lower than the degree, and never higher. Figure 5.3(c) illustrates the relation between node degree and core in the PPI network from CPDB. The values range from 1 to 58, and there is a positive correlation between node degree and core (Pearson correlation coefficient of 0.76). Table 5.1 lists 30 nodes with the highest degrees in the network and their k-cores. While the highest degrees range from 350 to 737, the core levels for these high degree nodes range between 38 and 58. Many of these nodes are associated with diseases such as cancer and other complex disorders.

Indeed, the study bias is reflected here by the high degree, however, it is not directly implied by a high core. For instance, the node with the highest degree in the network, with 737 connections, is *TP53*, a well-known tumor suppressor which is characterized as one of the main cancer drivers [103, 228]. In contrast, the highest core in the network is 58 and there are 71 genes in the network at this core level, with degrees ranging from 74 to 645. *TP53*, however, has a lower core of 51, i.e. despite having the highest degree in the network, it is not in the highest core level of the network. Figure 5.4 shows the distribution of degrees for the nodes in each core level in the network. Evidently, nodes with the same core level vary greatly in their degree. Moreover, there is a positive correlation in the variation: the higher the core, the higher the variation in the degree. For example, in core level 47 there are only two nodes, *MAP3K14* with 100 interactions, and *RNF2* with 485. This tends to be the case for all core levels between 39 and 57, which include (in each level) only a small number of nodes, with a rather high variance in degree. The only exception is the highest core level, which consisted mostly of ribosomal proteins, with degrees ranging between 74 and 211.

Even at lower core levels, the difference in the degree of the nodes is evident, and could suggest to study bias. For example Figure 5.5 displays the neighborhoods of two genes: (a) *LPAR1* (Lysophosphatidic Acid Receptor 1) and (b) *RSRC1* (Arginine And Serine Rich Coiled-Coil 1). Both genes are associated with height, according to the GWAS catalog [176], and have the same number of 21 connections in the network. However, *LPAR1* has a core of 8 whereas *RSRC1* has a core of 18. This difference in

Node	Node degree (max = 737)	Node core (max = 58)	Network of Cancer Genes (NCG)	GWAS Catalog gene sets
TP53	737	51	Yes	
XPO1	726	42	Yes	
CUL3	645	58	Yes	Schizophrenia
UBC	642	38		
EGFR	574	38	Yes	
NTRK1	538	38	Yes	
GRB2	500	43	Yes	
RNF2	485	47		
CDK2	483	53		Rheumatoid arthritis, Vitiligo
MCM2	479	56		
ESR1	472	51	Yes	Breast Cancer, Height
CUL1	470	58	Yes	
HDAC1	445	38		
EP300	438	38	Yes	Crohn's disease, Schizophrenia
COPS5	431	58		
NPM1	413	58	Yes	
SIRT7	412	42		
APP	407	38		
MYC	405	38	Yes	Breast Cancer, Prostate Cancer, Height
YWHAZ	400	38		
EED	386	42	Yes	
CSNK2A1	382	38	Yes	
BRCA1	378	38	Yes	
CDC5L	378	38		
CUL7	374	46	Yes	
SNW1	372	38		
TRAF6	366	38	Yes	
HNRNPA1	358	42	Yes	
HNRNPU	354	58	Yes	
HSP90AB1	350	38		

Table 5.1: Hub nodes in the CPDB PPI network: For each node its degree and core values in the network are noted. In addition it is noted whether the node has been associated with cancer, according to the NCG [240] and to which of the 11 GWAS gene sets (Table 6.1) from the GWAS catalog [176] it belongs.

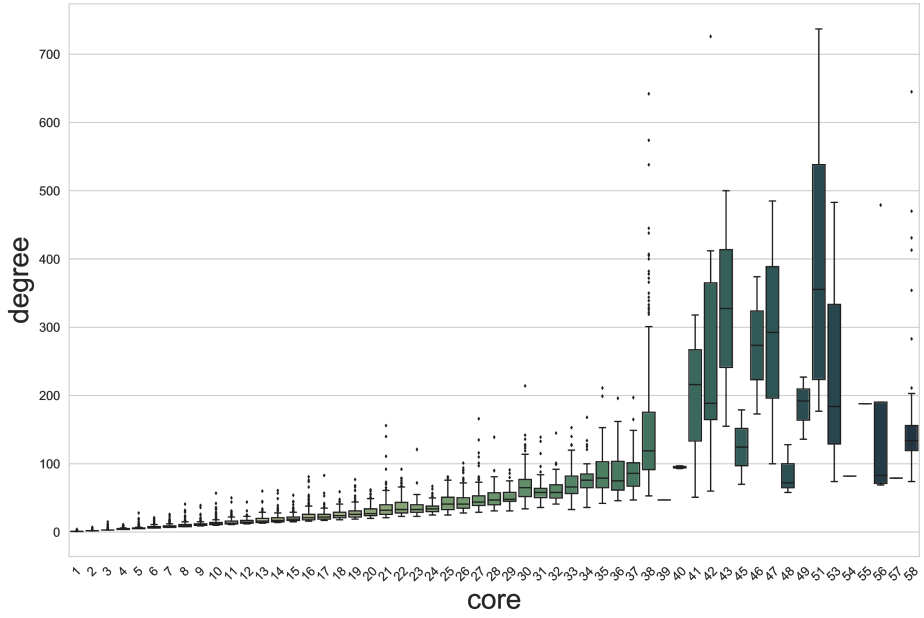


Figure 5.4: The relation between degree and core in the CPDB PPI network: For each core value the distribution of the degree values of the genes in this core level is illustrated.

core is reflected by the difference in the degrees of their neighbors. Out of *LPAR1*'s 21 neighbors, 16 have a degree that is lower or equal to 21, and the rest have a degree between 27 and 83. In contrast, *RSRC1* is connected to nodes with much higher degrees, all but one have degrees equal to at least 21 and up to 382. Thus, the core values reflect that *RSRC1* is located in a much denser region of the network than *LPAR1*. Although both genes have the same number of connections in the network, it is possible that *LPAR1*'s high degree is due to study bias, as it is connected to nodes which themselves are less connected to others. Indeed, the number of publications associated with *LPAR1* is 193, while *RSRC1* is associated with only 35 publications (based on publications in the GeneCards database [276]).

Node degree and core can be directly related to one another. Recently, Lü et al. [172] demonstrated how core can be estimated through a series of steps using the H-index (Section 4.1.2). They constructed an operator H for calculating the H-index of a node based on the degree of its neighbors. By applying this operator in a sequential and synchronous manner, starting from the node degree, the process soon converges to the node core. This means that degree can serve as a local property of the node, the H-index series as intermediate centrality measures, and the core as a global property. Furthermore, Kitsak et al. [151] have shown that core is a better indicator than both degree and BC for how influential a node is in a network. They examined the spread of information in a network using the susceptible-infected-removed (SIR) model, which is usually used for epidemiological models of infectious diseases [15], and found that nodes in the core of the network are the most efficient spreaders of information.

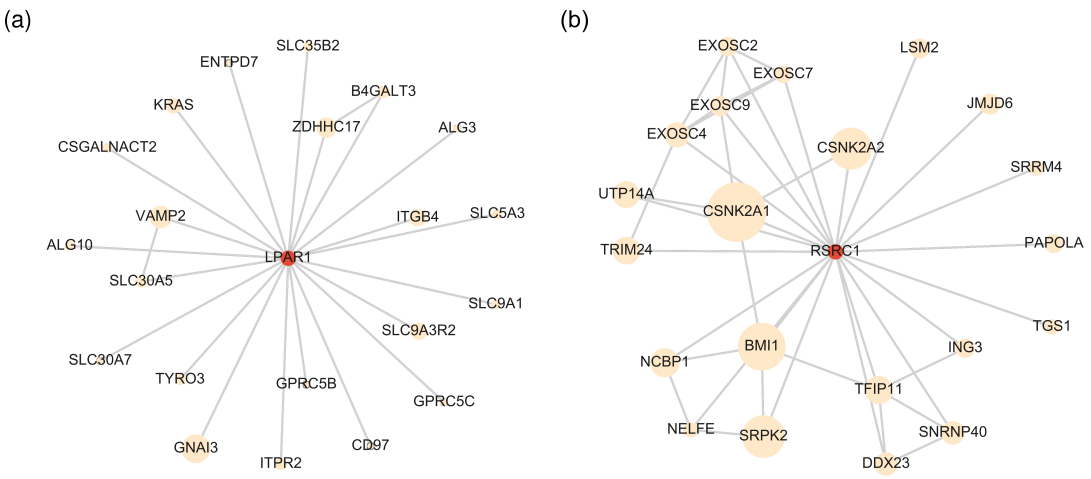


Figure 5.5: The neighborhood of (a) LPAR1 (degree = 21, core = 8) and of (b) RSRC1 (degree = 21, core = 18) in the CPDB PPI network. Node sizes of the neighbors are visualized in proportion to their degree.

We demonstrated here how the study bias effects the degree, while core is more robust. This study bias is particularly relevant for 'bait' proteins, which creates star-like structures in the network with connections to multiple 'pray' proteins (see Section 3.6.1). In these cases, despite the degree being high, the core is low, correcting for the high bias around such nodes.

5.3 Adjacency matrix normalization using node core

In most of the network propagation formulations the probabilities for walking from one node to another are calculated based on the adjacency matrix, which contains the structure of the network, and the node degree (see (Section 4.2). Thus, normalizing the adjacency matrix differently provides means to adjust the walking probabilities and results in different ranking of the nodes after the propagation. To address the study bias that is reflected in the degree we decided to use core, which we concluded as more robust to the bias. Since degree and core are interrelated, we explored multiple modifications for the normalization of the adjacency matrix, as an alternative to the standard degree normalization: using core alone or combinations of both core and degree together. As the study bias is reflected in nodes with a high degree but a low core, a large difference between the degree and the core of a node will suggest to the extent of the bias concerning it. In order to adjust the walking probabilities accordingly, we can penalize the probability of walking to such node with respect to this difference. In a similar way we can also utilize the ratio instead of the difference. Thus, we define three core-based normalizations which are detailed next.

Given a graph G, with a set of n nodes $V = (v_1, ..., v_n)$ and for each node its degree $D = (d_1, ..., d_n)$ and its core $K = (k_1, ..., k_n)$. In Section 4.2.2 we described the RWR

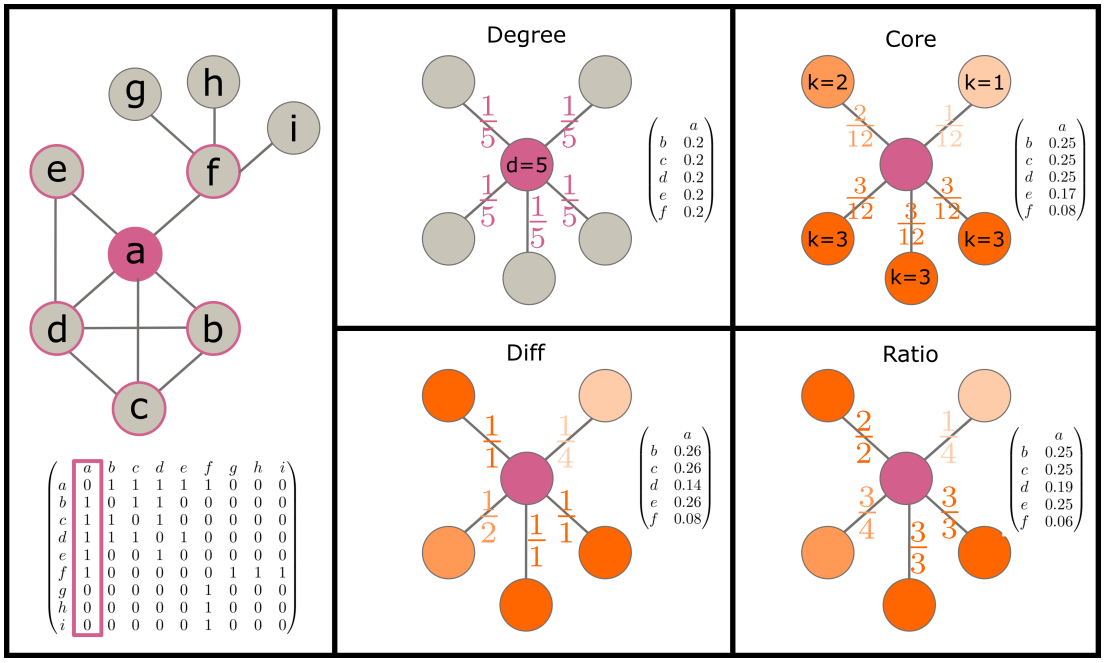


Figure 5.6: Adjacency matrix normalization: An example network with 9 nodes and 12 edges with the corresponding adjacency matrix. Node a has five neighbors. The normalization schemes for the neighbors of a are exemplified. When normalizing based on degree only, the probability for walking to any of the neighbors is 0.2, since the degree of a is 5. When normalizing based on core only, the probability is according to the core of the neighbor, and normalized by the sum of cores of all neighbors, such that the sum of probabilities is 1. The other two normalizations are based on the difference between the degree and core, or the ratio between them. The vectors show the probability values after normalizing by the sum of differences (or ratios), such that all the values sum up to 1.

formulation of network propagation. In this formulation, the adjacency matrix is most commonly normalized using the degree of the nodes, so we define $A^{degree} = AD^{-1}$, where A is the adjacency matrix and D is the diagonal degree matrix. Here, we apply three different normalizations to the adjacency matrix A using core and degree. Figure 5.6 illustrates these normalizations procedures on an example network. To keep the matrix stochastic, and thereby ensure convergence (see Section 4.2), the adjacency matrix is always normalized such that the sum of every column is 1.

- **core**: each column A_j is normalized using the core of the neighbors of node v_j. Thus, for each neighbor v_i of v_j we divide its core k_i by the sum of cores of all of v_j neighbors. This results in a stochastic matrix, as the sum of every column is always 1.

$$A^{core}_{i,j} = \frac{k_i}{\sum_{l \in N(j)} k_l} \tag{5.1}$$

  where $N(j)$ are the indices of the neighbors of node v_j.

- **degree-core difference**: each column A_j is normalized using the difference between the degree and the core of the neighbors of node v_j. The difference is

always non-negative. To keep the sum of the column 1, we then divide the normalized values by the sum of all values in the column.

$$A_{i,j}^{diff} = \frac{1}{(d_i - k_i) + 1} \tag{5.2}$$

$$A_{i,j}^{diff-norm} = \frac{A_{i,j}^{diff}}{\sum_{i \in (1,\ldots,n)} A_{i,j}^{diff}} \tag{5.3}$$

- **degree-core ratio**: each column A_j is normalized using the ratio between the core and the degree of the neighbors of node v_j. The ratio is always equal or lower than 1. To keep the sum of the column 1, we then divide the normalized values by the sum of all values in the column.

$$A_{i,j}^{ratio} = \frac{k_i}{d_i} \tag{5.4}$$

$$A_{i,j}^{ratio-norm} = \frac{A_{i,j}^{ratio}}{\sum_{i \in (1,\ldots,n)} A_{i,j}^{ratio}} \tag{5.5}$$

5.4 Statistical significance via permutation tests

The network propagation procedure produces new rankings for the genes, however those do not carry any statistical significance information. The raw scores can vary in size, depending both on the input scores and the topology of the network, and therefore it is crucial to be able to associate a result as a favorable one. The significance information allows for making educated conclusions and selecting suitable candidates for further studies.

Currently, there are several proposed models for statistical adjustment of network propagation results. Most models are based on randomized sets of the input seed nodes [37, 86, 187]. They differ in the way random seeds are generated, when some also make sure to maintain a similar degree distribution to that of the input seed nodes. On top of that, the `DADA` [86] suit also implements a likelihood-ratio test using eigenvector centrality. In this case, the score of a seed node is compared to its eigenvector centrality [47], which is equivalent to the score when executing the propagation with a restart probability of 0. Recently, Biran et al. [38] evaluated the performance of the aforementioned models, and developed a new model which is based on randomizations of the interaction network, rather than the input seed nodes. This model outperformed the others when applied to multiple sets of disease-related and function-related input seed nodes.

For NetCore, we chose to implement the method suggested by Biran et al. [38] which is based on random degree-preserving networks (RDPN). Figure 5.7 illustrates this procedure. In this model the propagation score of a node is compared to its propagation score when using a random network instead of the input network. The random networks are generated such that the degree of all nodes in the random network stays

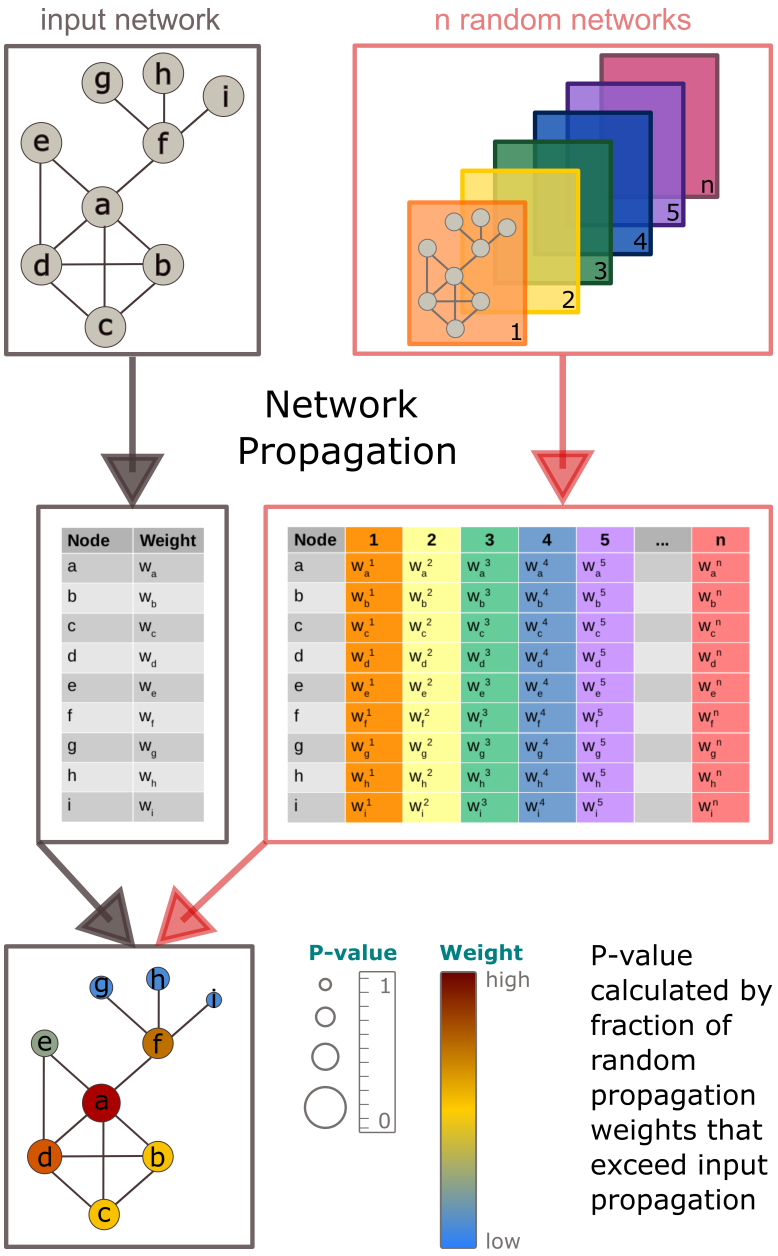


Figure 5.7: Statistical significance via permutation tests: Given an input network, n permutations of it are generated via some random graph generation process. Then, network propagation is applied, both on the input network, as well as on each one of the random networks, using some initial weights. The weights at the end of the propagation when the input network was used are compared with the weights when all the random networks were used. The P-value is calculated as the fraction of random propagation weights that exceeded the propagation weight of the input network. In the end, each node in the network is associated with a propagation weight and a P-value.

as in the input network, and only the interaction partners are changed. Once n such random networks are generated, the significance level is calculated using the propagation weights achieved with these random networks. Thus, the P-value P_v for node v, with its propagation weight w_v, using n random networks, is defined as the fraction of instances where the propagation weight w_v^i under a random network model i is larger than the propagation weight given the input network (Equation 5.6).

$$P_v = \frac{\sum_{i \in (1,\ldots,n)} \mathbb{1}(w_v^i) + 1}{n+1} \tag{5.6}$$

where $\mathbb{1}(w_v^i)$ is the indicator function:

$$\mathbb{1}(w_v^i) = \begin{cases} 1 & \text{if } w_v^i \geqslant w_v \\ 0 & \text{otherwise} \end{cases}$$

5.5 Degree-preserving PPI network randomizations

The statistical test that we implemented relies on the generation of random network models. In order to compare directly between the propagation results using the input network and the random ones, the degree of the nodes in the random networks must remain as in the input network. This is because, in the standard formulation, the degree is used for determining the walking probabilities during the propagation. However, we developed three other implementations for calculating the walking probabilities, which rely on core and not only degree. Therefore, for these it is required to generate random network models which also preserve node core. For this purpose we explored two models for generating random degree-preserving PPI networks. The first is based on a simple edge swap, where two existing edges are "switched" such that the number of connections for each node is not changed. The second is based on so called dk-distributions, and is able to generate random networks with both local and global metrics, including core, that are highly similar to the ones in the input network.

5.5.1 Edge swap

The edge swap algorithm can be used to generate random degree-preserving networks (RDPN) by applying m random swaps to a given input network G. At each step, two edges (u,v) and (x,y) are chosen at random. They are removed from the new network G', and two new edges are created (u,y) and (x,v), unless they already exist in the input network G. This swap keeps the degrees of the nodes fixed. However, it can also disintegrate the graph such that the connectedness of the nodes is not kept anymore. To avoid that, the algorithm can be modified such that swaps that disintegrate the graph are not kept. As a result, the number of retained swaps is only *at most* m. Clearly, m must be large enough for the final network G' to be different enough from the input network G, and therefore truly random. The general recommendation for m depends on the size of the network, i.e. the number of edges, together with a swap factor $f = 100$, and is usually defined by $m = f * |E|$ [193]. In this work we used the connected double-edge swap algorithm, as implemented in the `Python` software `Networkx` [108].

5.5.2 *dk*-random graphs

While the edge swap algorithm preserves the degree of the nodes, it does not necessarily preserve other node metrics. Therefore, we explored another approach which generates RDPN which also aim to replicate the distributions of several other node metrics. Orsini et al. [212] developed a set of algorithms for generating dk-random graphs, which were made available to use in the `RandNetGen` (Random Network Generator) software tool[1]. They make use of the dk-series [177], which is defined for a graph G with n nodes, as a collection of distributions of the sub-graphs of G. These sub-graphs are in sizes $d = 0, 1, ..., n$ and the nodes are labeled by their degree in G. Figure 5.8(a) shows an example for the dk-distributions of a graph with four nodes. The nk distribution includes only one sub-graph which is the graph G itself. The $0k$ 'distribution' is defined as the average degree of G. The $1k$ distribution includes sub-graphs of size 1, i.e. it counts the number of nodes for each degree k in the graph. Therefore, the $1k$ distribution is the degree distribution of the graph G. In a similar way the $2k$ distribution is the joint degree distribution, and the $3k$ distribution, which consists of triangles, defines clustering. In general, the dk-distribution characterizes the correlations of nodes with degree k in sub-graphs of size d. It is also possible to consider dk-distributions with fractional $d \in (2, 3)$, which in addition to characterizing the correlations between nodes with degree $d = 2$, also fixes some properties which are characterized in $d = 3$ distributions.

The sequence of dk-distributions can be used to define a sequence of random graph sets. For every $d = 0, ..., n$ the dk-graphs are a set of all graphs with a given dk-distribution. For a given input graph G, these dk-graphs have the same dk-distribution of G and therefore share the graph property that is reflected in this dk-distribution. Hence, all $0k$ graphs will have the same degree average as of the nk graph (which is equivalent to G), while $1k$ graphs will have the same degree distribution, and so on. Figure 5.8(b) provides an example for such dk-graphs given a graph G of size D. Due to certain characteristics of dk-series, which are out of the scope for this work but are detailed in [212], any property of a graph G can be reproduced by a high enough value of d. Therefore, it is possible to sample dk-random graphs (from the set of all dk-graphs) for a graph G such that some property is reproduced with a desired accuracy. However, due to limitations in the sampling process, which we will not elaborate on here, for real networks, it is only possible to sample dk-random graphs for $d = 0, 1, 2, 2.1, 2.5$. $2.1k$-random graphs are defined as $2k$-random graphs with an average clustering of $\bar{c}$. $2.5k$-random graphs are defined as $2k$-random graphs with an average clustering $\bar{c}(k)$ of nodes of degree k. All of the dk-random graphs can be generated using some modified version of the edge swapping procedure, as depicted in Figure 5.8(c), while preserving the desired dk-distribution. For $d = 2.1, 2.5$ each swap is accepted with probability P, which is designed to reach the target value of average clustering for $d = 2.1$, or degree-dependent clustering for $d = 2.5$. As a matter of fact, when $d = 1$ the exact same edge swap procedure that was described in Section 5.5.1 is applied. As a result, for $d = 1, 2, 2.1, 2.5$ the node degree is always preserved.

1 http://polcolomer.github.io/RandNetGen/

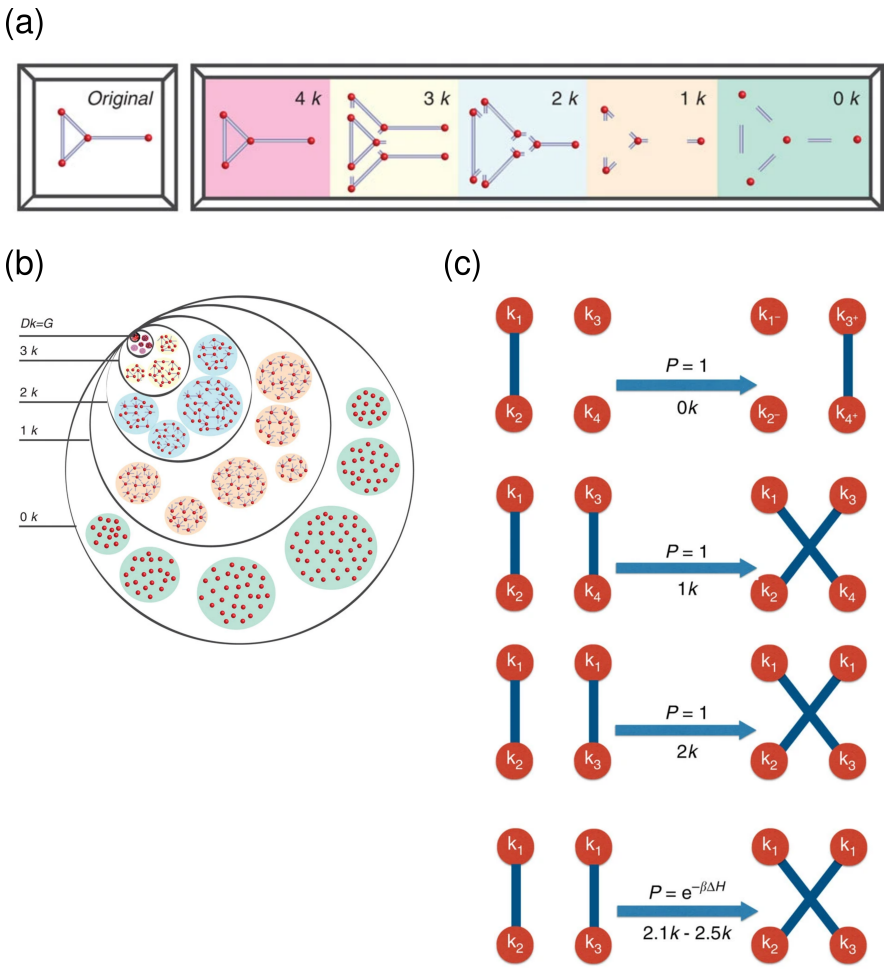


Figure 5.8: *dk*-distributions and graphs: (a) An example graph with four nodes and its dk-distributions. Each node is always marked with its degree (when the edges are not included in the sub-graph, only a short line is visible), except for the 0k distribution. The 4k-distribution is the graph itself. The 3k-distribution consists of the three sub-graphs of size three: one connecting nodes with degrees [2,2,3] and two connecting nodes with degrees [2,3,1]. The 2k-distribution specifies the number of sub-graphs of size 2 connecting nodes of different degrees: one connecting nodes with degrees [2,2], two connecting nodes with degrees [2,3] and one connecting nodes with degrees [3,1]. The 1k-distribution lists the number of nodes (sub-graphs of size 1) for each degree: one node with degree [1], two nodes with degree [2] and one node with degree [3]. The 0k-distribution is the average degree in the graph (which is 2). (b) An example for a hierarchy of dk-graphs, which are graphs with the same dk-distribution as for a graph G of size D. The set of 0k-graphs includes those with the same average degree and is thus the largest one. The set of 1k-graphs is a sub-set of 0k-graphs, because these are graphs with the same average degree (but not necessarily the same degree distribution). The higher d is, the smaller the set of graphs. The last set, where $d = D$, contains only the one graph G. (c) Edge swapping procedures for dk-graphs where $d = 0, 1, 2, 2.1, 2.5$. k indicates the node degree and P the probability of accepting the swap. For $d = 0$ a random edge from the graph is swapped with a new edge, i.e. an edge which does not exist in the graph. For $d = 1$ a random edge from the graph is swapped with another random edge from the graph. For $d = 2$ two random edges are selected from the graph such that the degree of two of the nodes (from different edges) are the same. For $d = 2.1$ and $d = 2.5$ the swap is accepted with probability P, which is derived from a simulated annealing procedure. Adapted from [212].

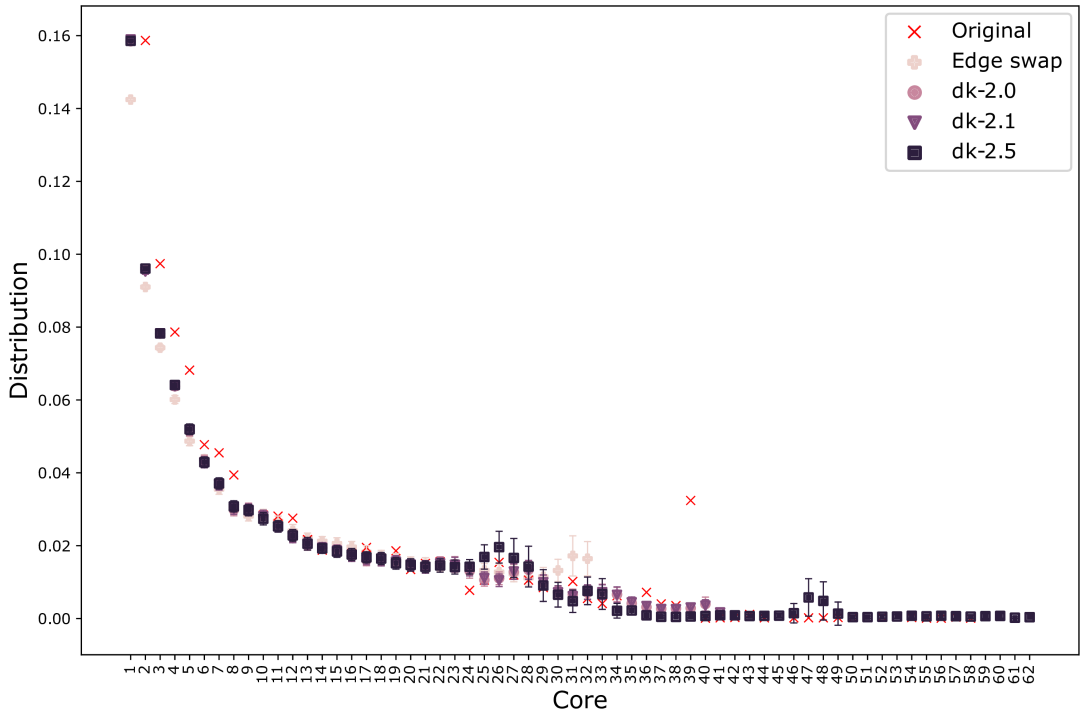


Figure 5.9: Core distributions: Distributions of core values in the CPDB PPI network ('original') and in 100 random networks generated with the edge swap algorithm or with dk-random graphs using $d = 2, 2.1, 2.5$. For the 100 random networks, and for every core value, each point represents the median and the standard deviation is depicted as confidence intervals.

Orsini et al. [212] applied their algorithms to six different networks,including one PPI network [248], and generated dk-random graphs for them. For each network they computed the following metrics: average degree, degree distribution, degree correlations, average clustering, and averaging clustering of nodes of degree k. Then, they compared these metrics for each network with the metrics for the corresponding dk-random graphs. They also computed a variety of other network metrics, such as core and betweenness centrality, and reported their means and deviations from the input networks. They showed that for almost all networks and all metrics, there is a convergence of the metrics with the increase of d. Depending on the metric, some already converged at lower d values, while almost all converged only when d reached $d = 2.5$, Thus, the generated dk-random graphs are able to replicate many important metrics of the input network.

In order to generate random networks that preserve the core distribution in the CPDB PPI network we used the `RandNetGen` package and generated dk-random graphs. Orsini et al. [212] found for another PPI network that the core metric already converged at $d = 2$, although for other interaction networks it only converged at $d = 2.5$. We therefore generated 100 dk-random graphs for the CPDB PPI network using $d = 2, 2.1, 2.5$. We did not generate networks for $d = 1$ as those would be equivalent to those generated using the edge swap procedure (Section 5.5.1). When $d = 0$ usually none of the node metrics are preserved, so we did not generate networks for this case either.

Finally, we compared the core distributions of the CPDB PPI network and the generated random networks, to examine whether the core metric is replicated in the random networks. Figure 5.9 shows the core distributions of the input network, as well as the random networks generated using the edge swap algorithm, and the dk-graphs with $d = 2, 2.1, 2.5$. In most core levels, the mean value of the 100 random networks reproduced well the value from the input network. The only exception is that when $d = 2.5$ some nodes reside in higher core levels ($59 - 62$) than in the input network (where the highest core level is 58). Otherwise, the difference between the three dk-graphs is almost undetectable, and all of them are also very close to the edge swap distribution in most core levels. On top of that, the standard deviation is usually very small, suggesting that most networks replicate well the core distribution in the majority of the levels.

In conclusion, both techniques for generating RDPN are able to preserve the degree and core distributions of the CPDB PPI network, with minor differences. While the dk-random graphs might be more appropriate for NetCore, due to the implementation of core-based normalizations, their generation requires longer computation times in comparison with the edge swap technique. The computation time also substantially increases with the size of the network, and therefore might not be suitable for larger networks. Accordingly, we implemented NetCore (see also Section 5.9) using only the edge swap technique, to allow for easy execution by other users. Nevertheless, for the purpose of this work we will compare the effect of the choice of technique on the results in Section 6.3.2.

5.6 Semi-supervised module identification

NetCore's propagation procedure results in a re-ranking of the nodes, that is each node is associated with a new weight and a significance level. These can be further used to identify sub-networks, which in turn represent modules that might be relevant to the phenotype in question. Previously suggested solutions to this problem either lack a significance level of the propagation results or direct connectivity between the nodes in the identified modules (Section 3.6.2). In NetCore, the goal is to produce modules by identifying a sub-set of nodes that are ranked highly after the propagation, have a significant propagation weight, and are interconnected in the PPI network. In order for these modules to be functionally relevant we applied a semi-supervised approach. The incorporation of prior knowledge allows us to enforce biological guiding on statistically significant propagation results in the search for interconnected sub-networks. This semi-supervised approach requires a collection of genes which are already known to be associated with the phenotype in question, typically based on previous studies. Such collections can be found, for example, in manually curated databases, or can be extracted from publications via text mining approaches. Once a set of genes is made available, it can be used in combination with the network propagation results to construct the final modules.

In order to allow the identification of modules also in the event where no prior knowledge is available, we also developed a procedure which is based only on the input weights. The aim is to generate a set of seed genes that are reasonably relevant to

the phenotype and therefore could be used as a "starting point" for identifying network modules. For example, if the propagation is to be executed on data derived from a differential gene expression analysis, then the seed nodes could be labeled as those with a significantly higher or lower expression, or as those with the highest fold change (e.g. the top 100 genes). The propagation itself can then be executed on the entire data, such that each node has a weight proportional to its expression level, and then the modules are identified for the selected seed nodes in combination with the results on the entire network.

The steps that are applied in order to identify modules in NetCore, given a set of seed genes, are first briefly summarized in three main points, and then described in detail:

1. Extract seed-induced sub-network (i.e. the seed genes and their interconnecting edges)

2. Extend seed-induced sub-network with nodes for which the following three conditions hold:

   a) Significant weight after propagation ($p < p_{max}$)

   b) Direct neighbor of at least one seed node

   c) High enough weight after propagation ($w > w_{min}$)

3. Separate extended seed sub-network into modules by identifying connected components.

As a first step, a sub-network that includes only the seed genes is extracted from the PPI network. We term this a seed-induced sub-network. In case no seed genes are available, the sub-network is extracted based on the original input weights. The genes are ranked according to their input weights, and the n top genes are used as seed genes. By default, n is set to 100, but this can be modified by the user, according to the data. The genes in the sub-network might not all be interconnected, as there might not be a direct path from every seed gene to every seed gene.

In the second step the propagation results are used to extend the seed-induced sub-network based on three principles. First, the genes must have a statistically significant propagation weight. The level of significance is set according to the applied permutation test, as will be further discussed in Section 5.7.2. Second, among those genes with a significant propagation weight, only those that are directly connected to at least one of the seed genes are further considered. Third, the weight of the genes after the propagation must be considerably large enough. We assume that the higher the weight post-propagation, the more relevant the gene is. A gene with a high propagation weight either already had a high initial weight, or gained a higher weight during the propagation process. Since the initial weights are propagated throughout the network, at the steady state all the nodes in the network will have some weight that is larger than 0 (due to the connectivity requirement). Therefore, the aim here is to discard nodes with a very small weight, that might still be statistically significant, yet are less likely to be biologically relevant. The minimum weight value that should be considered ap-

propriate depends largely on the initial weights, and the distribution of the weights post-propagation. Therefore, we decided to determine it based on the data, as will be described in Section 5.7.3.

Finally, once the seed-induced sub-network is extended with the finalized set of additional nodes, the final modules are defined as the connected components in the extended seed sub-network. Even though the goal is to connect all of the seed nodes in one module, those might still be separated in the extended seed sub-network. As a result, the connected components corresponds to the partition of the graph where each module includes as many seed nodes as possible within the topology of the extended seed sub-network.

5.7 Parameter selection in NetCore

The implementation of NetCore requires the setting of three different parameters. As with any other method, there is a need to set a value or threshold (for each parameter) for the optimal execution of the method. For NetCore we tried to set the parameters to the best of our abilities: 1) the restart parameter is optimized for the PPI network, 2) the P-value threshold parameter is set to the most strict one possible, and 3) the weight threshold parameter is based on the distribution of the propagation results. We acknowledge that setting the parameters is always a matter of trade-off between different requirements and therefore implemented the approach such that the user can decide on different parameters when necessary (see Section 5.9). In the following Sections we elaborate on the optimal settings for each one of the parameters. In Section 6.2 and Sections 6.4.5-6.4.6 we evaluate the performance given the settings of these parameters, and examine their influence on the results.

5.7.1 Restart parameter

In the RWR formulation, as defined in Section 4.2.2, the restart parameter α dictates the probability of the random walk to return to the input nodes and restart again. This allows us to control how much of the input weights will be diffused throughout the network. The lower the value, the less the walk restarts, and therefore more of the weight is spread in the entire network [52]. Yet, when the value is high, the diffusion of the weight exponentially decreases on the basis of distance from the source, which confines the weight to the local neighborhood of the source, even at steady state. Thus, if α is set too low then more weight is spread and more nodes will eventually gain higher weights at the final propagation step. On the other hand, when α is set too high, potential relevant nodes might end up being ranked lower, and as such relevant predictions could be missed. This means that the value of α essentially allows us to control the trade-off between finding novel phenotype-associated genes and including false predictions.

In general, for every PPI network, the restart parameter α has to be set individually. Once it is set for the network, it can then be used regardless of the initialization of the weights in the RWR formulation. Thus, α must only be optimized for the chosen

PPI network. Our aim therefore is to find an optimal value for the CPDB PPI network so that correct predictions can be made, but nevertheless the number of false ones is reduced. We wish to limit the diffusion such that the initial weight is mostly spread to nodes that are close to the source, and not throughout the entire network. This is especially necessary in PPI networks, where the average shortest path is usually very small [235]. In the CPDB PPI network more than 50% of the nodes are within only 3 "steps" from the rest of the nodes in the network (see Figure 5.2). Thus, for the purpose of identifying disease genes we decided to set $\alpha = 0.8$ in order to be able to still provide novel predictions, while reducing the number of potentially false ones. Furthermore, the semi-supervised module identification procedure in NetCore accounts only for direct neighbors of seed nodes (Section 5.6). Consequently, the value of α need not be very low in order for enough weight to be propagated to those neighbors. Thus, setting $\alpha = 0.8$ is sufficient for the addition of these nodes during the extension of the seed sub-networks.

5.7.2 P-value threshold

In Section 5.4 we introduced a permutation-based test for assessing the significance level of the weights of the nodes after the propagation. The test is based on a collection of random degree-preserving networks. The number of random networks that are generated determines the minimal possible level of significance that can be achieved. For n random networks, the lowest P-value is $1/(n+1)$. This P-value signifies that the weight of a node after the propagation, when using the input network, is always higher than the weights achieved when using any of the random networks. Therefore, we decided to use the P-values generated by the permutation test for the selection of non-seed nodes to be added during the module identification step. Since only nodes with a significant P-value should be considered, a minimal significance level must be set. Biran et al. [38], which showed that this approach performed best in the setting of network propagation, suggested to set the number of random networks at $n = 100$. Therefore, as a general rule of thumb, we also recommend using at least $n = 100$ random networks. In this case, we set the significance level at $p < 0.01$, as the minimal P-value is $p = 0.0099$. Additionally, we implemented NetCore such that the minimal significance level can be set by the user. For example, in order to include more potential nodes for the module identification step, it is possible to set a less strict level such as $p < 0.05$. On top of that, if the number of generated permutations is larger, than clearly the minimal P-value is lower, and the threshold can be set accordingly. However, the number of generated random networks is also limited by the time required to generate them (see Section 5.9). Therefore there is a trade-off between the time it takes to generate the set of random networks and the highest level of significance that can be achieved.

5.7.3 Weight threshold

In addition to the significance level of the propagation weights, we also exploited the weight value to determine which nodes should be considered for the module identification step. As previously discussed in Section 5.7.1, the amount of the weight that

is diffused from the source can be controlled via the restart parameter. Therefore, the weight that is accumulated in a node at the final propagation step depends on its initial weight, if available, and its position in the network, i.e. the weight which was propagated from its neighbors. Since the initial weights indicate experimental relevance to the phenotype in question, the final propagation weights also suggest towards potential relevance. Therefore, there is a need to set a threshold for the minimum weight that should be considered for identifying novel nodes. We estimated this threshold based on the distribution of the weights after the propagation. This distribution depends on the input weights, the network and the restart parameter, and therefore the threshold must be evaluated for every combination of those. Since we also only wanted to consider nodes with a significant P-value, we estimated the distribution only for those nodes with a significant P-value, rather than all the nodes in the network. Furthermore, as some of those nodes might be already labeled as seed nodes, we finally estimated the distribution only for those non-seed nodes which had a significant P-value. As a general rule-of-thumb we recommend to set the minimum threshold for the weight as the 75th percentile of this estimated distribution. This threshold allows us to make enough novel predictions and at the same time discard nodes which only accumulated a small amount of the propagated information, and therefore are less likely to be biologically relevant. Setting the threshold to a higher percentile naturally reduces the number of predictions, while a lower one increases them. As a result, the weight threshold also allows to control the trade-off between novel predictions and false ones.

5.8 Adaptation for edge-weighted networks

Network propagation can also be applied to weighted graphs, where each edge is scored with one weight. For example, in `MEXCOwalk` [5] an edge-weighted random walk was applied, and the weights were calculated for every pair of genes based on the mutual exclusivity and coverage in cancer mutation data. For the implementation of NetCore, we adapted the RWR formulation to allow for utilizing also edge-weighted interaction networks. The edge weights must be provided by the user. When available, we incorporated the edge weights into the adjacency matrix normalizations (Section 5.3). For the standard degree normalization, after the weighted adjacency matrix is multiplied by the reverse of the degree $A^{degree} = AD^{-1}$ it is further normalized such that the sum of its each column is 1 (and so the matrix is stochastic). For the three versions that utilize core, this procedure was already implemented for the unweighted version, therefore no further adjustments were required. In addition, we also provided an implementation for generating RDPN for edge-weighted networks using the edge swap algorithm (Section 5.5.1). To that end we modified the implementation of the algorithm from the `NetworkX` [108] library for `Python` and adapted it for edge-weighted networks. Given a pair of edges (u, v) and (x, y) that were to be swapped such that the new edges (u, x) and (v, y) were created, we assigned weights to the new edges according to: $w(u, x) = w(u, v)$ and $w(v, y) = w(x, y)$.

5.9 Implementation

NetCore is a command-line tool implemented in `Python` and is compatible with versions 3.6 and 3.7. NetCore is licensed under the MIT License and is freely available for download and use via `https://github.molgen.mpg.de/barel/NetCore`. NetCore is implemented using the `NetworkX` [108] `Python` library for graph analysis. The repository includes an implementation for the random walk with restart formulation of network propagation, using all the normalization schemes that are defined in Section 5.3. In addition, the repository also provides an implementation for generating RDPN in parallel using the multiprocessing module in `Python`. The user can choose to use the CPDB PPI network, or supply any other interaction network. In addition, the user must provide either a list of seed nodes or a list of weights to (some of) the nodes in the network, or both. The rest of the parameters are set to a default value for an optimized execution with the CPDB PPI network, as previously described in this Chapter. However, each one of those can be adjusted by the user. Finally, we provide the user a tutorial with a step-by-step guide of how to execute NetCore (which is available in the GitHub repository).

NetCore's run-time was tested (on a Linux machine with 64 cores) using the CPDB PPI network. We provide here the run-time analysis, which also includes the times needed for generating the random networks for the permutation test. The run-time depends mostly on the number of edges, yet is still feasible for most molecular interaction networks. The computation of the RWR formula, and the calculation of the steady state distribution (see Section 4.2.2), are implemented using the linear algebra module of the SciPy software for `Python` [299]. Given the size of the CPDB PPI network this computation is still very feasible and the running time for one such computation takes 30 seconds. Yet, with much larger networks this computation can rapidly increase in time. Since NetCore implements a permutation test that is based on random networks, the total running time depends on the number of permutations. Given 100 permutations, each one requires a computation of 30 seconds, and therefore a total of 50 minutes. The overall running time, including the module identification, requires no more than 60 minutes.

In addition, to execute NetCore's permutation test, we provided an implementation for generating RDPN. The running time for creating such networks depends both on the size of the network (number of edges) and a constant factor f, which controls the number of attempts to swap edges (see Section 5.5.1). For the CPDB network (114,341 edges) and a swap factor of $f = 100$, generating one random network takes up to 45 minutes. Since multiple networks can be generated at the same time, we provided a fast implementation that runs in a parallel fashion using `Python`'s multiprocessing module. Using that, the generation of 100 random networks requires a total running time of 90 minutes (instead of up to 75 hours without parallelization). The computation of the random networks needs to be executed only once for every input network, and can later be used repeatedly for running NetCore.

5.10 Summary

This Chapter provided a detailed overview on NetCore, the network propagation approach which we developed using node core. NetCore provides both a modified version of the RWR formulation and a semi-supervised procedure for identifying network modules after the propagation. We elaborated on the different components of the approach as well as the parameters which are required for its application. First we focused on the chosen PPI network from CPDB and explored its characteristics, and in particular the relationship between node degree and core within the network. We concluded that core is, in contrast to degree, more robust to the study bias, and presented three modifications to the RWR formulation which are based on node core. Second, we defined a statistical test which is based on RDPN and described two techniques for generating such networks, namely edge swap and dk-random graphs. We observed that both techniques were able to replicate well the core distributions of the CPDB PPI network and were therefore suitable for NetCore. Third, we provided a procedure to identify functionally relevant modules by incorporating the propagation results with prior knowledge. These modules, which integrate genes of interest with novel predictions, are likely to be relevant for the phenotype in question. Finally, we determined the optimal parameters for applying NetCore, all of which allow us to control the trade-off between novel and false predictions. Our implementation thus contributes both in producing less biased network propagation ranking and in detecting phenotype-associated modules which extend well-known genes together with novel predictions.

6 Evaluations and Performance

In this chapter we evaluate the NetCore approach which was described in the previous chapter. The evaluations are separated in two, the first is designed to assess the performance of the different normalization schemes, and the second of the identified modules. For both problems, of identifying disease genes and disease modules, there is no standard way of evaluation, and therefore we had to construct disease-gene sets which are based on previous studies and could be used for evaluating NetCore. In addition, we also examine the influence of the interaction network and the different parameters on the performance.

6.1 Evaluation of adjecency matrix normalizations

Evaluation of disease-gene predictions is not straight forward, since there is no real "gold-standard" available for any disease. Nevertheless, in order to evaluate the performance of the different normalization schemes that we implemented in NetCore, we had to compile a set of true-positive genes that could be used for the evaluation. Furthermore, to assess the performance, the prediction problem had to be regarded as a classification problem, such that predicted genes are classified as positives, and the rest as negatives. Then, a cross-validation scheme could be applied to evaluate the results.

6.1.1 Cross validations and performance measures

Given a set of n positive genes, we define a cross validation procedure as follows. The set is sub-sampled into a training set and a validation set, such that the size proportion is $1:4$ respectively. I.e. 80% of the genes in the set are used for training, and the remaining 20% are used for testing. Each training set is used to execute the RWR, such that the genes in the training set are scored with 1 and the rest of the genes in the network with 0. For NetCore this is repeated four times, once for each one of the normalization schemes. The weights at the steady-state are extracted and assigned with P-values according to NetCore's permutation test. Then, all the genes in the network are sorted according to their P-values, such that the classification can be made according to the level of significance. Given a minimal P-value p, every gene g_i with a propagation weight w_i that has a P-value $p_i < p$ is classified as positive, and the rest of the genes as negatives. Following that, the false-positive rate and true-positive rate are calculated for a series of P-values $p = 0.01, ..., 1.0$ and the receiver operating characteristic (ROC) curve is produced. Since the test set is small relative to the size of the network, the negative set has to be balanced accordingly. Therefore, a smaller negative set, in the size of the test set, is randomly sampled from the remaining nodes in the network (i.e. those that are neither in the training nor in the test set), such that the degree of the

Source	Disease (Trait)	Gene Set Size	Coverage in PPI network
GWAS Catalog [176]	Body Mass Index	177	97
	Breast Cancer	113	73
	Crohn's Disease	436	227
	Height	387	234
	Prostate Cancer	324	168
	Rheumatoid Arthritis	124	88
	Schizophrenia	395	226
	Systemic Lupus Erythematosus	137	90
	Type 2 Diabetes	124	78
	Ulcerative Colitis	345	203
	Vitiligo	111	76

Table 6.1: The 11 GWAS gene sets were extracted from the GWAS catalog [176] and applied to estimate the performance of NetCore.

sampled nodes is matched to the degrees of the nodes in the test set. In total, the entire procedure was repeated five times (for each normalization scheme) and the five ROC curves were then averaged into one ROC curve to get a consensus curve, from which the area under the ROC (AUROC) was calculated. We use the AUROC as the performance measure.

6.1.2 Performance on GWAS gene sets

To construct reasonable gene sets for the evaluation of NetCore we used the GWAS catalog [176] which is a curated collection of GWAS studies and their results. For a given trait or disease it provides information about genomic locations that were found to be significantly associated with it. Most of the genomic locations can be mapped to one or more genes. Hence, given a trait or disease, a list of genes that are associated with it can be extracted directly from the catalog. Such list can serve as a set of true positives for our cross validation scheme. For that purpose we extracted the genes for nine diseases and two quantitative traits. Table 6.1 lists the sizes of the gene sets, which range between 20 and 500 genes, and their coverage in the CPDB PPI network. The same 11 gene sets were also previously used to evaluate the performance of 21 PPI networks in the context of network propagation, including CPDB [133].

For each one of the four normalization schemes of the adjacency matrix we applied the 5-fold cross validation scheme to the 11 GWAS gene sets. We compared the results for standard degree normalization with three other normalization schemes using: core only, the difference between core and degree, and the ratio between core and degree. For each gene set we used 80% of the genes as training set and 20% of the genes as validation set. We used a binary node initialization scoring scheme, where the genes

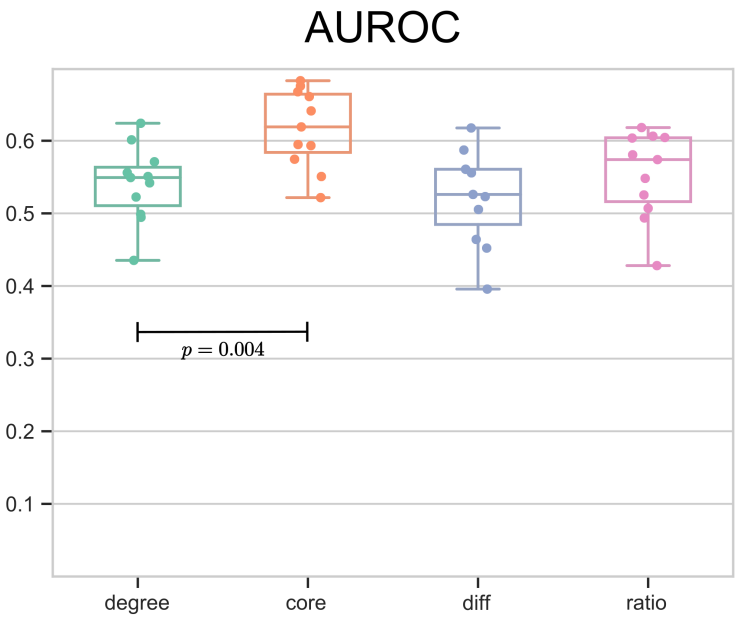


Figure 6.1: Performance of adjacency matrix normalizations: AUROC for 11 GWAS gene sets using different normalization schemes for the RWR matrix normalization. Degree = standard normalization based on node degree; core = normalization based on core; diff = normalization based on difference between degree and core; ratio = normalization based on ratio between core and degree. P-value was computed with a paired Wilcoxon test using the AUROC values of degree and core.

in the training set were scored with 1, and the rest of the genes in the PPI network with 0, and then computed the performance on the validation set after propagation. We calculated the average ROC curve and the AUROC (see Figure A.1). The results from all gene sets were compared based on the AUROC values of each one of the four normalization schemes. The results are displayed in Figure 6.1. It can be seen that for most of the gene sets, the core normalization achieved the highest AUROC. On average there is a significant improvement when using core- instead of degree-based normalization (Wilcoxon signed-rank test, $P = 0.004$). In fact, in all but one of the gene sets the AUROC was higher for core- than for degree-based normalization.

We also explored the degree and core of the genes in the 11 GWAS sets (see Figure 6.2) and observed that for all gene sets the variation of core values is much smaller than that of the degree. The coefficient of variation (CV) for all the genes in the GWAS sets is twice higher for the degree than the core (CV degree= 1.9, CV core= 0.9). Additionally, we note that the degree distributions include many extreme outliers, which is not the case for the core distributions. This points to the high degree bias around well-studied genes, and in particular disease genes. Indeed, there are some genes with a high degree that are present in many of the GWAS sets. For example, *HLA-B* (Major Histocompatibility Complex, Class I, B), which has a degree of 135 and core of 35, is present in six of the 11 sets.

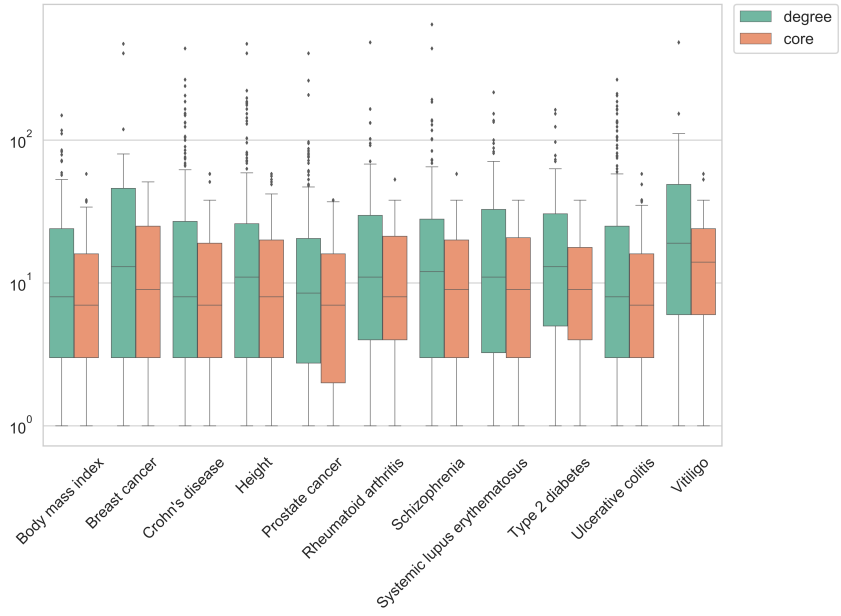


Figure 6.2: Degree and core of genes from 11 GWAS gene sets: Box plots of degree (green) and core values (orange) for the genes in 11 GWAS gene sets. X-axis denotes the phenotypic traits and diseases.

6.2 Influence of restart parameter

The restart parameter α defines the probability of the random walk to restart again, as previously discussed in Section 5.7.1. Since it enables the control of how much of the initial weight will be diffused throughout the network, the results after the propagation can vary, depending on the value that was set. For the CPDB PPI network we decided to set $\alpha = 0.8$ (see Section 5.7.1). In order to estimate the influence of the parameter on the performance of the normalization schemes we compared between three cases: low value ($\alpha = 0.3$), intermediate value ($\alpha = 0.5$) and a high value ($\alpha = 0.8$). We tested the performance for these three values when identifying genes from the previously described GWAS gene sets. We repeated the same cross validation scheme for each α value and evaluated the performance based on the AUROC measure. The results are displayed in Figure 6.3(a).

Overall, for all normalization schemes the highest performance, on average, is achieved when $\alpha = 0.8$, with the core-degree difference being the only exception, where $\alpha = 0.5$ is slightly better. Specifically for the core normalization, there is a small increase in the average AUROC with the increase of α, which reaches above 0.6 only when α is set to 0.8. By this measure, we confirm that setting α to 0.8 and applying core normalization generated the best results for the GWAS data sets.

For the significant genes that were recovered using core normalization we also calculated the ratio between the number of those which belong to the input GWAS gene set and the total number. We compared this ratio for the 11 GWAS sets, as depicted in Figure 6.3(b), and found that in five of the 11 sets the highest ratio was achieved for

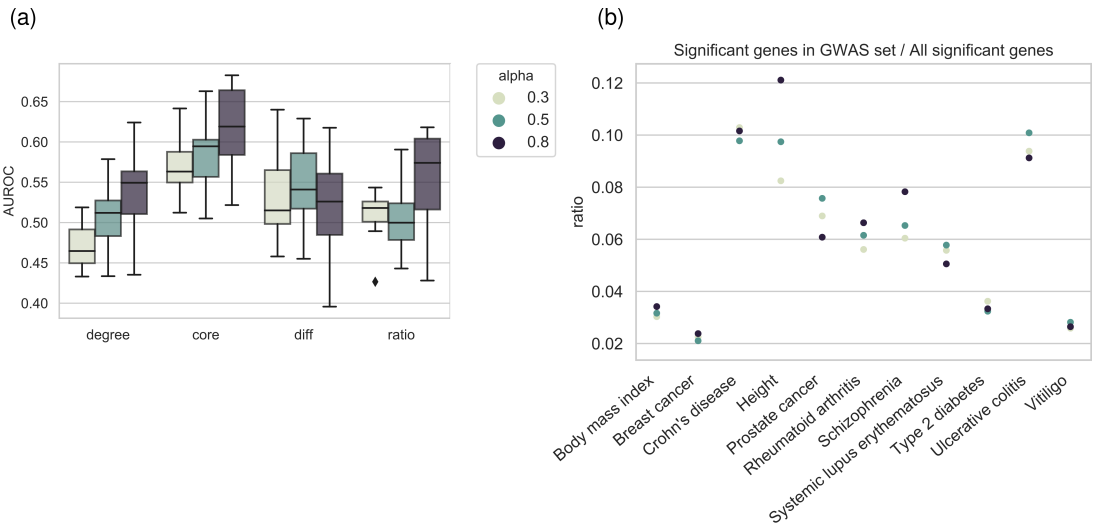


Figure 6.3: (a) The performance of the different normalization schemes in NetCore when identifying 11 GWAS gene sets for three different values of the restart parameter α: 0.3, 0.5 and 0.8. (b) For core normalization only: the ratio between the number of significant genes that were reported by NetCore and belong to the input GWAS gene set, and the total number of significant genes that were reported by NetCore.

$\alpha = 0.8$. The ratio varies depending on the gene set: it is the lowest for breast cancer genes (0.02) and the highest for genes that are associated with height (0.12). For five of the gene sets there is barely any difference in the ratio between the three α values, whereas for the rest of the sets the difference is rather minor. However, in some cases the number of significant genes that are from the GWAS set hardly changes, whereas the number of total significant genes is reduced. For example, in height, the number of significant genes that are from the GWAS set is reduced by 10% (from 39 at $\alpha = 0.3$ to 35 at $\alpha = 0.8$), whereas the total number of significant genes is reduced by almost 40% (from 473 at $\alpha = 0.3$ to 289 at $\alpha = 0.8$). This indicates that setting a higher value for α reduces the false positive predictions, while maintaining most of the true positive ones.

6.3 Influence of interaction network

The interaction network is one of the two main components in network propagation. Consequently, the choice of the network is crucial and can have a direct effect on the results. CPDB is a meta-database, which contains PPIs from multiple resources (see Section 5.2), and its performance in identifying disease genes via network propagation exceeded most of the other 20 networks which were compared by Huang et al. [133], including smaller networks that were derived from data-bases contained within CPDB. Nevertheless, we sought to estimate the influence of the CPDB PPI network on NetCore's results. In addition, since NetCore's permutation test relies on the production of RDPN for the interaction network, we also evaluated how the generation of those affects the results.

6.3.1 Comparison of CPDB versions

To explore the influence of the PPI network from CPDB we compared between three different versions which can be extracted from CPDB: 1) the high confidence PPI network with 10,586 nodes and 114,341 interactions, 2) the full PPI network with 16,526 nodes and 264,493 interactions, and 3) the full edge-weighted PPI network, making use of the `IntScore` [143] weights for the edges (see Section 5.2). For the edge-weighted network we adapted the adjacency matrix normalizations in order to incorporate the weights directly into the RWR formula (see Section 5.8). We used each network for the identification of 11 GWAS disease gene sets (Table 6.1) and evaluated the predictions by comparing the AUROC.

Figure 6.4 displays the results for two restart parameters: (a) 0.5 and (b) 0.8. Although we previously showed that setting the restart parameter to 0.8 performs best for the high confidence network, we nevertheless compared the results also for a lower value. Huang et al. [133] previously proposed a linear model which predicts the restart parameter based on the number of interactions in the network, where larger networks should apply a smaller restart parameter. Therefore, since the full CPDB network is much larger than the high confidence one, we also used a smaller restart parameter for assessing the networks' performance.

On average, for almost all normalization schemes, the high confidence network outperforms the full network, with and without edge weights. The only exception is for the ratio-based normalization, where, on average, the performance of the full network (without weights) is slightly improved in comparison with the high confidence one. Despite the difference in size, the performance for both the high confidence network as well as the full one is improved when the restart parameter is set higher ($\alpha = 0.8$). Furthermore, core-based normalization outperforms the other ones, for both versions. The performance of the full edge-weighted network is however strikingly worse and the differences between the normalization schemes are not as prominent as for the other two networks. We thus conclude that incorporating the edge weights directly into the propagation formula is not beneficial, and in fact using the weights to produce the high confidence network results in the best performance.

6.3.2 Influence of randomized networks

In Section 5.4 we presented the statistical test that we applied in NetCore, which is based on a permutation test of the input network. We suggested two different techniques to generate permutations of the input network that also preserve the distributions of degree and core: the edge swap algorithm (Section 5.5.1) and random dk-graphs (Section 5.5.2). We compared the distributions of the core values for the different random networks and concluded that the core distribution from the original network is well replicated. Despite the difference being slightly more observable when using the edge swap algorithm, in comparison to random dk-graphs with $d = 2.5$, we nevertheless implemented NetCore using the edge swap algorithm, mostly due to the extremely long running times for generating random dk-graphs, and in particular when setting $d = 2.5$. We did, however, evaluate the performance of NetCore for identifying

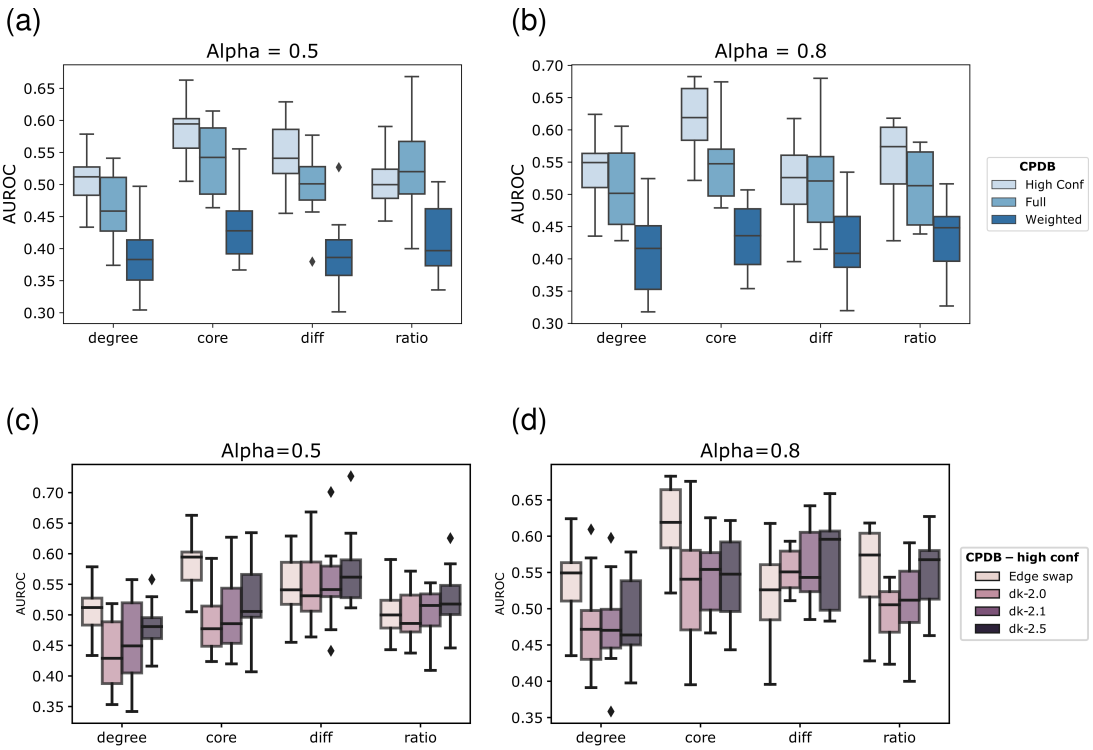


Figure 6.4: The influence of the chosen network and random network generations: Three different versions of the CPDB PPI network were compared, using two different restart parameters: (a) $\alpha = 0.5$ and (b) $\alpha = 0.8$. For the high confidence PPI network from CPDB, NetCore was applied using four different versions of random network generation: edge permutations and random dk-graphs using $d = 2, 2.1, 2.5$, using two different restart parameters: (a) $\alpha = 0.5$ and (b) $\alpha = 0.8$.

disease genes when using these two techniques for generating permutations of the high confidence CPDB PPI network.

The performance, shown in Figure 6.4(c-d) for two different values of the restart parameter, varies depending on the normalization scheme that was applied. When $\alpha = 0.5$, for degree and core normalizations, on average, the performance of the edge swap permutations exceeded the performance of dk-graphs for all d values, with core normalization outperforming degree in all cases. The only exception is for the genes in the rheumatoid arthritis set, where, for all d values, the performance achieved for core normalization exceeded the one with edge swap permutations. When $\alpha = 0.8$, the overall performance, for both techniques, and in all normalization schemes, is increased, as seen previously when comparing the different version of CPDB. When comparing the results for the difference- and ratio-based normalizations, for both α values, the best performance is achieved when using dk-graphs, and specifically when setting $d = 2.5$. In these cases, the performance of the edge swap permutations is comparable with that of $d = 2.0$, and in most cases the overall performance for dk-graphs increases with higher values of d. Furthermore, the performance for the difference-based normalization is higher, in all cases, than of the ratio-based one. All in all, despite the slight improvement in performance for dk-graphs, which was mostly limited to the ratio- and

difference-based normalizations, we conclude that the performance of the core-based normalization is superior with the edge swap technique.

6.4 Evaluation of module identification

Once we established core normalization as the best performing method, we sought to identify network modules based on its propagation results. To enforce biological guiding of the network propagation through incorporation of prior knowledge, we applied a semi-supervised approach for module identification. As network modules are not very well defined (see Section 3.4.1), there is also no standard way for evaluating them either. Furthermore, the module is always a trade-off between false and novel predictions, and therefore various criteria can be used to estimate its relevance. For this work we focused on two different criteria. The first is a very straightforward criterion, showing functional content and disease relevance, using over-representation analysis. The second criterion introduced in this work is connectivity regarding known disease genes, which we measured via entropy. We elaborate further on both criteria in the following Sections, and apply them to the modules that were identified for the 11 GWAS genes. To exemplify the novelty that is possible via NetCore's modules we focus on the results for type 2 diabetes. In addition, as NetCore's module identification step requires to set two parameters (see Sections 5.7.2-5.7.3), we also evaluate the influence of these parameters on the results.

6.4.1 Evaluation via over-representation analysis

Over-representation analysis allows us to identify if a computed set of genes, e.g. a network module, is statistically significantly enriched in another pre-defined gene set, for example a pathway, given a background list of all genes. The statistical significance is obtained via a hypergeometric test, where a P-value is calculated based on the number of identifiers that are present in the computed gene set and in the pre-defined gene set. This type of analysis is available via the CPDB web server [118]. CPDB includes a total of 5,436 pathway gene sets from 12 different resources: Pharmgkb [318], EHMN [175], HumanCyc [249], WikiPathways [149], INOH[1], Netpath[2], Reactome [139], SignaLink [154], KEGG [146], BioCarta[3], SMPDB [93], PID[4]. In this work, we applied over-representation analysis via CPDB to the genes from the different modules that were identified by NetCore. We used a minimum module size of 10 genes for the analysis. Furthermore, we used all the genes from the high confidence PPI network in CPDB as background, and extracted only the significantly enriched pathways. A minimum overlap of two genes between the input list and the pathway gene list was required. Only pathways with a P-value of at most 0.01 were extracted, and we used the Q-values, which are corrected for multiple testing using the false discovery rate (FDR) method [35], as a measure for the level of enrichment.

1 http://www.inoh.org
2 http://www.netpath.org
3 http://www.biocarta.com
4 http://pid.nci.nih.gov

6.4.2 Evaluation via a connectivity measure

The module identification in NetCore aims to connect well-known disease genes with novel ones. This follows the guilt-by-association principle [211], where interconnected genes share the same function. Thus, novel disease genes that are connected with known disease genes are more likely to contribute to the disease than totally unrelated genes. In order to evaluate the connectivity of the identified modules, we measured the entropy of the seed nodes distribution with respect to the modules and compared it to the maximum possible entropy. If, for example, after network propagation two disease genes remain in different unconnected modules, the connectivity is lower than in the case where a novel gene, which connects the two disease genes, is included, such that they are all connected in one single network module.

Since we defined the modules to be the connected components of the extended seed sub-network (see Section 5.6), we calculated the entropy of the seed nodes contained in these $M = \{M_1, \ldots, M_m\}$ modules by:

$$E = -\sum_{i \in (1,\ldots,m)} p_{M_i} \log p_{M_i} \tag{6.1}$$

such that $p_{M_i} = \frac{k_{M_i}}{n}$, where k_{M_i} is the number of seed nodes in module M_i, m is the total number of modules and n is the total number of seed nodes. The maximum entropy, calculated by $E_{max} = \log n$, reflects the case where all seed nodes are in different modules. Therefore, we measured the connectivity as the difference between E_{max} and E, which reflects the distance from the maximum entropy. The larger the distance, the more seed nodes are interconnected in the same modules, and the nodes are less distributed over a large number of smaller modules. The same calculation can also be applied to the connected components from the seed sub-network. Thus, the connectivity of the seed sub-networks and the extended seed sub-networks can be directly compared.

6.4.3 Performance on GWAS gene sets

After applying a binary scoring scheme to the genes in the 11 GWAS gene sets, and applying core-normalized propagation, we also used the genes as seed nodes and extended their induced sub-networks according to the propagation results. For each gene set we applied the following workflow. First, we extracted the sub-network that connects the seed genes only, which we further refer to as seed sub-network. Then, we extended the seed sub-network by adding neighbors of seed nodes, based on the propagation results. Namely, we only added neighbors if their weight after the propagation was larger than w_{min}, and their P-value was significant ($p < 0.01$). w_{min} was computed based on the data and was therefore determined for each gene set separately, according to the distribution of the weights after the propagation, as described in Section 5.7.3. We added the new nodes such that we also added all of their respective connections from the PPI network to other nodes in the sub-network.

We note that for many of the GWAS gene sets, the majority of the genes are not directly connected to one another in the seed sub-networks (see Figure A.2). In fact, in

most cases, the majority of the genes have no connections to any of the other genes in the seed sub-network. Some are connected only to a few genes in rather small components, while the rest of the genes are usually connected in one bigger component. For some gene sets the biggest component includes a larger number of genes, whereas in other cases it is rather small, usually depending on the size of the entire set. Even for the largest gene set, of Crohn's disease, only 40% of the genes are connected in one component in the seed sub-network. On top of that, the number of connections in this component is still rather small, with most of the genes having less than five connections. Likewise, for the gene set that is associated with body mass index, only 3% of the genes are connected in the seed sub-network.

Extending the seed sub-networks resulted in larger connected components, with more seed nodes connected to each other through the intermediate nodes that were added to the seed sub-networks (see Figure A.3). For all 11 GWAS sets, the number of nodes, shown in Figure 6.5(a), as well as the number of edges, in Figure 6.5(b), increased in the final modules in comparison to the largest components in the seed sub-networks. The number of nodes that are included in the largest module is higher by a factor between 2.2 and 10.6. This is due to more seed nodes being connected in one component, as well as to the addition of non-seed nodes to the sub-networks. This is further emphasized when looking at the number of nodes in the largest module that are from the input seed list, i.e. from the 11 GWAS gene sets, versus the non-seed nodes. In Figure 6.5(a), the number of seed nodes in the largest module (in orange) is always higher than the number of non-seed nodes (in gray). Nevertheless, there are always at least a few additional non-seed nodes which in turn serve as potential novel predictions.

Not only are more seed genes connected to each other in the largest modules (see Figure A.4), but also the number of connections is increased. Figure 6.5(b) shows the number of edges in the largest component of the seed sub-network versus the largest module, indicating that the number of edges in the largest module is larger by factors between 2.4 and 17.4. As a numerical indicator for the connectivity among the seed nodes, we measured the entropy derived from the different components of the sub-networks and compared it to the maximum entropy, where each seed node is in its own component. Figure 6.5(c) shows that the connectivity of the seed nodes is always higher for the extended seed sub-networks, by factors ranging from 2.0 up to 17.86. We also estimated the functional relevance of the largest modules by applying over-representation analysis. We compared the enrichment of the nodes in the largest modules with the enrichment of the nodes in the largest component in the seed sub-network (see Figure A.5). In most of the GWAS gene sets there is an increase in the enrichment of the genes that are in the modules. That is to say, the nodes that were added to the seed sub-networks usually participate in the same pathways as the seed nodes, which is also evident by the larger overlap with the pathway genes. For three of the gene sets, namely body mass index, schizophrenia and type 2 diabetes, the size of the largest component in the seed sub-network was too small to detect enriched pathways. In these cases NetCore provided largest modules with interconnected seed genes which could only then be associated with some function based on the pathway enrichment.

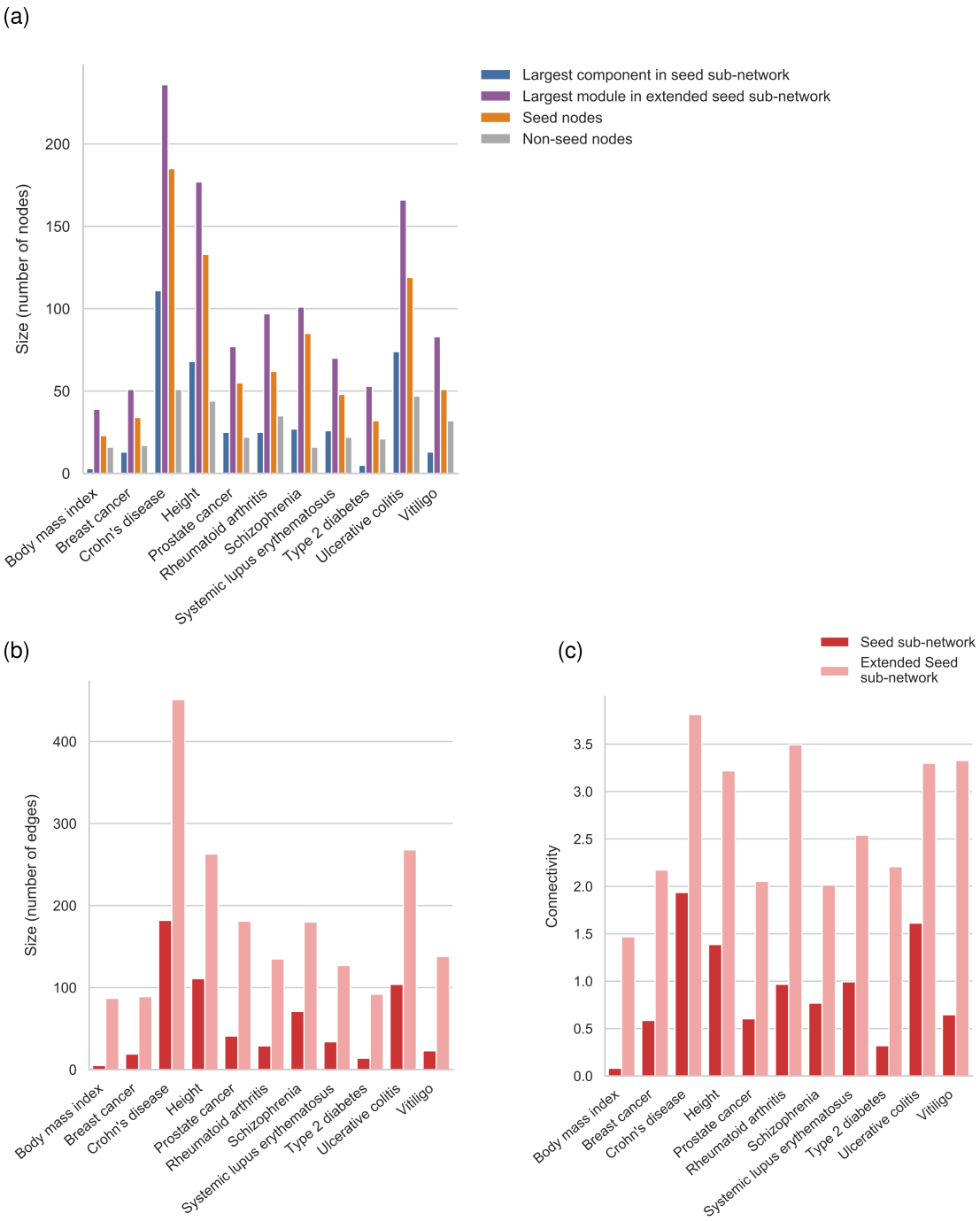


Figure 6.5: (a) Number of nodes for each GWAS gene set in : 1) the largest connected component for the seed sub-network (in blue), 2) the largest module using in the extended seed sub-network from NetCore (in purple), 3) the number of seed nodes in the largest module (orange), 4) the number of non-seed nodes in the largest module (in gray). (b) The number of edges in the seed sub-network and in the extended seed sub-network. (c) Connectivity of modules in seed sub-network and of modules in extended sub-network after network propagation. the connectivity is measured by the difference in entropy from the maximum entropy and the respective modules.

The 11 GWAS sets that were used for the evaluation of NetCore are associated with traits and diseases that are extremely different from each other, both on a genomic level, but also in the mechanisms that drive them. Therefore, a comprehensive examination of the results, in combination with a vast understating of the disease or trait, is necessary for the full functional evaluation of the modules and the novel predictions. Unfortunately, this would require expert knowledge which we could not provide for this work. Instead, to demonstrate the potential relevance of NetCore's results we present in the next Section a short example that focuses on type 2 diabetes, which aims to suggest novel disease-gene candidates for further investigations.

6.4.4 Focus on Type 2 Diabetes

We focus on the results for the type 2 diabetes genes from the GWAS catalog. Figure 6.6(a) visualizes the seed sub-network, which includes 78 seed nodes and 14 edges only. Most nodes are not connected to one another, and the connected components are all small, in sizes that range between two and five (connectivity measure of 0.3). Clearly, it is difficult to extract a functional module that is relevant to the disease based on this sub-network alone. Therefore, we added to the sub-network intermediate nodes which are connected to seed nodes in the PPI network, depending on their propagation results with NetCore ($p < 0.01$, $w > 0.015$). This resulted in an addition of 39 nodes and 78 edges to the sub-network. Figure 6.6(b) visualizes the seed extended sub-network, which consists of 15 connected components, in sizes between two and 53 (connectivity measure of 2.2). The largest connected component, displayed in Figure 6.6(c), consists of 53 nodes, out of those 32 are from the input GWAS seed set and 21 are not, with a total of 64 edges. Far more seed genes are now interconnected, which corresponds to a 6.9 higher level of connectivity. In addition, the seed genes are now also connected to other genes that serve as novel predictions for the disease. For example, *NTRK1* and *APP* are both connected to many of the seed genes and also have the highest weights after the propagation (among the non-seed genes that were added to the sub-network). On top of that, *UBXN7* which is ranked third (among the non-seed nodes after the propagation) is also included in the network, and is connected to both *NTRK1* and *APP*, as well as to other seed genes. *UBXN7* is already associated with body mass index according to the GWAS catalog.

We evaluated the functional relevance of the largest module from NetCore via over-representation analysis. Figure 6.7 shows the 20 most enriched pathways ($Q-value < 0.012$) for the module, and for each pathway the number of nodes from the module that are part of this pathway, and whether they were included in the original seed list or not. In many relevant pathways the amount of novel candidates is even higher than the amount of original seed genes. Some of the identified novel predicted genes participate in more than one of the most enriched pathways. For example, *IGF1*, *IGF2* and *LEPR*, which have all previously been associated with diabetes [16, 63, 259, 331]. These genes are enriched with pathways related to leptin, which has also been linked to diabetes before [75, 148, 306], as well as some IGF (Insulin Like Growth Factor) related pathways. Additional genes include members of the JAK/STAT signaling path-

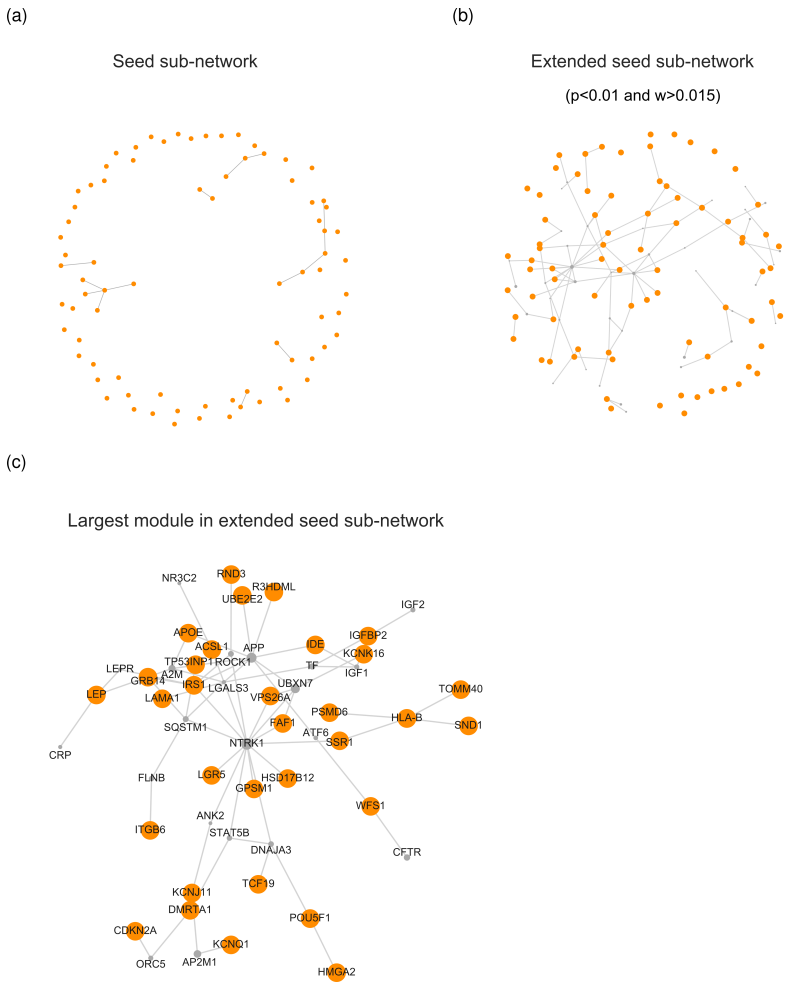


Figure 6.6: Sub-networks for type 2 diabetes genes from the GWAS catalog and NetCore: Orange nodes show genes in the original seed list, and gray nodes show significant genes that were added by NetCore. Graphs show: (a) seed sub-network; (b) extended seed sub-network, and (c) the largest module of the extended seed sub-network. In (b) and (c) the size of the nodes is proportional to the weight after propagation.

way, e.g. *STAT5B* and *ROCK1*, where it has been previously argued that this pathway is dysregulated in metabolic diseases including obesity and diabetes [79].

6.4.5 Influence of P-value threshold

The P-value criterion is set based on the statistical test that we chose to apply for NetCore. It is based on a permutation test, which requires the generation of random networks, and is therefore dependent on the number of executed simulations. We chose $p < 0.01$ as our criterion since it is the most stringent one, given $n = 100$ random networks. To show how robust the modules are to changes in this criterion we examined how the size of the extended seed sub-network (measured by number of nodes, number of edges, and number of nodes in largest module) depends on the chosen P-value in the case of the 11 GWAS data sets. Figure 6.8 displays the results for 10 P-values

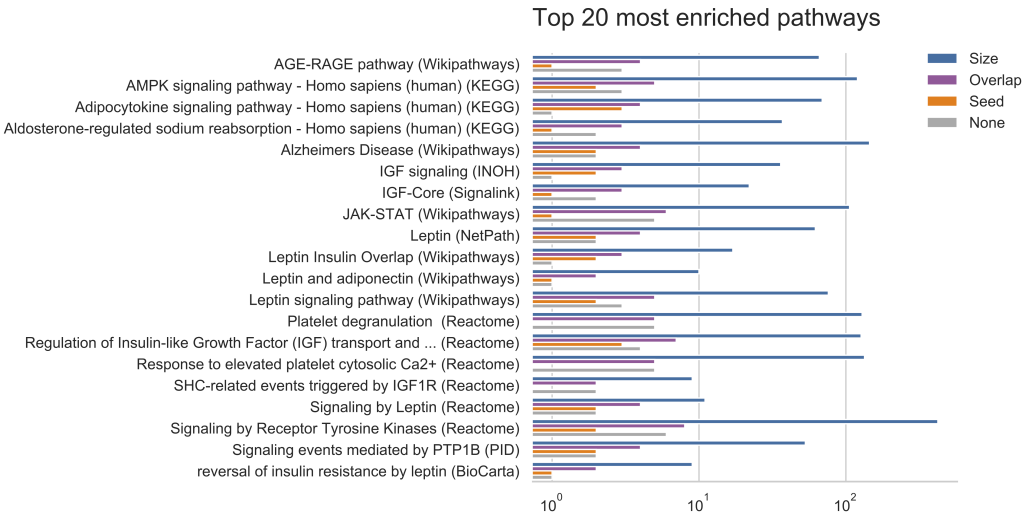


Figure 6.7: The most enriched pathways for the genes in the largest module for type 2 diabetes: In blue is the size (number of genes) of the entire pathway (according to CPDB), and in purple is the overlap of the genes from NetCore's largest module with the pathway genes. Orange indicates genes are from the seed list, and gray not.

ranging between 0.01 and 0.1, where two potential thresholds of $p < 0.01$ (red) and $p < 0.05$ (purple) are marked. The weight threshold was fixed for all P-values and was set according to the 75th percentile from the weights distribution, as described in Section 5.7.3. We note that the difference in the number of nodes, given the highest and lowest P-values which were tested, varies between 43 and 102. That is, the number of nodes at $p < 0.1$ was increased in comparison with $p < 0.01$ by between 19.1% and 83.6%, with the majority of the gene sets increasing by up to 40%. For most gene sets the number of nodes already stabilizes with $p < 0.05$, i.e. only a few more nodes are additionally added in comparison with $p < 0.01$. Beyond this level the number of edges increases greatly, and the modules increase so that it is more difficult to process them and provide reasonable candidates for further studies. To conclude, just as the P-value determined the balance between correct and false predictions (in Section 6.1.2) it also controls the addition of novel predictions to the final modules. Therefore, the value is set to include only those predictions which are relevant with the highest confidence. If the value is set higher, than more false predictions are potentially added, which also increases the size of the modules beyond desire.

6.4.6 Influence of weight threshold

Since the weights after the propagation depend on the input weights, we could not set a general threshold for the minimum weight we wished to consider. Therefore, we decided to set the minimum weight according to the distribution of the weights after the propagation. First, we explored the distribution of the weights after the propagation for the 11 GWAS gene sets (see Figure A.6). These distributions were generated using only those nodes with a significant P-value ($p < 0.01$) that were not included in the 11 GWAS gene sets, and were therefore potential candidates to be added to the

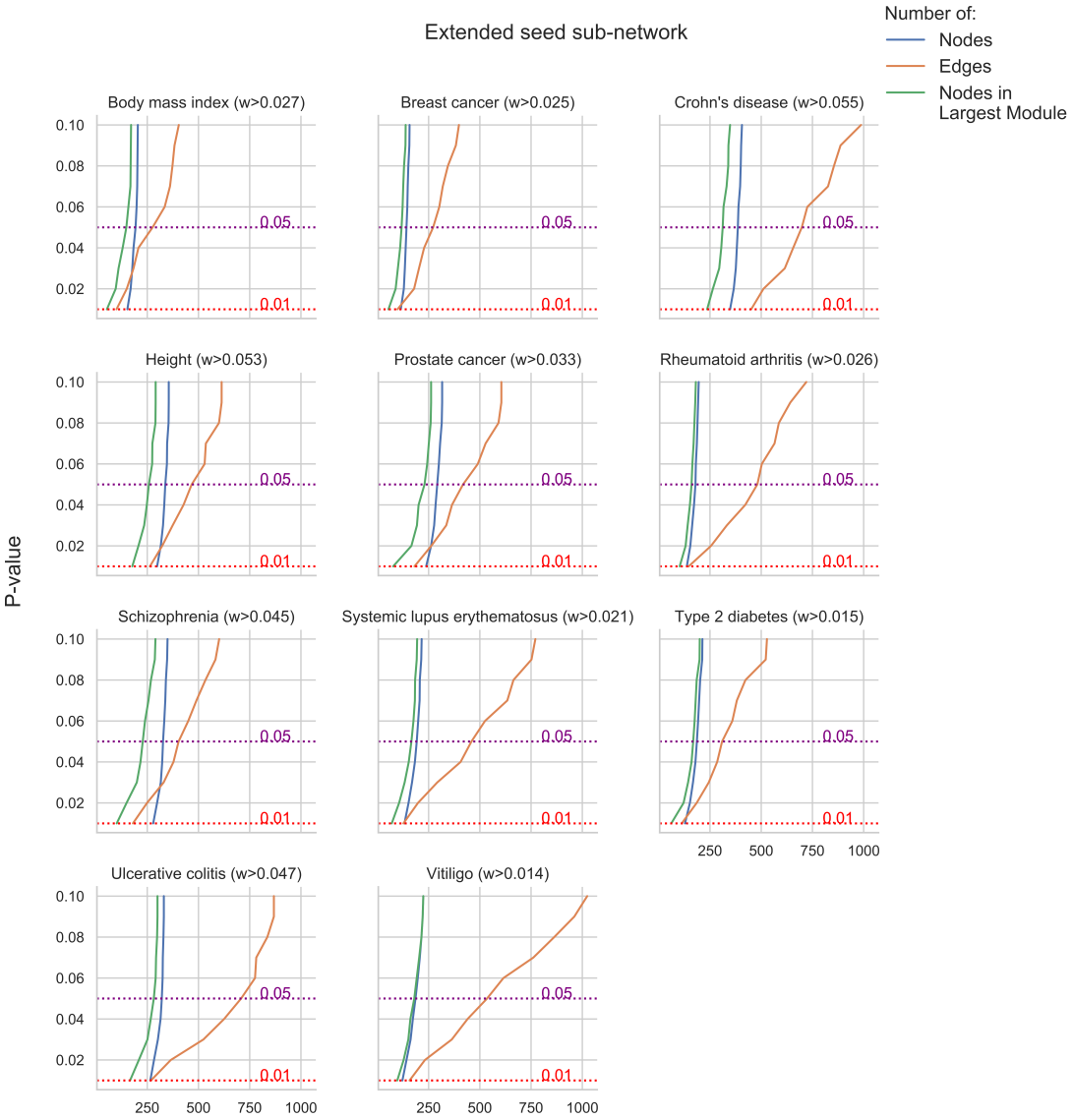


Figure 6.8: Influence of P-value threshold: The effect of the P-value threshold was examined when applying NetCore to 11 GWAS gene sets. The effect was measured, in the extended seed sub-network, by the number of nodes (blue), number of edges (orange) and number of nodes in the largest module (green). Y-axis is the P-value, X-axis is the measured size. For each gene set the chosen weight threshold was fixed, and the P-values range between 0.01 and 0.1. Two significance levels are marked: 0.01 (red) and 0.05 (purple).

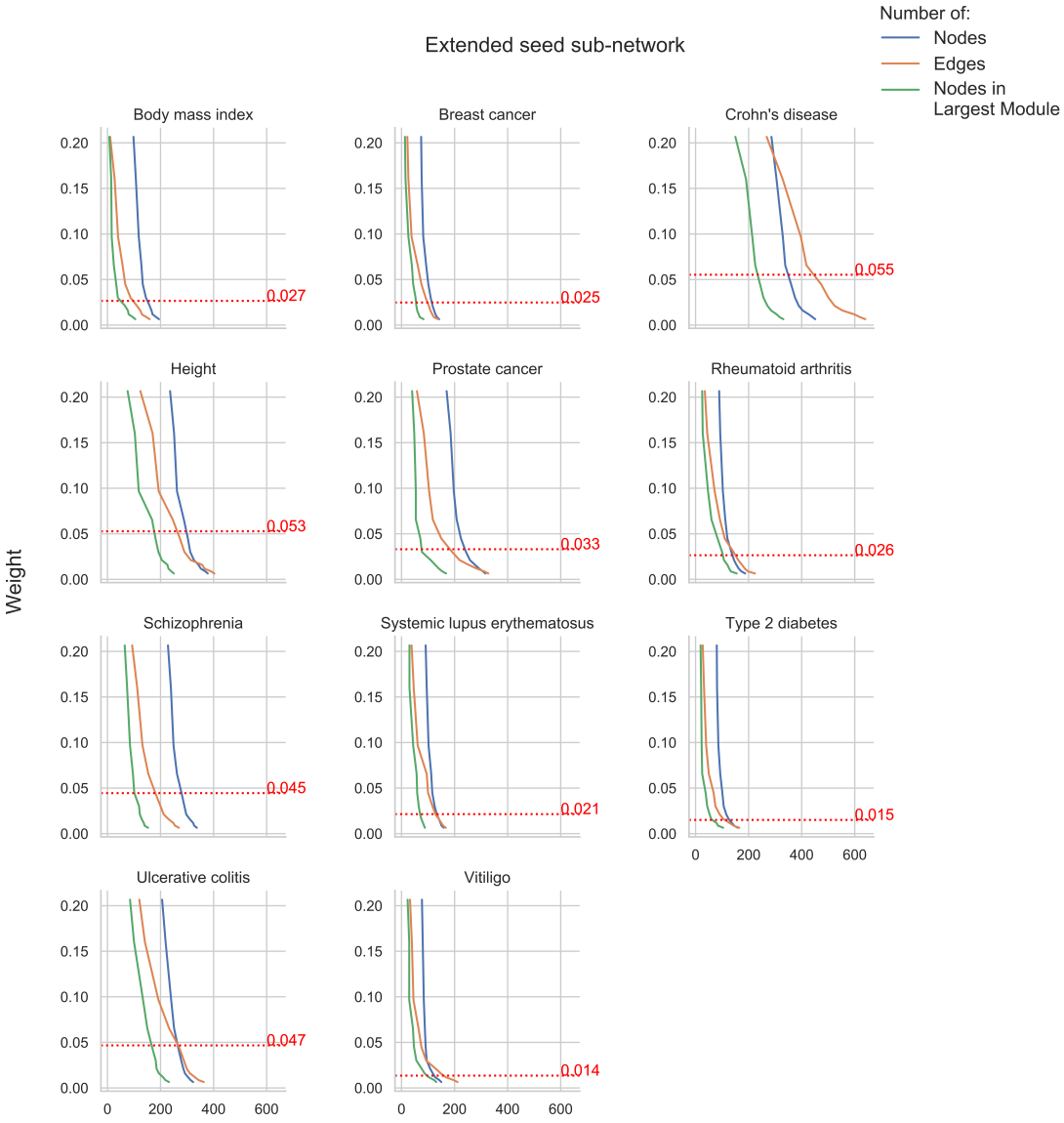


Figure 6.9: Influence of weight threshold: The effect of the weight threshold was examined when applying NetCore to 11 GWAS gene sets. The effect was measured, in the extended seed sub-network, by the number of nodes (blue), number of edges (orange) and number of nodes in the largest module (green). Y-axis is the weight after propagation, X-axis is the measured size. For each gene set the chosen weight threshold is marked (dashed red line), which was calculated by the 75th percentile of the weights after propagation, among the significant ($p < 0.01$) nodes which are not in the input seed list.

seed sub-networks. We note that for all gene sets, the majority of the nodes obtain only a very small weight after the propagation (all of these nodes had an initial weight of 0 before the propagation, as they are not included within the input seed gene sets). Thus, our aim was to set the threshold such that these nodes are excluded and so only nodes with higher weights at the end of the propagation are further considered.

To evaluate the influence of the weight threshold on the results we measured the change in the size of the modules for different weight values. Figure 6.9 shows the number of nodes, number of edges, and number of nodes in the largest module for weight values which vary between 0.006 and 0.2. For each gene set we calculated these sizes given 11 weight values, equivalent to the values of the 99th to the 50th percentiles of the weight distributions. The number of nodes increases from the lowest weight value to the highest one by between 47.8% and 110.1%, with most gene sets more than doubling in size. We note that the differences in sizes largely depend on the size of the GWAS set. For larger gene sets the sizes increase when the weight threshold is lower. However, for smaller gene sets the differences in sizes are less extreme. Indeed, all three sizes (see Figure A.7) are highly correlated with the number of seed genes (Pearson correlation coefficients of 0.99, 0.86 and 0.83 for the nodes, edges and largest module, respectively).

In conclusion, we generally find that setting the threshold at the 75th percentile allows us to generate reasonably sized modules which could still be biologically interpretable. Nevertheless, since the sizes of the modules depend both on the range of input weights as well as the number of seed nodes (in this case also the number nodes that are scored above 0), we recommend to closely examine the generated modules and adjust the weight threshold accordingly.

6.5 Summary

In this Chapter we evaluated NetCore's performance with the goal of identifying disease genes and disease modules. Since there are no "gold standards" for either problem, we compiled 11 sets of (disease) genes based on GWAS studies and tested NetCore's ability to predict those. We compared the degree- and core-based normalizations and concluded that the best performance is achieved with a core-based normalization. We further examined the influence of different components, including the version of the CPDB PPI network, the technique for generating network permutations, and all of NetCore's parameters. In general we found that the high confidence CPDB PPI network performs best, in particular when setting the restart parameter to a high value. Furthermore, we provided two approaches for evaluating the network modules generated by NetCore and applied them on the 11 GWAS gene sets, while particularly focusing on the functional relevance of the largest module for type 2 diabetes. To test the robustness of NetCore's modules to both the P-value and weight thresholds we studied the changes in their sizes, focusing on the number of novel nodes that were added to the modules. We observed that up to $p = 0.05$ the differences are rather minor and the modules are still small enough to be functionally interpretable. The weight threshold however can increase the sizes of the modules to a larger extend, mostly depending on the size of the gene set, and therefore must be carefully adjusted given the input.

The identification of modules is therefore a delicate process, and the parameters help control the balance between the number of new predictions and a reasonable size of the sub-networks.

7 Applications to Data and Results

The following chapter provides examples for the type of analysis that is available with NetCore, as well as comparisons to other network propagation-based methods. We demonstrate the application of NetCore on two different complex diseases, namely schizophrenia and cancer, using mutational data and public resources for the identification of disease modules. We compare NetCore's results with three other propagation-based methods and estimate its ability to produce disease-relevant predictions with respect to those. In addition, we provide another application of NetCore where we try to elucidate the toxic effects of drugs by using gene expression levels measured with RNA-seq from a 3D cardiac microtissue model and identifying drug toxicity response networks.

7.1 Comparison to other network propagation methods

Several methods which are based on network propagation have already been applied to genomics data for the purpose of identifying disease associated genes and disease modules. In Section 3.5.1 we reviewed some of the methods, and specifically those which apply the RWR formulation. Most of these methods utilize the standard degree normalization of the adjacency matrix, yet usually vary in the post-processing of the propagation weights and the identification of network modules. The caveats of the current methods, which served as the main motivation for the development of NetCore, were previously discussed in Section 3.6. To illustrate how NetCore's implementation resolves these caveats we compared it with three other methods: NAGA [53], HotNet2 [166] and Hierarchical HotNet [242]. While NAGA is a network propagation method specifically designed for the analysis of GWAS data, both HotNet2 and Hierarchical HotNet were developed for cancer mutation data. In the next Sections we briefly review the technical details of the methods, focusing on the main differences to NetCore. Then we provide a comparison between the results of NetCore and the three methods, using suitable data sets. In order to allow fair comparisons, all other network propagation methods were applied using the CPDB PPI network, with a restart parameter of $\alpha = 0.8$, as was previously selected for the application of NetCore.

7.1.1 NAGA method

NAGA is a network propagation method tailored for the analysis of results from GWAS studies [53]. The workflow includes three main parts. In the first part, the GWAS results are summarized into P-values which are converted to gene scores. In the second,

the genes are re-ranked using network propagation with an interaction network from the Network Data Exchange (NDEx) database [234]. Finally, a set of prioritized genes are extracted to be evaluated against a gold standard list and to create an associated sub-network. The network propagation scheme in NAGA is based on the standard RWR formulation, with degree normalization of the adjacency matrix. After the propagation, the authors recommend re-ranking the genes and taking the top 100 genes as the prioritized list. Then, a sub-network which includes these 100 genes can be constructed, and network modules can be computed using `ModuLand` [282], a network clustering method implemented in `Cytoscape` [267], a popular network analysis software. NAGA does not apply any significance test, neither for the re-ranking step nor the module identification one. It was evaluated on nine GWAS data sets, using three different PPI networks, and compared with two other network-based methods designed for GWAS data analysis, which are not based on network propagation. Comparing NetCore with NAGA allows us to demonstrate both the advantages of core normalization as well as the incorporation of prior knowledge for the module identification procedure.

7.1.2 HotNet methods

HotNet2 [166] and Hierarchical HotNet [242] were both developed for the identification of significantly mutated sub-networks based on pan-cancer mutation data. They are also both based on the same network propagation scheme that is using the RWR formulation, with degree normalization of the adjacency matrix, yet apply a different approach for extracting network modules. Both methods define a similarity matrix S using the RWR formulation and the input weights. S is computed by representing the input weights as a diagonal matrix (instead of a vector) in the RWR steady-state equation (see Section 4.2.2). HotNet2 then builds a fully connected graph from S, removes edges below a minimum threshold δ, and extracts the strongly connected components (SCCs) from the graph, which then serve as the final modules. The output consists of four values for δ, which are estimated from the data and the results. We compared NetCore only with HotNet2 results which were extracted using the minimal δ value, because this value yielded the largest final modules. Hierarchical HotNet constructs a hierarchy of clusters from S consisting of SCCs, estimates the optimal cut for the hierarchy, and reports back the generated clusters as modules. Of these modules, we only compared NetCore to those that consisted of at least two genes. Both methods apply a statistical test which is based on input randomization of the weights or the network. HotNet2 is an extension of the previous HotNet algorithm [295], which was based on an undirected heat diffusion process. It was extensively evaluated on three different interaction networks, and was compared with HotNet as well as two other standard pathway-based enrichment methods: `DAVID` [131] and `GSEA` [280]. The performance of Hierarchical HotNet was also evaluated on three different interaction networks, and compared with HotNet2 as well as three other network-based methods which aim to identify mutated sub-networks in cancer: `heinz` [78], `MUFFINN` [60] and `NetSig` [129]. HotNet2 is considered the state-of-the art method for identifying cancer modules, while Hierarchical HotNet demonstrated the best performance for predicting candidate can-

Source	Disease	Gene Set Size	Coverage in PPI network
DisGeNET [231]	Schizophrenia	1436	1072
Pardiñas et al. [219]	Schizophrenia	945	426
NCG [240]	Cancer Consensus	711	626
	Cancer Candidate	1661	1050
DisGeNET [231]	Cardiomyopathy	773	602
KEGG [146]	Cardiomyopathy	115	106

Table 7.1: Gene sets which were used as prior knowledge for the identification of NetCore modules, or for the evaluation of the results. A schizophrenia gene list was downloaded from the DisGeNET database [231] and used for module identification. The cancer genes were taken from the Network of Cancer Genes (NCG) web resource [240]. The cancer consensus were used for identifying the modules, whereas the cancer candidates were used for evaluating the results. For evaluating the results of anthracycline drug-toxicity data we used the genes which are associated with cardiomyopathy in the DisGeNET database [231]. For identifying anthracycline drug-toxicity modules with NetCore we used three cardiomyopathy disease pathways from KEGG [146].

cer genes. Therefore, comparing NetCore to both allows for a comprehensive assessment of its results for cancer mutation data.

7.2 Schizophrenia GWAS

We applied NetCore to a schizophrenia genetic variations data set by the Schizophrenia Psychiatric GWAS Consortium [243], which was also used by Carlin et al. to apply NAGA [53]. The data set includes P-values for SNP associations to schizophrenia according to a GWAS study, which was based on the analysis of 9,394 cases and 12,462 controls. Based on the SNPs, genes were assigned with P-values, according to a predefined genomic region of 10 kilobases (kb) up- and downstream of the gene. Each gene is assigned the lowest P-value from the SNPs that are within its defined region. This P-value is then $-\log_{10}$ transformed such that each gene is associated with one weight. A total of 14,966 genes had a weight above 0, but only 9,033 of them were covered in the CPDB PPI network, and could be used as input.

For validating the results and later identifying schizophrenia modules with NetCore we extracted a list of genes that are associated with schizophrenia according to the DisGeNET database [231]. We downloaded the *BeFree* gene-disease associations (which are obtained by text mining of abstracts [45, 46]) from the database and extracted all the gene names relevant for schizophrenia. This list included 1,436 genes, with 1,072 of them covered in the CPDB PPI network (Table 7.1). In addition, we evaluated NetCore's performance for identifying schizophrenia-associated genes using another much larger GWAS data set by Pardiñas et al. [219] (with 40,675 cases and 64,643 controls), which was produced later than the one by the Schizophrenia Psychiatric GWAS Consortium. We downloaded the meta-analysis summary statistics and extracted all the available

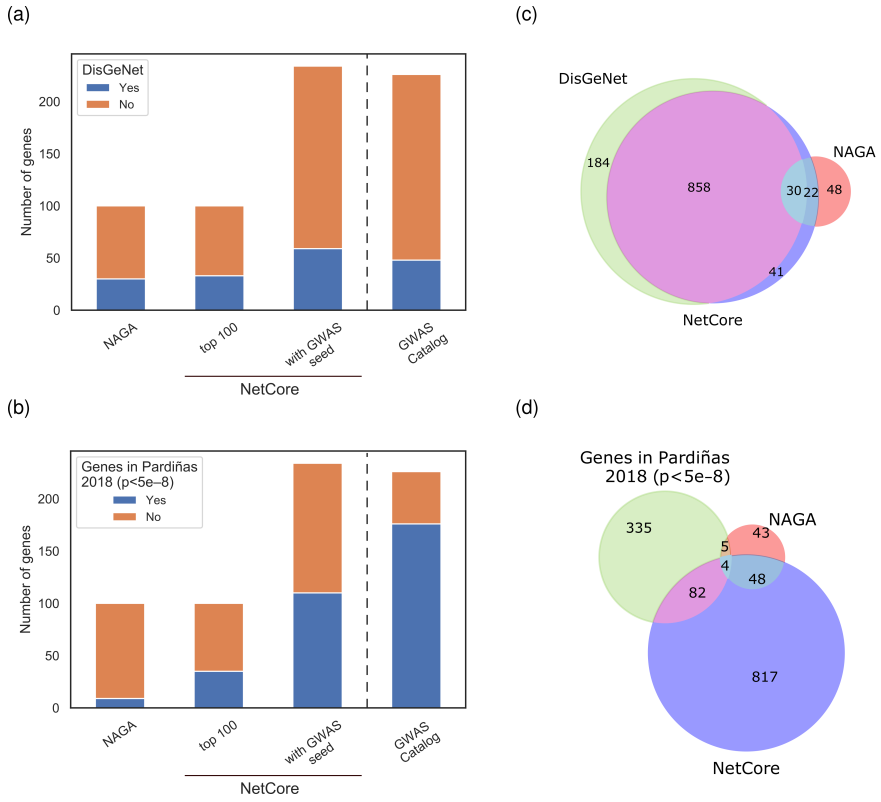


Figure 7.1: Schizophrenia genes in network modules: (a) The number of genes, and whether they are contained in the DisGeNET schizophrenia disease gene list (blue) or not (orange), for the following (left to right): top 100 genes computed with NAGA, top 100 genes computed with NetCore; NetCore predicted genes with the 221 GWAS catalog derived genes as seed list. The last bar shows the overlap of the 221 GWAS catalog genes with the DisGeNET genes. (b) The same analysis results as in (a) but with the overlap computed with the genes that were significant ($p < 5e-8$) in a recent Schizophrenia GWAS study by Pardiñas et al. [219]. (c) Venn diagram for: DisGeNET schizophrenia genes, NAGA's top 100 genes and NetCore's largest module when using DisGeNET as the seed list. (d) The same analysis results as in (c) but with the overlap computed with the significant genes from Pardiñas et al.

SNP to P-values associations. The SNPs were then associated with genes, according to a predefined genomic region of 10kb, up- and downstream of it. Each gene was assigned the lowest P-value from the SNPs that were identified for its region. We applied a significance level of $p < 5e-8$, which was also used by Pardiñas et al. [219], to identify significant SNPs. 945 significant genes remained, and 426 of them were covered in the PPI network and used for the evaluation of NetCore (Table 7.1).

We applied NetCore such that core normalization was used together with the computed gene weights that are based on the GWAS P-values. In addition, we also applied NAGA to the same input weights and the CPDB PPI network. Then, as suggested for NAGA, we extracted the top 100 genes according to the re-ranking after propagation. To allow direct comparison of the methods, we then computed the re-ranking after propagation from NetCore, without applying any module identification, and also extracted the top 100 genes. As both NetCore and NAGA apply the RWR procedure, and since the same input was used, the only difference in this case is due to the core nor-

malization in NetCore, versus the standard degree normalization in NAGA. 63 of the 100 genes were reported by both methods, i.e. each method identified 37 genes that the other did not.

In order to highlight the advantages of including prior knowledge for the module identification step, we also applied NetCore using 226 genes from the GWAS catalog (Table 6.1) as seed nodes, computed an extended seed sub-network from these genes along with the propagation results, and extracted the largest module. We evaluated the results by calculating the overlap to schizophrenia-associated genes derived from the DisGeNET database [231]. Although the DisGeNET gene list is not independent of the list from the GWAS catalog, this comparison demonstrates the power of incorporating prior knowledge in NetCore. The performance (measured by overlap with DisGeNET) improved when the genes from the GWAS catalog were used as seed nodes. Figure 7.1(a) shows the overlap between the different gene lists and the 1,072 schizophrenia-associated genes in DisGeNET. NAGA's top 100 genes have an overlap of 30, while NetCore's top 100 genes overlap with 33, an increase of 10% in comparison to NAGA. NAGA does not apply any module identification steps, and therefore all of its reported top 100 genes will be included in the extracted sub-network. With NetCore, however, we can apply a module identification step, which will in fact include different genes from the top 100 that we used for the comparison with NAGA. The list of genes generated using the 226 GWAS catalog seed nodes increases the overlap to the DisGeNET list further to a total of 59 genes. The original GWAS seed set of 226 genes includes only 48 genes from DisGeNET, i.e. when using NetCore with seed genes from GWAS, we added 11 more disease-associated genes from DisGeNET and thus improved the consistency of the two data sets. The higher number of non-DisGeNET genes in this case is due to the high number of non-DisGeNET genes already within the GWAS seed set.

In order to further test consistency of network propagation among disease gene sets, we evaluated the different network propagation outcomes with a second set of genes that were found to be significant (GWAS $P-value < 5e-8$) in a larger and more recent schizophrenia GWAS study [219]. Figure 7.1(b) displays the performance as measured by overlap to the 426 significant genes from this study. Whereas NAGA's top 100 genes overlap with only nine of those significant genes, NetCore's top 100 genes overlap with 35 of them, i.e. more than 3 times larger. Hence, NetCore is more robust with respect to different validation sets of the same disease.

In addition, we further compared the enriched pathways for the gene sets that were used in Figure 7.1(a). We extracted the top 20 most enriched pathways for the disease genes from the GWAS catalog, and examined the level of enrichment for those pathways in the other gene sets, which were identified by network propagation. The results are displayed in Figure 7.2. Although NAGA's enrichment is higher for some pathways, overall NetCore is enriched for more pathways, with a stronger enrichment than for those genes which are only in the GWAS catalog list. The incorporation of prior knowledge is demonstrated here again by the higher enrichment of NetCore with GWAS seed in comparison to NetCore's top 100 only.

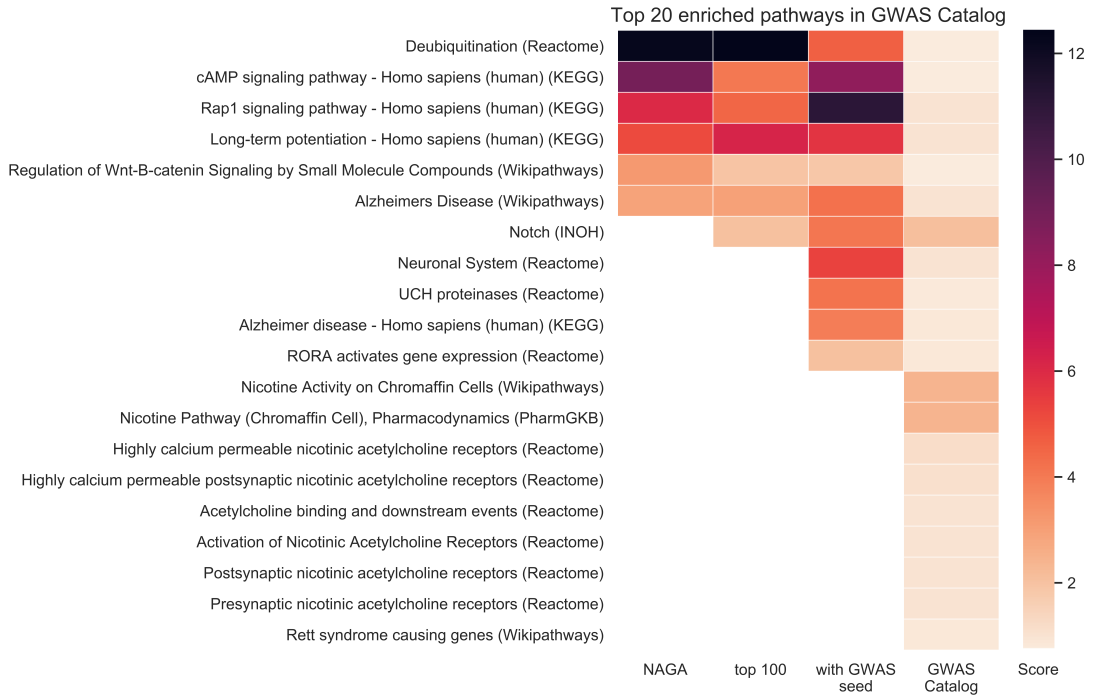


Figure 7.2: Enriched pathways for schizophrenia modules: Over-representation analysis for the schizophrenia results from: NAGA top 100, NetCore's top 100, NetCore with GWAS seed, and genes from the GWAS catalog. Displayed are the 20 most enriched pathways for the genes in the GWAS catalog, and their enrichment in the other gene sets. The color indicates the enrichment score (measured by $-\log_{10}$ of the enrichment Q-value).

As the incorporation of prior knowledge improved NetCore's results, we applied NetCore once more, this time using the genes from DisGeNET as seed genes, since this is the most comprehensive gene list (1,072 genes). With these seed genes NetCore identified 1,136 genes in the extended seed sub-network, where the largest module consisted of 951 genes. Of these, 888 genes are shared with DisGeNET, which is to be expected, as they were used as seed genes (Figure 7.1(c)). Furthermore, the overlap with the significant genes from Pardiñas et al. increased to 86 (Figure 7.1(d)). 63 genes are potential novel disease genes, 22 of them were also among the top 100 genes predicted by NAGA. Figure 7.3 lists the 63 genes in NetCore that are not in DisGeNET, grouped according to whether they were also predicted by NAGA, and ranked according to their weight after propagation with NetCore. We noticed that the genes with the highest weights after the propagation, for example *TRAF6* and *SRC*, are also predicted by NAGA. This is to be expected, as NAGA takes the top 100 ranked genes after propagation. However, there are also some genes with intermediate weights, which are only predicted by NetCore, such as *BTN2A1* and *AP2M1*. *BTN2A1* had an initial propagation score of 6.3 and was also found to be significant ($p = 4.9e-40$) in the more recent schizophrenia GWAS study [219], in addition to seven more genes that were predicted by NetCore (marked in blue in Figure 7.3).

We further explored two non-DisGeNET genes, which are among NetCore's novel predictions. *SRC*, which is the second highest prediction, and is also predicted by NAGA,

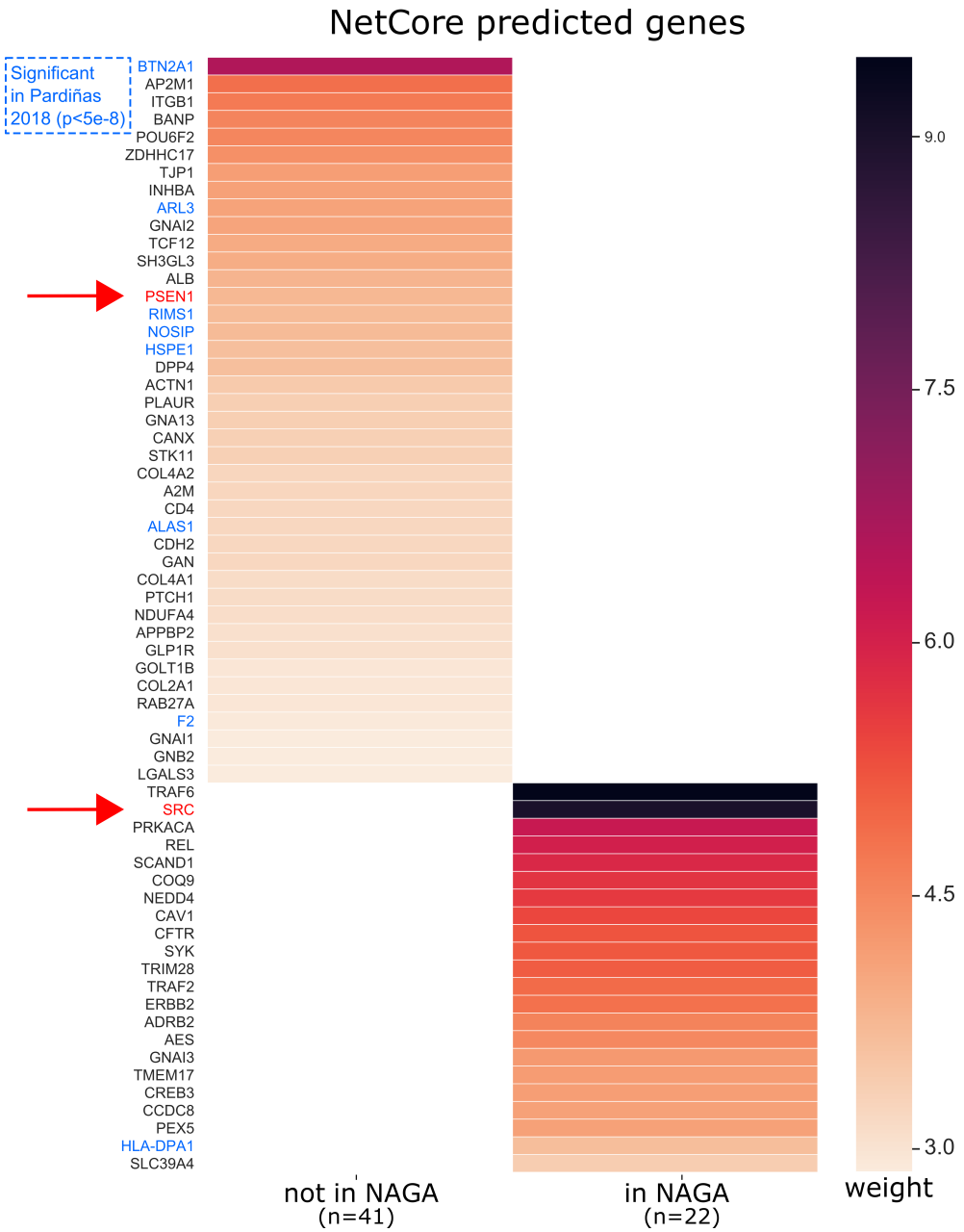


Figure 7.3: Novel schizophrenia genes predicted by NetCore: 63 genes predicted from NetCore that are not contained in the DisGeNET list, according to whether or not they were also predicted by NAGA. The blue text indicates whether a gene was found significant in the GWAS study by Pardiñas et al. [219]. In red are highlighted two of the genes which were further explored, and their neighborhoods are visualized in Figure 7.4.

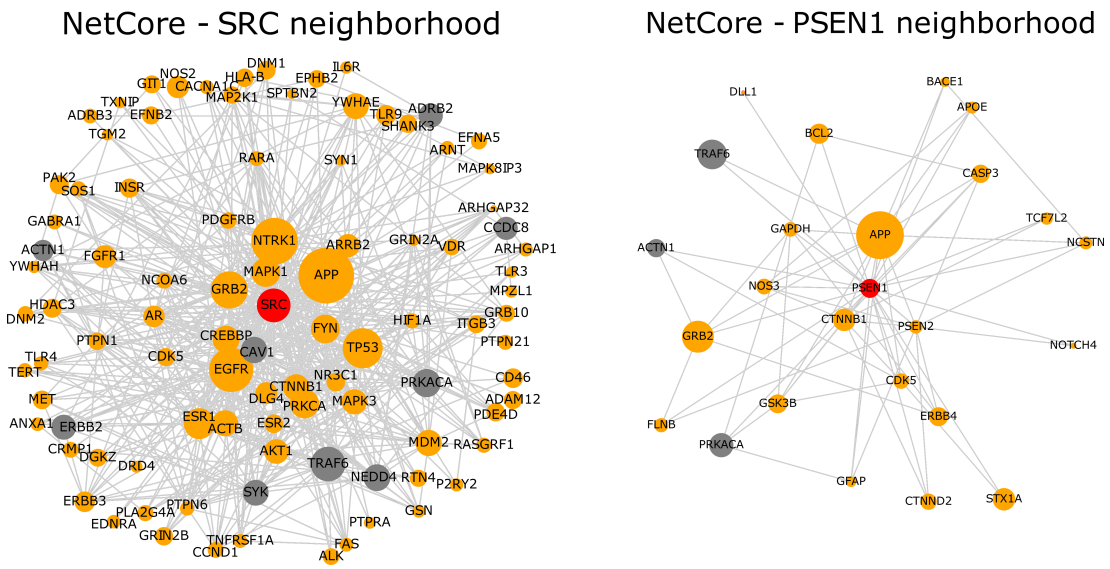


Figure 7.4: The neighborhoods of *SRC* and *PSEN1* in NetCore's largest module. Orange genes are in the DisGeNET list, gray are novel predictions. The sizes of the nodes are proportional to the weights after propagation.

as well as *PSEN1*, which was predicted by NetCore only. Both nodes had initially rather low weights (1.3 for *SRC* and 1.7 for *PSEN1*) which were greatly increased at the end of the propagation (9.0 for *SRC* and 3.7 for *PSEN1*). Their neighborhoods within the largest module in NetCore are visualized in Figure 7.4. *SRC* has 97 neighbors, 87 of them from DisGeNET (orange) and 9 are novel predictions (gray). The sub-network is fairly dense, with 548 interactions, 97 alone belong to *SRC*. Other highly connected genes in the sub-network are: *EGFR*, *GRB2*, and *ESR1*, all of which are already associated with schizophrenia according to DisGeNET. Among the newly predicted genes are *TRAF6* and *PRKACA*, which are ranked first and third respectively from the non-DisGeNET genes, and also appear in the predictions made by NAGA. Both of these genes had a rather small initial weight based on the data (1.8 for *TRAF6* and 0.9 for *PRKACA*), and a substantial increase in their weight after the propagation (9.5 for *TRAF6* and 6.2 for *PRKACA*). From the nine predicted neighbors of *SRC*, only *ACTN1* is not predicted by NAGA, and has a rather low weight after the propagation, yet was still found to be significant by NetCore. These three genes, *TRAF6*, *PRKACA* and *ACTN1* are also in the neighborhood of another gene that was only predicted by NetCore, *PSEN1*. *PSEN1*, is not directly connected to *SRC*, but is connected to other well-known disease genes too, such as *APP* and *GRB2*. The neighborhood of *PSEN1* is smaller, with 21 neighbors and 68 interactions in total. All of the neighbors, apart from *TRAF6*, *PRKACA* and *ACTN1*, are already associated with schizophrenia according to DisGeNET.

In conclusion, NetCore is able to provide gene predictions which are indeed relevant to Schizophrenia. For example, *BTN2A1* would be a promising candidate for further studies as it has already been associated with other disorders such as dyslipidemia [94, 120, 128]. *SRC* is a tyrosine-protein kinase, which is connected to many genes that are known to be associated with schizophrenia, for instance *MAPK1* and *NTRK1*, among others. Dysregulated *SRC* has been previously linked to schizophrenia together with

the activity of *NMDA* [24, 233, 252, 310]. *GRIN2A* and *GRIN2B*, which are some of the sub-units of *NMDA*, are also connected to *SRC* in NetCore's module. *PSEN1* is also connected to other well-known disease genes, but is not a neighbor of *SRC*, and thus might be involved in a different mechanism that affects the disease. *PSEN1* encodes a presenilin protein, which is associated with other neurodegenerative diseases, in particular Alzheimer's disease. In fact, mutations in *PSEN1* have been identified as one of the first genes related to early-onset Alzheimer [268]. Furthermore, *PSEN1* has also been implicated in other neurodegenerative and neuropsychiatric disorders [17].

7.3 Pan-cancer mutations

We applied NetCore to a pan-cancer somatic mutation data set, provided by Lawrence et al. [161], that was previously applied to both HotNet2 and Hierarchical HotNet. The data set consists of primarily whole-exome sequences from tumor–normal pairs, with a total of 4,742 samples from 21 tumor types, both from TCGA projects and from non-TCGA projects at the Broad Institute. The mutations from all samples were combined together, such that duplicated patients and duplicated mutations were removed. For each tumor type a total of 18,388 genes were analyzed, and three significance metrics were calculated using the following methods: MutSigCV [162], MutSigCL and MutSigFN [169]. MutSigCV calculates for a gene the number of non-silent mutations, and determines its significance according to a background model, which is based on the number of silent mutations in the surroundings of the gene. MutSigCL measures the significance of the positional clustering of the observed mutations, while MutSigFN measures the evolutionary conservation in the positions of the mutations. Both measures are assigned a P-value based on a permutation test of the non-silent coding mutations. Finally, the significance levels from the three metrics were combined into a single P-value, which was then corrected for multiple testing into a single FDR Q-value using the Benjamini-Hochberg FDR procedure [35]. This resulted in 1,489 genes with $Q-value < 1$, 929 of them covered in the PPI network. We transformed the Q-values using the $-\log_{10}$ and used those as node weights to initialize network propagation with NetCore.

For evaluating the results, we made use of the NCG catalog [240]. The catalog contains manually curated information from publications about more than 2,000 cancer-associated genes, which are known or predicted to have driver roles in cancer based on somatic mutations. These genes are divided into two categories: (1) 711 cancer consensus genes, which include both tumor suppressors and oncogenes, and (2) 1,661 cancer candidate genes, which were identified by mutational screenings and have strong support to be involved in cancer development. We used the cancer consensus gene list as seed nodes for identifying network modules in NetCore, and evaluated the results using the cancer candidate gene list (Table 7.1).

For comparison with NetCore, we also applied the same data, using the CPDB PPI network, to both HotNet2 and Hierarchical HotNet. We extracted the modules from the three methods and compared them based on their genes. HotNet2 reported in total 59 genes in two modules of sizes 57 and 2. Hierarchical HotNet reported in total 35

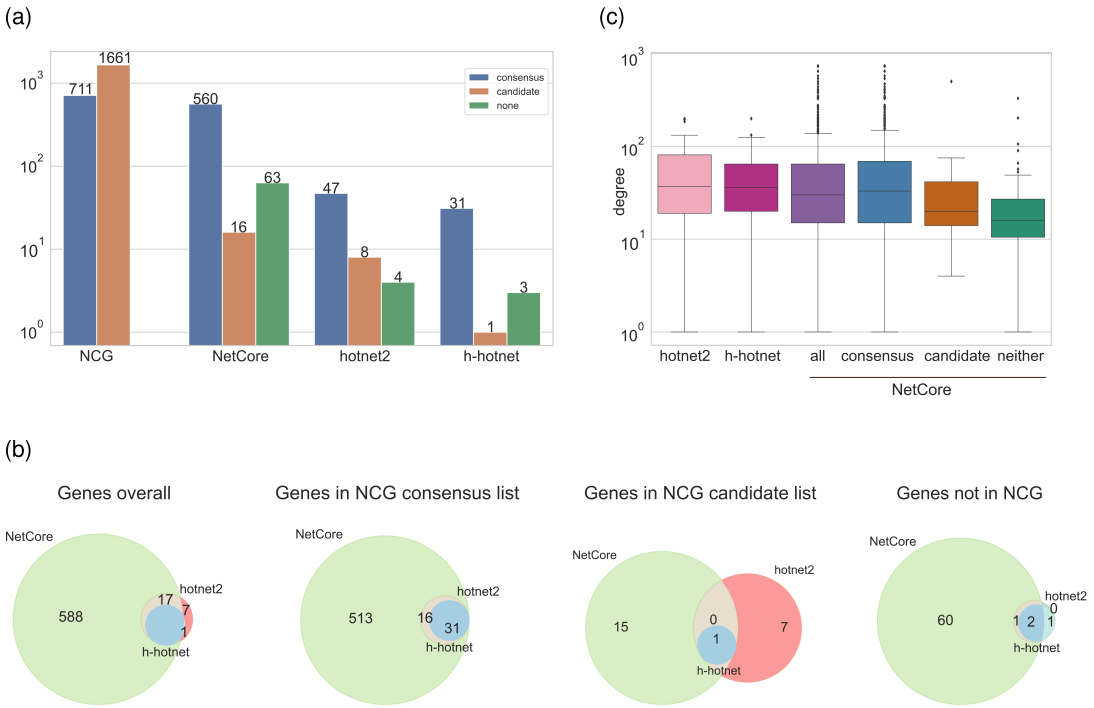


Figure 7.5: Pan-cancer mutation results: (a) The number of genes in the predicted modules reported by NetCore, HotNet2 and Hierarchical HotNet (h-hotnet). Colors indicate the different categories of the NCG lists (blue = cancer consensus genes, orange = cancer candidate genes, green = genes in neither of the lists). (b) Venn diagrams showing the overlap between the three methods for all the genes in the computed modules, and according to the different NCG categories. (C) Box plots of the node degrees of the computed module genes. For NetCore this is also broken down to the different NCG categories.

genes in four modules of sizes 26, 4, 3 and 2. NetCore reported in total 639 genes in three modules of sizes 633, 4 and 2. Figure 7.5(a) shows the number of genes in all the modules, and their overlap with the genes from the NCG cancer consensus and candidate lists. Both Hotnet2 and Hierarchical HotNet have a relatively small number of genes in their modules, however most of them belong to the consensus list. NetCore includes most of the cancer consensus genes, as those were used for the module identification step, however, it also retrieves the most candidate genes. This is further shown in Figure 7.5(b), where NetCore's genes almost completely overlap with the other two methods, but include 15 more genes from the candidate list. HotNet2 reported seven genes from the candidate list, yet those are not overlapping with the 15 genes reported by NetCore. Figure 7.5(c) illustrates the degree of the nodes in all the modules. We observed high degrees for the genes which are in both the HotNet2 and Hierarchical HotNet modules. This is due to HotNet2 and Hierarchical HotNet mainly reporting genes from the consensus list, and hardly reporting any genes with lower degrees. We divided the genes from NetCore's module according to the NCG lists, and noticed that both the genes from the candidate list as well as the genes that are in neither of the lists have lower degrees.

We further evaluated the cancer-association potential of the genes that have been identified by NetCore and that are neither consensus nor candidate genes. We ap-

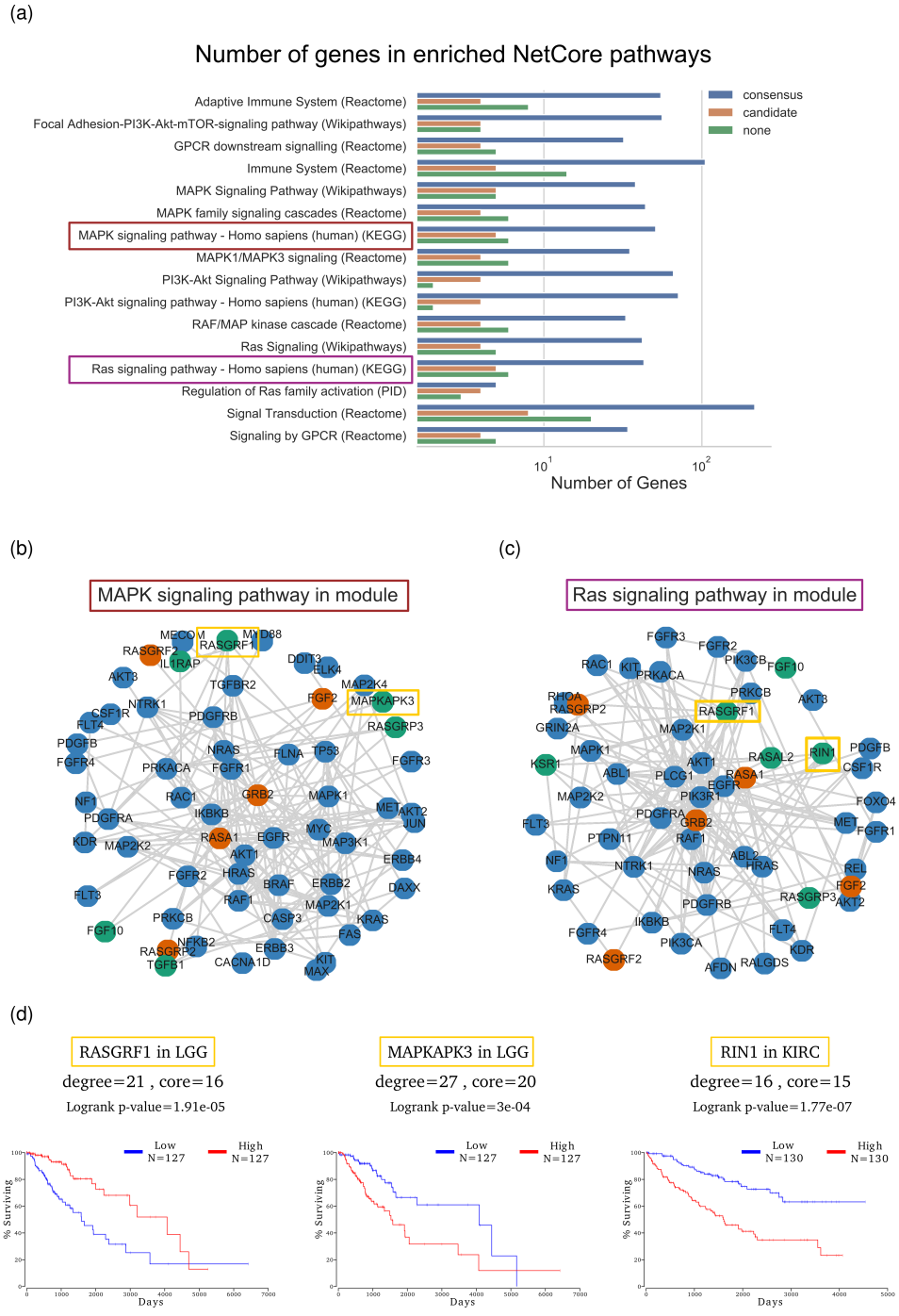


Figure 7.6: Signaling pathways in pan-cancer modules from NetCore: (a) Pathways with at least three predicted genes from the candidate list in the largest module (633 genes) from NetCore. Color indicates the different NCG categories. (b) The genes from the largest module that are part of the "MAPK signaling pathway". (c) The genes from the largest module that are part of the "RAS signaling pathway". Blue genes are in the consensus cancer list, orange in the candidate and green are new predictions. (d) Cox regression plots for three of the novel predicted genes from the modules in (b) and (c), marked with a yellow square, generated using `OncoLnc` [13]. *LGG* refers to Brain Lower Grade Glioma. *KIRC* refers to Kidney renal clear cell carcinoma.

plied an over-representation analysis to the 633 genes in NetCore's largest module and extracted the most enriched pathways ($Q-value < 0.01$). We then investigated those enriched pathways that contained at least three genes from the candidate list ($Q-value < 2.7e-6$).

Figure 7.6(a) lists the identified 16 pathways and the number of genes in NetCore's largest module, according to the NCG lists. All pathways also include genes that are neither consensus nor candidate genes, where some pathways include more genes from the candidate list, and others do not. We focused on two KEGG pathways: "MAPK signaling pathway" ($Q-value = 1.11e-19$) and "Ras signaling pathway" ($Q-value = 2.05e-20$), which are both well known to be involved in cancer development (see for example [300]).

Figure 7.6(b)-(c) display the sub-networks from NetCore's largest module that include the genes which are enriched for these pathways. The majority of the genes are in the consensus list (blue), while a smaller number of genes are in the candidate list (orange) or in neither of the two lists (green). In total, both pathways include nine genes that are in neither of the NCG lists: *RASGRP3*, *FGF10* and *RASGRF1* are present in both pathways; *RASAL2*, *KSR1* and *RIN1* in "Ras signaling pathway" only; and *IL1RAP*, *MAPKAPK3*, and *TGFB1* in "MAPK signaling pathway" only. All of these genes, apart from *RASGRP3*, did not have an initial weight, i.e. they were not found to be significantly mutated, yet they were identified as significant by NetCore and had a high enough weight to be included in the final modules.

We argued that these novel genes might still be cancer-relevant biomarkers since they are connected with many consensus and candidate genes. We thus examined the associations of these genes to cancer survival data by generating Cox regression plots using `OncoLnc` [13]. With this tool one can explore patient survival correlations to gene expression data for 21 cancer types from TCGA. Therefore, we utilized this tool as an independent resource for estimating the relevance of the predicted genes to cancer development and progression. And so, we applied it to the previously identified nine genes, which were not listed as cancer consensus or candidate genes, and extracted the results for the cancer types with the lowest FDR-corrected P-values. In Figure 7.6(d) we show the results for three of the novel predicted genes from the modules, which were part of the MAPK and Ras signaling pathways: *RASGRF1*, *MAPKAPK3* and *RIN1*. The expression levels of all three genes were significantly associated with survival of cancer patients, and therefore could potentially be used as biomarkers.

In addition, we repeated the same analysis for genes that are part of the overview "Pathways in Cancer" list from KEGG [146], which were predicted in NetCore's largest module but are in neither of the NCG lists. This is the most enriched pathway in the module ($Q-value = 1.34e-65$). It consists of 475 genes, which are also in the PPI network, 147 of them are in NetCore's largest module. From those, 137 are from the consensus list, three from the candidate list, and seven are in neither of the lists. We identified six of the seven novel genes as potential biomarkers due to their significant correlations to patient survival data: *CTBP1*, *FGF10*, *LPAR1*, *LRP5*, *RASGRF3* and *TGFB1*. *FGF10* and *TGFB1* were also identified as part of the "MAPK signaling pathway". All of these genes had an initial weight of 0 at the start of the propagation, yet were identified as significant with a high enough weight after the propagation by NetCore. Figure 7.7

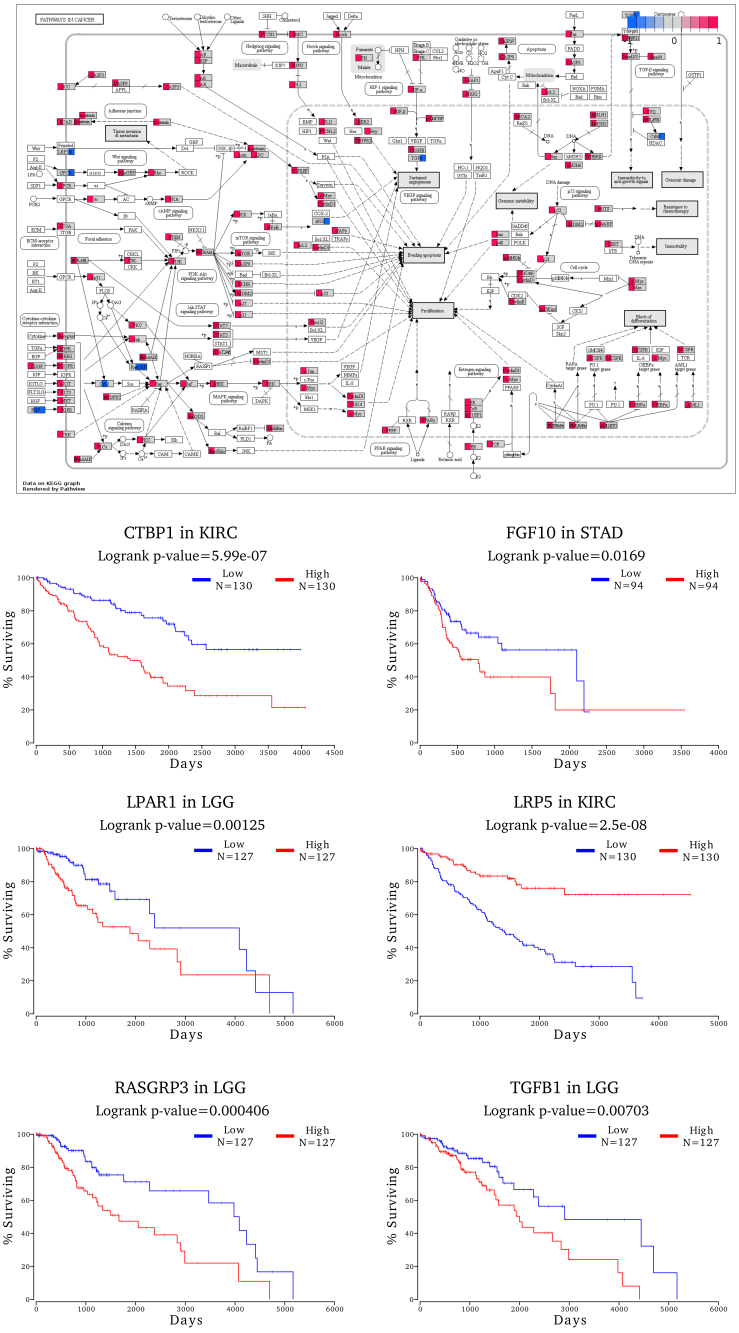


Figure 7.7: Pathways in Cancer: The pathway as depicted by KEGG and generated using `Pathview` [173]. The colored nodes are present in the module from NetCore. Red nodes are present in the NCG cancer consensus list. Blue nodes are newly predicted genes, some are present in the NCG cancer candidate list, and some are not. The Cox regression plots are based on TCGA survival data for six genes in the pathway that were predicted by the module and are not present in either the consensus or candidate cancer lists. The plots were generated using the `OncoLnc` [13] tool. The results are shown for the cancer type with the lowest FDR-corrected P-value. *LGG* refers to Brain Lower Grade Glioma. *KIRC* refers to Kidney renal clear cell carcinoma. *STAD* refers to Stomach adenocarcinoma.

illustrates the genes from the pathway, marking the genes which are also present in NetCore's largest module, and the correlations of the six genes to patient survival data.

In conclusion, NetCore identified several genes that can potentially serve as biomarkers for cancer. Some of the genes have already been suggested to be involved in the disease. For example, over-expression of *CTBP1* promotes the progression of multiple cancer types, as recently reviewed by Blevins et al. [39]. *RIN1* expression has been implicated in tumor development and invasion [116, 264, 308]. Hypermethylation of *RASGRF1* has been suggested as a biomarker for colorectal cancer [56]. Finally, *MAPKAPK3* was also suggested as a potential biomarker for colorectal cancer [28], however further investigations are still required to confirm this observation.

7.4 Toxicogenomics - drug toxicity expression levels

In this Section we aim to demonstrate the application of NetCore to drug toxicity studies, and in particular to toxicogenomic studies. Elucidating and understanding the mechanisms that drive drug toxicity are crucial for preventing the toxic effect and eventually improving drug development processes [114]. Following the advancements of sequencing-based technologies, the concept of toxicogenomics was developed in order to study the toxic effects of drugs by genomic analysis [190]. It has previously been shown that gene expression signatures can be used to predict toxicity levels, i.e. identifying differentially expressed genes (DEGs) and associating them with toxic phenotypes [14, 127, 222]. Accordingly, NetCore can be applied to toxicogenomics data in order to provide network modules which suggest pathways and mechanisms that might be involved in causing the toxic phenotype. To that end, gene expression measurements upon drug treatments must be converted into scores, which can then be used as input weights for the propagation.

We have previously constructed a workflow for the analysis of toxicogenomics data which included a scoring scheme for the application of network propagation [29]. We showed that the enrichment level of the identified network modules was increased in comparison to the enrichment of the significant DEGs only, and therefore amplified the functional information which is relevant to toxicity. On top of that, the modules extended the mutual effects between different drugs, as more genes were shared between the detected modules in comparison to the DEGs only. Here we describe the scoring scheme that was used for the workflow, and adapt it for gene expression levels measured by RNA-seq, as it was originally designed for microarray data.

Our previously developed workflow for the analysis of toxicogenomics data in the context of pathway and network analysis aims at identifying the mechanisms that lead to drug toxicity [29]. In short, the workflow, illustrated in Figure 7.8, allowed us to analyze gene expression levels from drug toxicity studies together with network and pathway information from CPDB [145]. The pathway analysis was previously established in-house by Hardt et al. [113], which was made available via the ToxDB web interface[1],

1 http://toxdb.molgen.mpg.de/

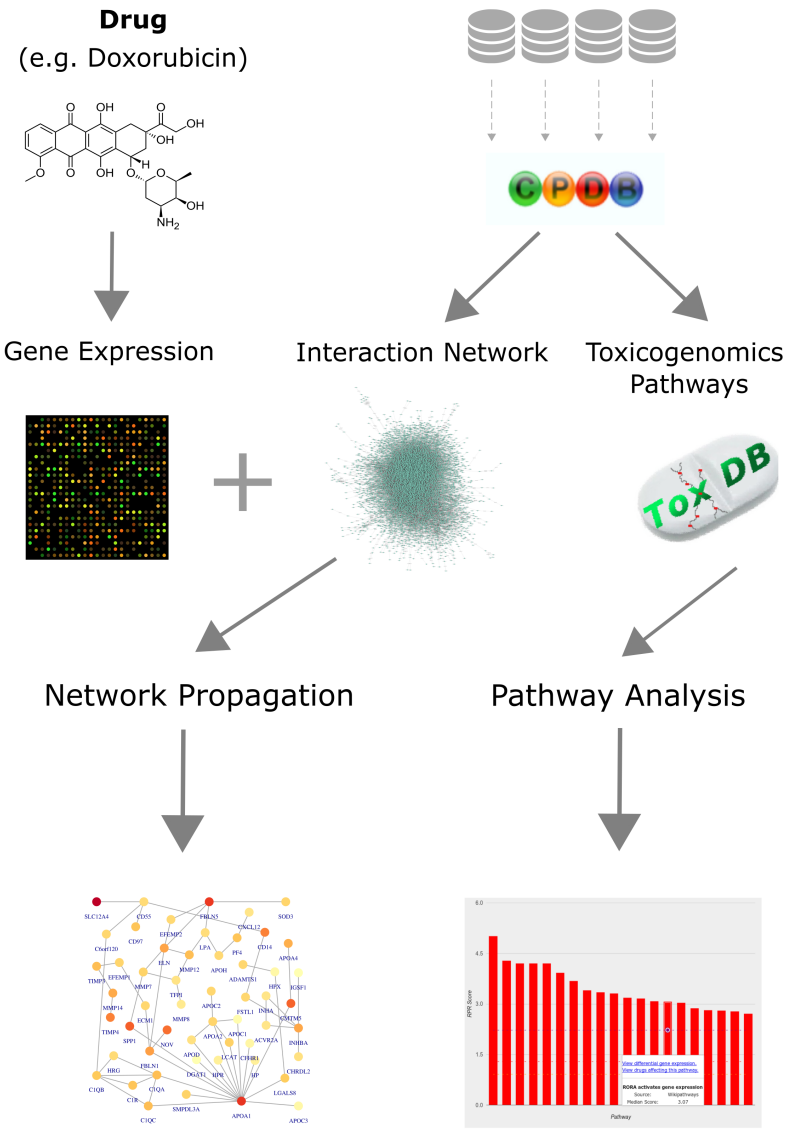


Figure 7.8: Network and pathway analysis of toxicogenomics data: Workflow for analyzing toxicogenomics data at the network and pathway level. Gene expression data is collected upon drug treatment and used in combination with pathway and network information from CPDB. The pathway analysis can be executed via ToxDB [113] and the network analysis is applied using network propagation. The workflow allows for the identification of both relevant pathways, that are disrupted upon drug treatment, and network response modules, that hold functionally relevant genes and novel candidates. Taken from [29].

and provided drug-toxicity response pathways. The network analysis was executed by us using the HotNet2 [166] network propagation algorithm (see Section 7.1.2), and produced functionally relevant network toxicity modules. The workflow was designed for the analysis of gene expression levels derived from microarrays, however could easily be adapted to RNA-seq. The main requirement is for the control and drug treatments to be compared so that DEGs can be computed. This way, each gene is associated with a FC which is assigned a significance level, given the null hypothesis that there is no change in expression between treatment and control. The FC levels are either positive (up-regulated) or negative (down-regulated). However, to execute network propagation it is requires to have positive scores only, and thus we computed new scores, which are positive only, and reflect the degree of change in comparison with the control. In addition, we also wanted the scores to include the significance level, in order to focus the analysis mostly on those genes whose expression is significantly changed (regardless if increased or decreased).

For toxicogenomics data, where different time and dose conditions are available for each drug, we proposed the following scheme for scoring the genes. For every gene i, drug j, and time-dose treatment k, the score for gene i under the conditions j and k is defined by:

$$S_{ijk} = |\log_2 r_{ijk}||\log_{10} p_{ijk}| \tag{7.1}$$

where r_{ijk} is the FC and p_{ijk} is the P-value from the differential expression analysis. This score describes a weighted FC of the gene, such that the more significant the change is, the higher the weight is. It also measures how much the gene is affected by the treatment, regardless of the change in expression, i.e. if the gene is up- or down-regulated. In addition, the score can be calculated for all genes, and therefore there is no need to assign a cut-off to the significance level.

The workflow was previously applied to rat *in vivo* data from DrugMatrix [96] and was focused on cardiotoxic compounds, and specifically on four compounds from the anthracycline family: daunorubicin (DAU), doxorubicin (DOX), idarubicin (IDA), and epirubicin (EPI). Anthracyclines are widely used in cancer chemotherapies and have been shown to be extremely effective despite the fact that they induce cardiotoxicities in up to 23% of the patients [167, 275]. Anthracycline-induced cardiotoxicity can result in cardiomyopathy and heart failure, in many cases only after a long period of time post-treatment [100]. Anthracyclines can interact in the cell with several components, as depicted in Figure 7.9. Although it is known that anthracyclines disrupt the synthesis of DNA and RNA [100], and that they lead to mitochondrial dysfunction [189], the mechanisms that cause the cardiotoxic effects still remain largely unclear [291]. Previous studies have tried to elucidate this problem, however, there is still need for further investigations so that detection and prevention can be improved [239].

Very recently, in a collaborative work by us and others [263], the same four anthracycline compounds were studied in iPSC-derived human 3D cardiac microtissues with the goal of identifying adverse mechanisms of cardiotoxicity. The experimental design and analysis workflow that were applied in this study are illustrated in Figure 7.10. Over a period of 14 days the cell models were challenged with the four anthracycline

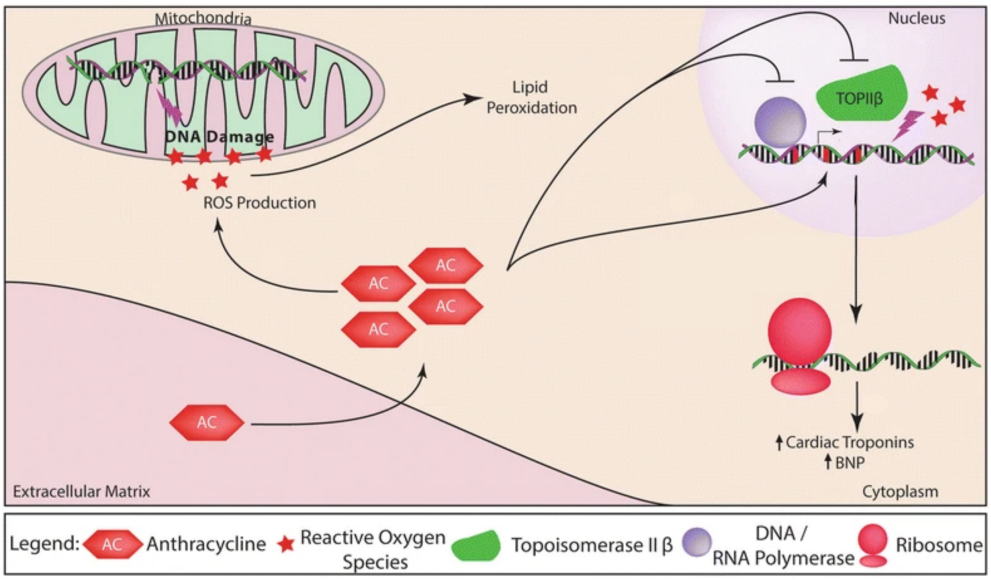


Figure 7.9: Anthracycline induced cardiotoxicity: Anthracyclines (AC) that enter the cell impair DNA and can cause mitochondrial damage by inducing reactive oxygen species (ROS). AC inhibit DNA and RNA synthesis and also inhibit the enzyme Topoisomerase IIβ (TOPIIβ), which leads to activation of DNA-damage response that leads to cell death. AC can also function as transcriptional inhibitors, i.e. affect gene transcription, which leads to cardiac injury that can be detected by increased levels of cardiac troponins and brain natriuretic peptides (BNP). Adapted from [195].

compounds, which were dissolved in dimethyl sulfoxide (DMSO) at two physiologically relevant doses (therapeutic and toxic) [157], and measured at seven time points (2h, 8h, 24h, 72h, 168h, 240h, 336h), collecting three replicates for each one. The effects were measured for dynamic quantitative proteomics (LC-MS), transcriptomics (RNA-seq) and methylation (MeDIP-seq). The measurements were compared with control profiles which were derived from time-matched DMSO-treated microtissues. The results from the longitudinal expression analyses were used to identify a network module which represented a common signature of the effects of all four compounds. The results from this integrated *in vitro* approach were also shown to be clinically transferable to cardiac biopsies taken from patients. This integrated approach was established as an efficient method to capture dynamic drug responses across time and dose in an *in vitro* modelling system that enables to bypass animal-based testing, which have been previously shown to translate poorly to human conditions [114].

For the purpose of this work, and in order to identify clinically relevant drug toxicity modules upon anthracycline treatments, we used the gene expression levels which were measured in the iPSC-derived human 3D cardiac microtissues mentioned above. In contrast to the data that was previously analyzed by us [29], these gene expression measurements are more relevant as they were derived from a human *in vitro* model, and in addition they were measured using RNA-seq, which is more advanced than microarrays. The raw RNA-seq data were previously processed by Selevsek et al. [263] using the Genedata Profiler® software (v.11.0.). Transcripts and genes were identified with an algorithm based on `Cufflinks` [290] and the differential expression analysis was performed with `DESeq2` [171]. Thus, for every drug-dose-time experiment the FC

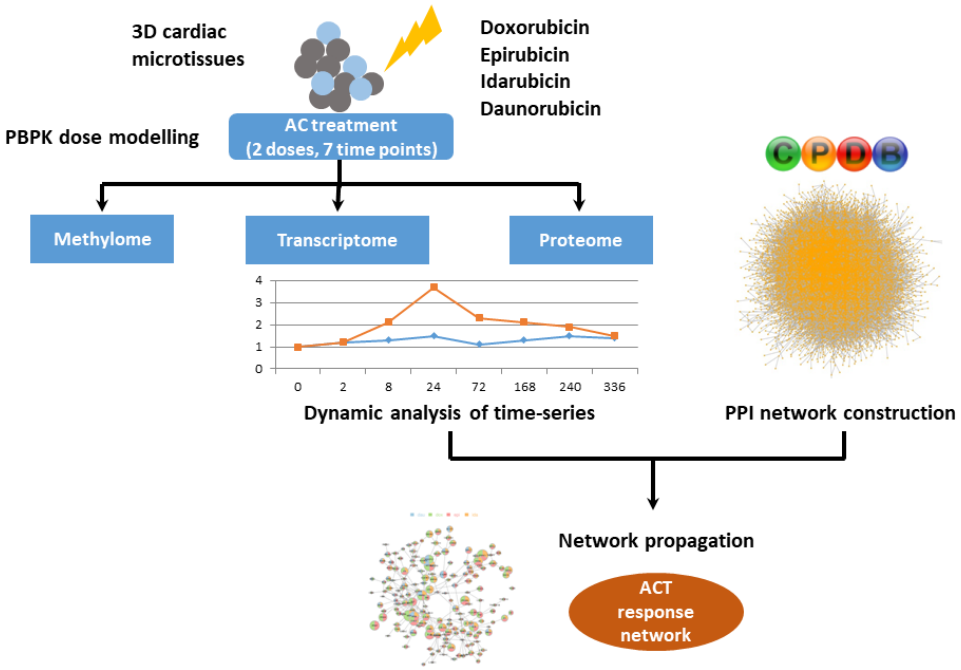


Figure 7.10: Experimental design for measuring anthracycline induced cardiotoxicity in iPSC-derived human 3D cardiac microtissues: 3D cardiac microtissues were grown for 14 days and treated at two doses calculated by physiologically based pharmacokinetic (PBPK) modeling with four anthracycline drugs: DAU, DOX, EPI and IDA. Measurements of the proteome, transcriptome and methylome were taken at 7 different time points. Proteome and transcriptome measurements were used to characterize dynamic cellular responses. Time series data was mapped to the CPDB PPI network in order to identify a common response network across all treatments. Adapted from [263].

and significance level were extracted. Based on those we computed the gene scores, in the same way as described in Equation 7.1. In order to apply NetCore we needed to generate one score for every gene. Hence, we combined the scores from all the seven time points by taking their mean, while separating the therapeutic scores from the toxic scores, i.e. applying NetCore for each compound twice (Table 7.2). In this way the effect over time is summarized, yet the effect of the dose remains separated. In addition we also applied HotNet2 to the same data in order to compare the results with NetCore.

To evaluate the results produced by NetCore and compare them with those from HotNet2 we downloaded disease-gene associations from the DisGeNET database [231] for Cardiomyopathy (Familial Idiopathic). This included a list of 773 disease genes, 602 of them were covered in the CPDB PPI network (Table 7.1). In addition, in order to apply module identification in NetCore using prior knowledge, we downloaded the genes which are associated with three KEGG [146] cardiomyopathy pathways: 1) dilated cardiomyopathy, 2) arrhythmogenic right ventricular cardiomyopathy (ARVC), and 3) hypertrophic cardiomyopathy (HCM). The pathways are comprised of 89, 72 and 83 genes respectively, with a total of 115 genes combined, and 106 of those were covered in the CPDB PPI network (Table 7.1). The overlap between the cardiomyopathy genes from DisGeNET and those from the three KEGG pathways is 41.

Drug	Dose	Input gene scores - all	Input gene scores - significant only
DAU	The	6,892	132
DAU	Tox	7,659	40
DOX	The	8,224	179
DOX	Tox	8,254	115
EPI	The	8,064	341
EPI	Tox	8,518	157
IDA	The	8,756	434
IDA	Tox	8,678	158

Table 7.2: The number of genes which are covered in the CPDB PPI network and were used as input for running NetCore for each drug-dose combination. The scores were computed once for all genes, and once only for genes that were significantly differentially expressed in all measured time points.

For each drug-dose combination we extracted the modules from both NetCore and HotNet2. Both methods can produce more than one module for each condition (see Section 5.6 and Section 7.1.2), and therefore we compared their results based on the genes in the largest reported module only, and on the genes in all of the modules combined. Figure 7.11(a) displays the number of genes in the largest module for each condition, and whether those were included in the cardiomyopathy gene set from DisGeNET. For all conditions, the number of genes in the HotNet2 modules is larger than the number of genes in the NetCore modules. Although the overlap with the DisGeNET genes is mostly higher in the modules from HotNet2, the overlap of the modules from NetCore is only slightly lower in most conditions, with an ever higher overlap for DAU (therapeutic) and DOX (toxic), despite having smaller modules. Moreover, when comparing the genes in the largest modules in all of the conditions together (Figure 7.11(b)) we note that the overlap of NetCore with the DisGeNET genes (22%) is twice as high as that of HotNet2 (11%), relative to the total number of the reported genes in each method.

When comparing the genes from all modules (Figure 7.11(c)) we note that HotNet2 produced a much larger number of modules and genes, and therefore the overlap in each condition with the DisGeNET genes is higher than that of the NetCore genes. However, the overlap of the genes from all conditions in the HotNet2 modules is reduced to less than 8%, whereas in the NetCore modules it remains above 20% (Figure 7.11(d)). Indeed, NetCore's genes are interconnected in fewer modules, as depicted in Figure 7.11(e). The largest modules from NetCore consist of more than 70% of the genes from all modules, whereas HotNet2's genes are dispersed in many modules, with the largest modules consisting of less than 40% of the genes, in all conditions. This is due to the fact that HotNet2 produced many modules of small sizes, with an average module size of four genes only.

To evaluate the robustness of NetCore towards the size of the input (i.e. number of genes with a weight above 0) we applied a more stringent scoring scheme. Instead of

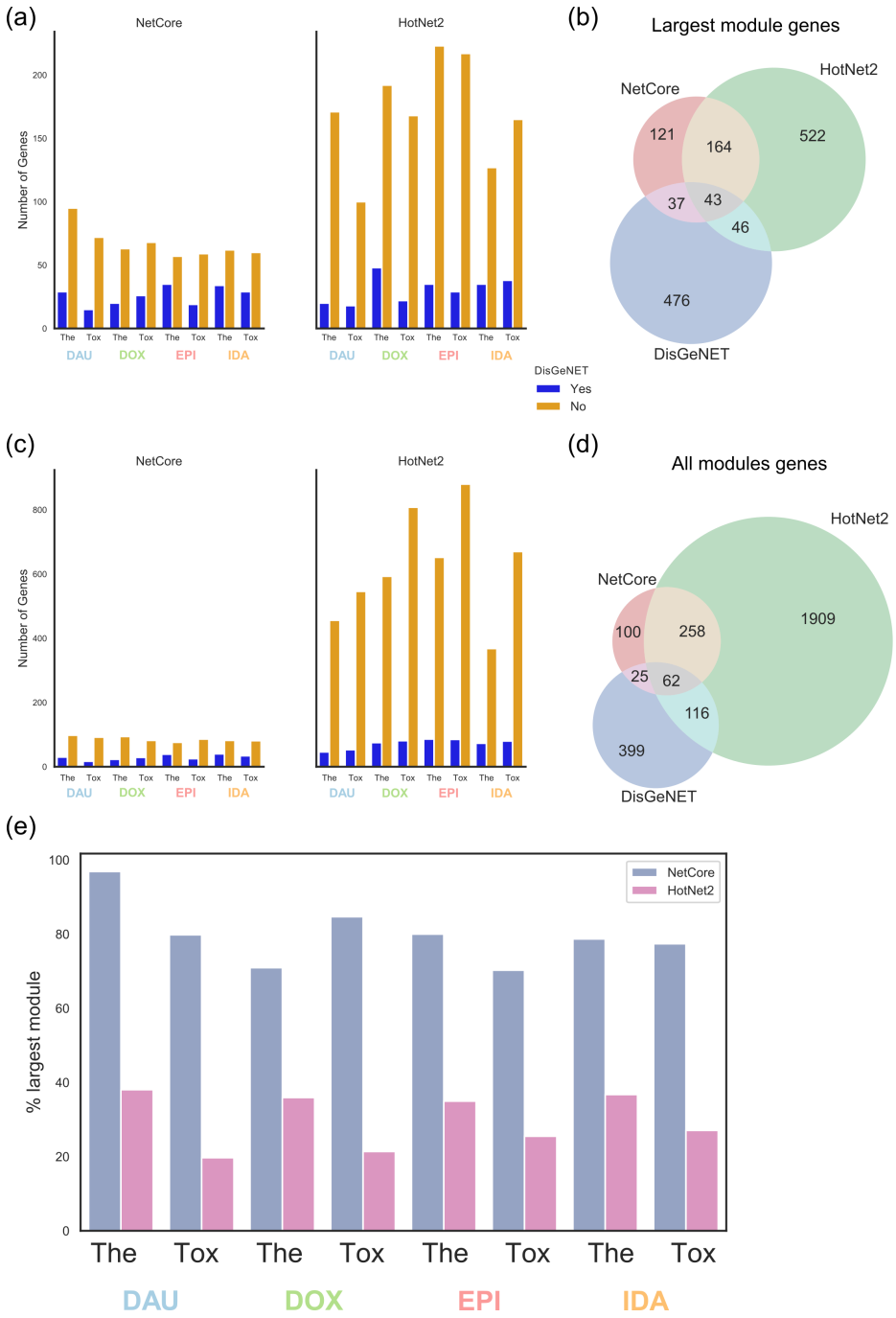


Figure 7.11: Comparison of modules generated for anthracycline toxicity measured by gene expression: The modules were computed using NetCore and HotNet2. (a) The number of genes in the largest module for every drug, in both therapeutic (The) and toxic (Tox) doses, and the overlap with the cardiomyopathy genes from the DisGeNET database [231]. (b) The genes in all of the largest modules (for all drug-dose conditions) from NetCore and HotNet2, and their overlap with the cardiomyopathy genes from DisGeNET. (c) and (d) display the same results as in (a) and (b) respectively, but for all the genes from all the modules (each method reported more than one module for each drug-dose condition). (e) The number of genes in the largest module, relative to the total number of genes in all modules, for each drug-dose combination.

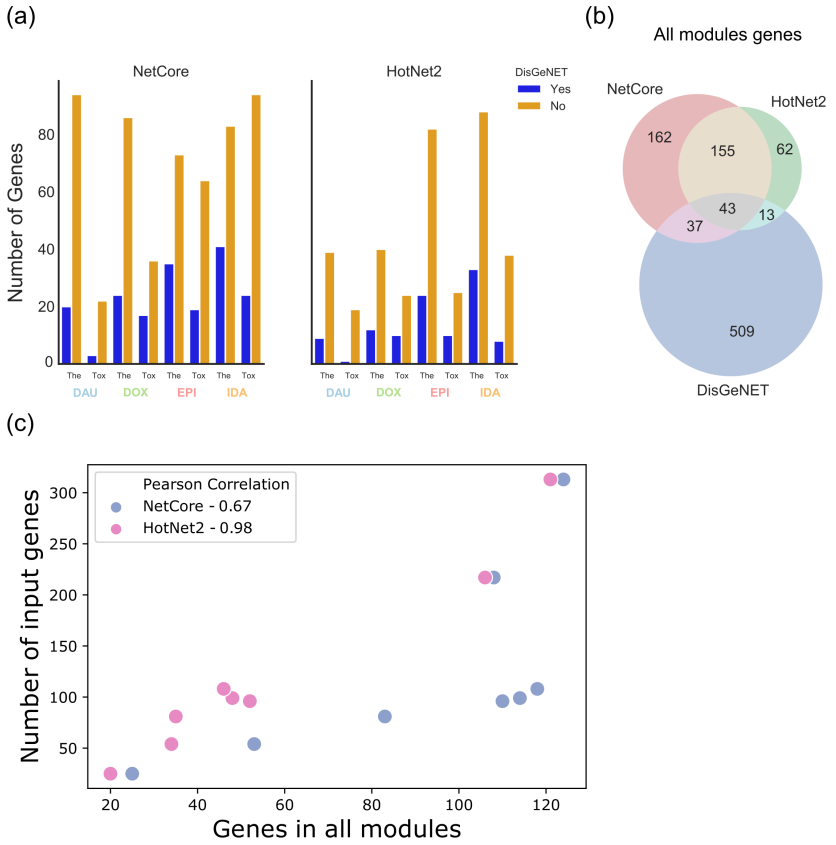


Figure 7.12: Comparison of anthracycline toxicity modules generated using a stringent scoring scheme: The anthracycline toxicity modules were generated once again, using a stringent scoring scheme such that the number of input nodes was reduced. (a) The number of genes in the largest module for every drug, in both therapeutic (The) and toxic (Tox) doses, and the overlap with the cardiomyopathy genes from the DisGeNET database [231]. (b) The genes in all of the largest modules (for all drug-dose conditions) from NetCore and HotNet2, and their overlap with the cardiomyopathy genes from DisGeNET. (c) The number of genes in all of the modules versus the number of input genes (with a score above 0), for every drug-dose combination.

computing a score for every gene, we calculated the mean over the scores from all seven time points only if the significance level (P-value) in each one of the time points was under 0.1. The aim is to consider genes with a non-significant FC as irrelevant, and therefore exclude them from the input to NetCore. For the anthracycline expression data, this resulted in a sharp decrease in the number of input genes which had a score above 0 (Table 7.2). Figure 7.12(a) displays the number of genes, from all modules, in each condition, for both NetCore and HotNet2. Using these scores both methods produced smaller modules than before (Figure 7.11(c)), however, NetCore reported more genes than HotNet2 in all conditions. The overlap of all the genes with the genes from the DisGeNET set, as displayed in Figure 7.12(b), is the same for both methods (20% with respect to the total number of reported genes), suggesting that NetCore is still as accurate as HotNet2, despite reporting more genes in total. Furthermore, HotNet2 is more sensitive to the size of the input, as it is highly correlated with the size of its output (Figure 7.12(c)), whereas NetCore is more robust to the input size.

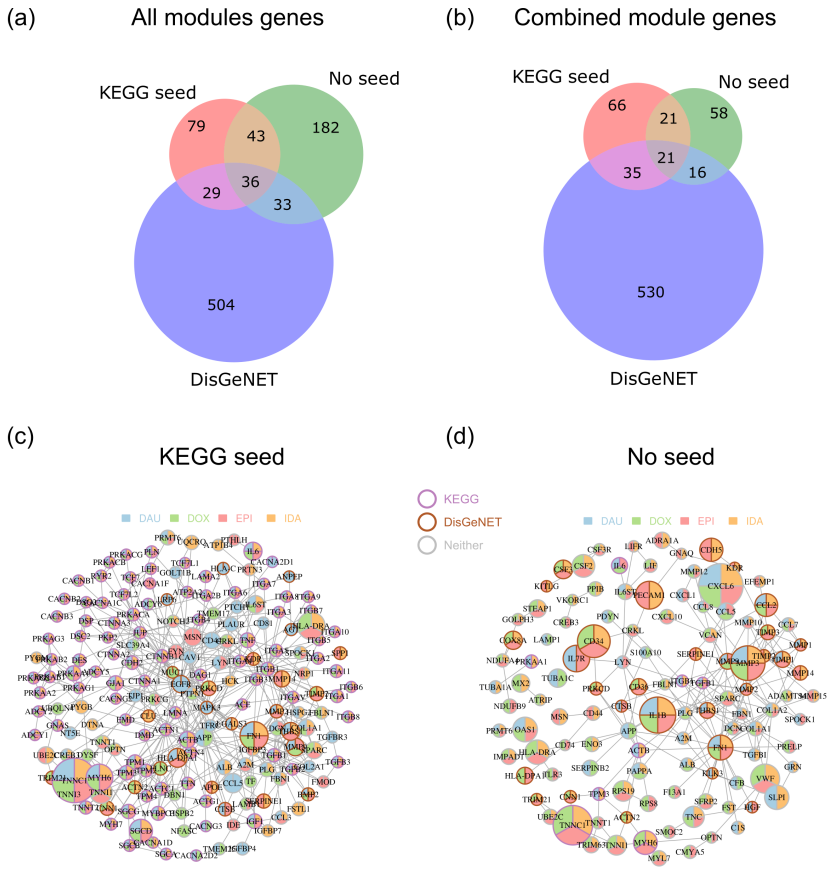


Figure 7.13: Anthracycline toxicity modules for therapeutic doses identified with NetCore using KEGG seed genes: (a) The genes for all four drugs from NetCore modules identified with and without KEGG seed genes. (b) The genes in the combined module for all four drugs, where only genes that were present in at least two of the four drug modules were extracted. The combined module for all four drugs when using the KEGG seed genes (c) and when not (d).

To demonstrate the benefits of NetCore's semi-supervised module identification procedure we applied NetCore once more, using the scores for all genes, and in addition used the 106 genes from the cardiomyopathy KEGG pathways (Table 7.1) as seed genes for the module identification procedure. Our previous analysis did not exploit any prior knowledge, and therefore the modules were generated by extending the sub-networks that included the 100 highest scoring genes (from the input). Figure 7.13 demonstrates the results for the therapeutic dose. We compared the genes in all of the modules, for all drugs, with and without the seed genes from KEGG, by computing their overlap with the cardiomyopathy genes from DisGeNET (Figure 7.13(a)). Although the total number of genes from all the modules without KEGG seed is larger, the overlap with the DisGeNET genes is higher for the KEGG seed modules (23% in comparison with 35%, respectively). This is to be expected, as there is already an overlap of 41 genes between the DisGeNET and KEGG genes. Of the 65 genes in the KEGG seed modules that overlap with the DisGeNET genes, 38 were already included by adding the KEGG genes as seed genes, i.e. NetCore identified 27 genes from the DisGeNET set which are

not in the KEGG set.

In order to produce a combined module for all four drugs, we extracted the genes that were present in at least one of the modules of two or more drugs. When comparing the genes from these combined modules (Figure 7.13(b)) we note that the number of genes is larger when using the KEGG seed genes. Hence, the commonality between the modules of the different drugs is increased when using the KEGG seed genes. Figure 7.13(c) visualises the combined modules when the KEGG seed genes were used, and Figure 7.13(d) when they were not. The number of edges in the KEGG seed combined module is also larger, and the nodes are more interconnected, as the majority of them are connected in one component, whereas in the combined module without the seed genes they are separated into smaller components. Both combined modules include sarcomere-related genes, such as *TNNC1*, which is also associated with cardiomyopathy according to both DisGeNET and KEGG, as well as genes which are neither in the KEGG set nor in the DisGeNET one, such as *TNNT1* and *TNNI1*, who are also from the same family of troponins. In addition, *TNNT2* and *TNNI3*, which are in the KEGG set, are indeed included in the KEGG seed module, yet not in the module without the KEGG seed. *TNNT2* was indeed found to be up-regulated in most time points, according to both transcriptomics and proteomics, in all drugs except for DAU [263].

In conclusion, we established that NetCore could easily by applied to gene expression data from drug toxicity studies. We showed that NetCore is able to produce relevant predictions regardless of the sizes of the input. On top of that, the modules identified by NetCore benefited from the incorporation of prior knowledge and allowed to increase the common toxic signature between all drugs. These modules also included genes from the cardiac troponins family, which have recently been implicated as targets of anthracyclines (see for example [1, 292]).

7.5 Summary

This Chapter included three examples for the application of NetCore to real data. The first two focused on disease phenotypes measured by mutational data, while the third focused on drug-toxicity effects measured by gene expression levels. Our aim was to provide a comprehensive description of the type of analysis which is available via NetCore, and demonstrate that it can be used for diverse types of problems and data. Furthermore, we underlined the benefits of NetCore by comparing it to previous network propagation-based methods. By applying NetCore to schizophrenia data we were able to show how core normalization improves over degree normalization, as well as identify novel candidate disease genes through applying our semi-supervised module identification approach. Using the same approach we were also able to identify candidate cancer genes, which participate in relevant cancer pathways, and could potentially serve as biomarkers. We also introduced NetCore as a tool for analyzing a different type of genotype-phenotype associations by applying it to gene expression levels that were measured upon anthracyline treatments. The various types of data, input and output that were applied to and generated by NetCore throughout this work are finally summarized in Table 7.3.

Disease / Phenotye	Input Data	Nodes with input weights	Seed genes resource	Nodes as seed genes	Nodes added by NetCore	Number of Modules	Largest module (% from extended seed sub-network)
Type 2 Diabetes	Binary	78	GWAS catalog [176]	78	39	15	45%
Schizophrenia	Mutations (P-values)	9,033	DisGeNET [231]	1,072	63	12	84%
Cancer	Mutations (P-values)	929	NCG [240] (consensus)	626	79	3	90%
Drug Toxicity (DOX The)	RNA-seq (weighted FC)	8,224	KEGG [146]	106	39	3	85%

Table 7.3: NetCore application summary: Gene sets that were applied in NetCore and were used together with prior knowledge from different resources. Type 2 diabetes is used as example for one of the 11 GWAS gene sets. DOX in therapeutic (The) dose is used as an example for one of the anthracycline drug-dose conditions.

8 Discussion

Diffusing information via network propagation is a well-established technique, which in recent years has been repeatedly adopted in the field of molecular biology [71]. The concept is relatively simple to implement and is therefore easily applied to any molecular interaction network and data. Although it has been successfully utilized to generate novel genotype-phenotype associations (mostly for complex diseases), there are still several limitations that hinder existing methods. In the scope of this work we identified and focused on two such limitations. The first arises from the frequent use of PPI networks, which suffer from study bias that directly affects the propagation results. The bias is reflected in high degree nodes, which are subsequently visited more often during the propagation process, and are thus more likely to be predicted as associated with the phenotype. The second limitation concerns the identification of network modules at the end of the propagation. To date, most methods provide modules which are poorly connected, with many of the genes scattered in distinct modules, making it challenging to determine the functionality of the modules.

To overcome these limitations, we developed NetCore, a network propagation method that uses node core instead of degree in the mathematical formulation of random walk with restart. We proposed alternative normalizations to the adjacency matrix that allow us to adjust the random walk probabilities. Thereby, the walk is expected to arrive less often at high degree nodes, and thus the study bias is addressed. We compared the performance of the normalizations in the task of identifying GWAS gene sets and concluded that core normalization is performing significantly better than degree. In addition, we presented a semi-supervised approach to identify network modules based on the propagation results. By incorporating prior knowledge we were able to identify functionally relevant modules, which improved connectivity between the genes and included novel predictions. We demonstrated the usability of the method on complex (disease) phenotypes and highlighted its benefits in comparison to other propagation-based methods.

For the purpose of this work we extracted a PPI network from CPDB. We included in this network only binary interactions from the database, and excluded complex ones. Therefore, the majority of these interactions were measured using the yeast two hybrid technique, and are therefore prone to both technical and selection biases. These biases result in star-like structures of 'bait' proteins, which will be connected to multiple 'pray' proteins.

In order to account for the fact that the study bias results in high degree nodes we explored other node metrics in the network and examined how they relate to degree. While clustering coefficient is a local metric, and betweenness centrality a more global one, both measures still reflect the study bias, albeit less than degree. Moreover, both measures have been identified as significantly higher in well-studied cancer genes

[240]. On the other hand, we observed that node core is more robust to the study bias, and that the difference between core and degree reflects the level of bias. Core is also a global metric, which can directly be associated with node degree [172], and therefore we found it to be more suitable for network propagation purposes.

This prompted us to develop three variations for the normalization of the adjacency matrix. The first is exclusively based on core. The second is based on the difference between core and degree, and the third on their ratio. Even though we showed that the difference indicates the level of study bias, when one uses it for the normalization of the adjacency matrix it can produce very low probabilities, especially in cases when the difference is very large. To avoid such low probabilities, we also provided a normalization based on the ratio between the core and the degree.

The normalization which is based on the difference between the core and degree has, on average, the lowest performance, even lower than that of the degree normalization. This could be indeed as result of the very low walking probabilities that are calculated for nodes with a high difference, i.e. nodes with a very high degree, yet a much lower core. For example, the gene *MYC*, which is associated with breast cancer, prostate cancer and height (according to the GWAS catalog), has a degree of 405 and a core of 38, i.e. the difference is 367. In this case the probability for walking to the node would be very low, and therefore its weight after the propagation would not be significant, which might result in a wrong prediction for the gene. In such cases the ratio-based normalization produces a higher probability, which could still result in a significant weight, and therefore the gene would be correctly predicted. Indeed, the ratio-based normalization outperforms the difference-based one, and is on average also slightly better than the degree-based one.

Even with core normalization, the performance measured by the AUROC was at best still under 0.7, and sometimes as low as 0.5. One potential explanation is that the prediction is based on the P-value of the weight at the end of the propagation, which is calculated according to a permutation test that aims to account for the study bias. As a result, some high degree nodes will not have a significant weight at the end of the propagation. However, disease genes tend to have higher degrees, as previously demonstrated for cancer [257], and as is reflected in the GWAS gene sets that we selected (see Figure 6.2). Therefore, some high degree disease nodes might have been wrongfully classified as negatives, thereby reducing the overall performance. This issue was already previously identified as problematic when adjusting propagation scores [86], and can be addressed, for example, by combining the weight and the significance level to produce more balanced predictions [36].

At the same time, it is important to note that the negative set which was selected for the evaluation might in fact include genes which are associated with the disease, yet were not included in the GWAS catalog. This would result in correct predictions that, under this scheme, are marked as false-positives. One possible solution would be to construct a "real" negative set or to use different sets for the evaluations, and estimate the average performance.

Choosing an interaction network is central to running network propagation. To date there is a plethora of PPI resources from which PPI networks can be constructed [21]. Yet, these resources still differ from one another, especially since the interactions can

be established via multiple technologies, including *in silico* predictions (see for example [302]). Recently, Huang et al. [133] evaluated 21 different molecular interaction networks when identifying disease genes via network propagation. They found that the performance varies greatly between different networks, depending on the set of disease genes. They concluded that although the number of interactions was the only strong predictor of a network's performance, agglomerating interaction data from multiple resources is even more essential for improving the performance. This prompted us to exploit CPDB, which is a meta-database and contains PPI interactions from 19 different resources. Since the interactions in CPDB are also associated with a confidence score, we could focus only on the high ones, to assure accuracy.

By comparing our propagation-based results we confirmed that the performance of the high confidence network was improved in comparison to the entire network. Even though the high confidence network is much smaller (with less than 50% interactions), its quality is favorable when trying to identify disease genes via network propagation. On the other hand, taking the entire network and incorporating the weights of the interactions into the propagation formulation was disadvantageous, and the performance was drastically reduced. This is likely due to our permutation-based test, since the generation of random graphs also prompts random assignments of weights for the interactions. As a result, the propagation weights of fewer nodes are statistically significant, and thus there are much less correct predictions.

We thus conclude that in addition to agglomerating interaction data from multiple resources, it is also crucial to carefully account only for the most accurate interactions. This holds true for identifying disease modules via other approaches, as sparsifying the network by taking only "strong" interactions was also beneficial in unsupervised module identification for GWAS disease modules [62].

Beyond the interaction network itself, the technique for generating random networks is also central to the performance, as the statistical significance is calculated based on a permutation test of the interaction network. The test calls for generating random networks which are based on the input network and also preserve the degree for each node. While the edge swap algorithm ensures that the degree is preserved, it does not necessarily maintain other node metrics like core. To that end, we generated dk-random graphs, which produced similar core distributions to the one in the CPDB PPI network.

It is important to note that even in dk-random graphs the core does not always remain exactly the same for each one of the nodes. As a matter of fact, for $d = 2.5$ we observed nodes with a core which was higher than the maximal core in the CPDB PPI network. Nonetheless, to the best of our knowledge, there are no current techniques for generating network permutations which preserve the core for each and every one of the nodes. In the future, it would be useful to examine alternative techniques for generating random networks, and perhaps account for this issue directly within the statistical test.

So far, existing propagation-based approaches were applied to experimental evidence, i.e. to results from a specific study, or to previously established prior knowledge. The latter relies on the guilt-by-association principle, where evidence is spread from phenotype-associated genes with the goal of identifying additional phenotype-relevant ones. Yet, at the time of the development of this work, none of the propagation methods combined

in their framework both experimental and prior evidences. Only very recently Hristov et al. [130] proposed to inform the propagation of new evidence by incorporating existing knowledge of disease genes. They showed that even when the propagation was guided using only a small number of known cancer genes the results were improved in comparison to the standard approach (using data alone). The approach that we proposed incorporates the prior information only at the module identification step, and not during the propagation. The generated network modules connect the well-known genes with novel predictions. Hence, the main idea is the same: to enhance existing knowledge and use it to improve predictions of novel genes.

Additionally, in this work we also observed that the genes from most of the 11 GWAS gene sets are scattered in the network. In fact, it has already been observed that for many diseases the genes associated with them are poorly interconnected, and the largest sub-graph they induce comprises, on average, only 20% of the entire disease gene set [191]. When most of the genes are disconnected from one another, and when the existing components are rather small, it is difficult to extract modules that are functionally relevant to the disease. This issue was also addressed by our approach as it contributed to producing disease modules with higher connectivity between disease genes, by connecting them through intermediate novel nodes. This allowed for the genes to be less dispersed in multiple modules, also in comparison to modules produced by another propagation-based method.

On top of that, incorporating prior knowledge also facilitates the construction of network modules which include high degree nodes that are associated with the disease. As discussed, adjusting the propagation weights to account for study bias might result in the exclusion of high degree nodes from the final modules. However, some of these high degree nodes might in fact already be associated with the phenotype, and so by guiding the procedure with prior knowledge we were able to include them in the network modules, regardless of the significance of their propagation weight.

NetCore runs with three essential parameters: 1) the restart parameter, 2) the maximal P-value threshold and 3) the minimal weight threshold. These parameters were optimized for the purpose of this work and their influence on the results was examined.

The restart parameter allows us to control the trade-off between false and correct predictions. We evaluated the influence of the restart parameter on the performance, using low, intermediate and high values, and concluded that the highest one was the most appropriate for our setting. Nonetheless, it should be emphasized that while the restart parameter was adjusted for the CPDB PPI and optimized for the purpose of identifying GWAS-derived gene sets, it is not necessarily suited for other networks or other kinds of data. In fact, any propagation-based method ought to suggest the ideal value for its execution, and several other approaches have been previously tested.

For example, Leiserson et al. [166] optimized the restart parameter such that the amount of weight that is diffused to the neighbors of the source is larger than the weight that is diffused to the neighbors of the neighbors. They also found that varying the optimal value by +/-10% only slightly changed the results. Reyna et al. [242] attempted to achieve the same goal by optimizing the restart parameter such that the overall probability of walking to the neighbors of the source is equivalent to the prob-

ability of walking to any other node in the network (which is not in the first order neighborhood). Recently, Huang et al. [133] constructed a linear model which is based on the network size (measured by the number of edges) to estimate an optimal restart parameter. They found that for 21 different networks, and a set of genes from hallmark pathways, the best performance is achieved when the restart parameter was proportional to the $\log_{10}$-adjusted number of interactions in the network.

Ultimately, before applying any propagation-based method, including NetCore, the user must consider the effect of the restart parameter, and seek to determine the optimal one for both the network and the data.

The other parameters in NetCore determine the genes which will be included in the final network modules. Both the P-value and weight thresholds are computed from the data itself, at the convergence of the propagation, and thus can not be generally optimized. Setting a P-value threshold depends in principle on the statistical test that was applied. Accordingly, we opted to set it to the minimum significance level that can be achieved under our statistical model. Furthermore, we established that the ideal weight threshold depends largely on the range of the input weights, and therefore estimated it based on the distribution of the propagation weights.

Some propagation-based methods, for instance NAGA, do not apply any statistical test, and therefore rely only on the propagation weights. Yet, in order to provide a reasonably sized prediction list, they are still required to set a threshold based on the weights. Other methods, which do apply some statistical test, like HotNet2 and Hierarchical HotNet, must also estimate some parameter for an ideal execution of their approach, according to their model. In fact, HotNet2 provides four versions of the results, which correspond to four alternative parameter values, and alter both the number of predicted genes as well as the sizes of the reported modules.

We acknowledge that the sizes of the final modules in NetCore partially depend on the setting of these thresholds, and so they govern the trade-off between small modules (with less novel predictions) and larger ones (with more potential false predictions). Furthermore, we also observed that there is a correlation between the size of the input (i.e. number of seed genes) and the size of the modules. Our evaluations were focused on the GWAS gene sets, and therefore the results are with respect to binary weights only. Nevertheless, since the threshold is set according to the distribution of the weights after the propagation, it is in principle also suitable for any range of input data.

At last, even though we attempted to select the optimal thresholds, it is still encouraged to examine the results with respect to the data, and adjust the parameters accordingly, if needed.

To demonstrate the usability of our method to diverse phenotypes and data sets we applied NetCore to a genome-wide schizophrenia study, pan-cancer mutation data, and expression levels measured upon drug treatment.

By combining experimental evidence from a large genetic variation study, and a list of well-known disease genes extracted from a curated database, we were able to predict novel candidate genes that might be associated with schizophrenia. We demonstrated NetCore's relevance by first comparing it with NAGA, which is a propagation-based method tailored for the analysis of GWAS data. Not only did we find that NetCore predicts more novel genes than NAGA, but also it can predict novel genes that are sig-

nificant in future GWAS studies. Furthermore, the modules identified by NetCore were strongly enriched in disease-relevant pathways, stronger than the enrichment of the genes detected by NAGA. This was also further demonstrated by the disease-relevance of two of the predicted genes, namely *SRC* and *PSEN1*. Both of these genes reside in regions of the network that appear to be enriched with schizophrenia-associated genes, and were also previously linked either directly to schizophrenia or to diseases that arise form similar mechanisms.

We also tested NetCore's ability to predict novel cancer genes based on cancer mutation data and compared the results to the state-of-the-art methods in the field, HotNet2 and Hierarchical HotNet. Both methods have previously been shown to predict the highest numbers of candidate cancer genes [242] in comparison to other network-based methods for cancer gene predictions. However, many cancer consensus genes are greatly affected by the study bias and therefore have high degrees in PPI networks [257]. Thus, applying a standard degree normalization in network propagation will, on the one hand, facilitate the identification of such cancer genes, yet on the other hand, prohibit the detection of cancer genes with lower degrees. Indeed, we demonstrated that both HotNet2 and Hierarchical HotNet, while being very specific, reported a rather low number of novel cancer genes, and mainly those with a relative high node degree. In contrast, NetCore, which benefited both from using node core and from incorporating prior knowledge, detected more candidate cancer genes, which have lower degrees, and are also connected to well-known consensus cancer genes. This indicates the power of a) using core instead of degree and b) using a pre-defined list of well-known cancer genes in order to predict novel cancer genes. We additionally confirmed the relevance of these novel predictions using an independent data set, where the novel genes (which were identified by mutation data only) were associated with disease progression based on their expression levels.

We further exemplified that NetCore can be suitable for diverse types of questions and data by applying it to gene expression levels measured upon treatment with anthracyclines in order to identify toxicity mechanisms. The toxicogenomics workflow, which was previously developed by us, was adapted in order to analyze the results from a more recent study, where expression levels were measured by RNA-seq. We proposed both a general scoring scheme and a more stringent one, in order to accommodate for instances where the user wishes to provide scores only for a selected number of genes, for instance those that are significantly differentially expressed, rather than the entire transcriptome. In these scores the time effect was summarized by averaging over all time points, however other measures can be used to compute dynamic longitudinal scores. One could apply a mathematical model to detect differential expression over time, e.g. as the one suggested by Conesa et al. [66]. Using both score variations, in comparison with HotNet2, not only did NetCore produce fewer false-positive predictions, but it was also more robust with respect to the size of the input. Without prior information the results from NetCore reflect a combination between the most differentially expressed genes and the significant genes which were detected via network propagation, but not necessarily via the differential expression analysis. Yet, without prior knowledge the overlap between the genes in the final modules of all drugs is rather low. Incorporating prior knowledge increased this overlap, and thus we were able to combine the drug-modules into one module with relevant cardiotoxicity signals.

This work describes how NetCore improves network propagation results for the identification of novel disease genes and modules. However, in addition to the modifications that we proposed, there are still several challenges to address and room for advancement.

First, the study bias in PPI networks can be further reduced by designing more accurate experiments for studying PPIs, without focusing on well characterized nodes only. This will hopefully promote better representations of the interactions so that each node will be associated with a correct number of connections in the network. In addition, while CPDB is already integrating over multiple resources, it has been shown that summarizing information from multiple PPI networks can even further improve the results [133]. One could also consider other interaction networks, which represent different types of molecular interactions and consist of alternative modules (see for example [168] and [164]).

Second, for an ideal performance, NetCore requires a set of seed genes in order to identify comprehensive modules. Great efforts have been made in the field of cancer for comprising lists of genes that are associated with the disease [22, 240, 273]. Yet, even when the prior knowledge about causal genes for a given disease is limited, it is nevertheless useful to exploit it in order to improve predictions [130]. For this work we exclusively used (non-cancer) disease-gene associations from the GWAS catalog and the DisGeNET databases. To date there are many more resources which could be beneficial to consider. That being said, it is important to remember that the level of agreements between these resources varies (see for example [32, 230]), and that the results should be carefully inspected, and perhaps validated using other independent resources or studies. In the future, it would be very useful if similar curated lists as in the case of cancer would be produced for other diseases too.

Third, it is in principle possible to tune the final modules in NetCore with additional post-processing steps, according to different node metrics. For example, if one wishes to further exclude genes from the periphery of the network, and focus only on core genes, one could remove genes with a low core value from the final modules. For instance, if we were to apply a threshold of core > 5 for NetCore's results on pan-cancer mutation data, this would only exclude one of the genes that are in the candidate cancer list, and reduce the detected novel candidates from 63 to 57 (and by that also the potential false positives), while preserving the potential biomarkers. On the other hand, if one wishes to expand the final modules, it is also possible to include additional nodes which are discarded by NetCore. Currently NetCore considers only nodes which are direct neighbors of the input seed genes. Alternatively, it could be adapted to include also nodes which are not direct neighbors of any of the seed genes. This, however, would require to find an appropriate path between the potential gene and at least one of the seed genes, in order to maintain the connectivity between the nodes in the final modules.

Fourth, network propagation can further be used as a method for integrating multiple types of genomic data [71]. For example, it has been applied to the identification of cancer genes based on both mutation data and gene expression levels [223, 251]. In [263] we proposed, in collaboration with others, an integrated approach to analyze proteomics and transcriptomics data by calculating a combined score and applying network propagation to identify integrated modules. The functional enrichment of the integrated modules was higher than when the modules were computed separately.

Moreover, the integrated modules included proteins which were only measured by gene expression levels, and were not detected using proteomics alone. For example, *TRIM55* and *TRIM28*, which belong to the superfamily of *TRIM* (Tripartite motif-containing) proteins, were identified in the integrated modules, even though they could not be fully identified by proteomics only. A similar approach could also be implemented using NetCore. In fact, *TRIM21*, which is from the same family of proteins, was already identified in NetCore's combined drug-toxicity module, and is connected to two cardiomyopathy-related genes, *TNNI3* and *TNNC1*.

Finally, NetCore could further be used beyond the purpose of genotype-phenotype associations. If applied to proteomics data, NetCore could assist in identifying proteins which are not detectable due to technological limitations (see for example [274]). By propagating the experimental evidence from the measured proteins, it is possible to also reach the ones that were not measured, and assign a propagation score to them. This score could help to infer the missing data and determine the relevance of the protein to the experiment. Another possibility is to use NetCore as an initial step for re-ranking of genes and extracting relevant features in the context of machine learning. For example, network propagation was recently applied to the detection of additional drug targets, which were then used as input for predicting anticancer drug sensitivity with neural networks [213].

To conclude, even though there exist other modifications of random walk with restart, for instance random walk with extended restart (RWER) [141] where each node has its own restart probability, such adjustments of the adjacency matrix as suggested in this work had yet to be introduced. We thus advise researchers to apply core normalization to improve network propagation results, in particular for genotype-phenotype associations. In the future, other modifications could be adapted to improve the results even further. We additionally advise researchers to combine experimental evidence together with prior knowledge to promote improved predictions and construction of comprehensive network modules. Very recently, two other propagation-based approaches advocated for the incorporation of prior knowledge [107, 130], albeit using alternative techniques. In consistency with our observations, this supports the idea of leveraging well-established knowledge in order to advance it further. Presumably, more and more new methods that lead to further progress in the field will soon emerge.

A Supplementary Figures

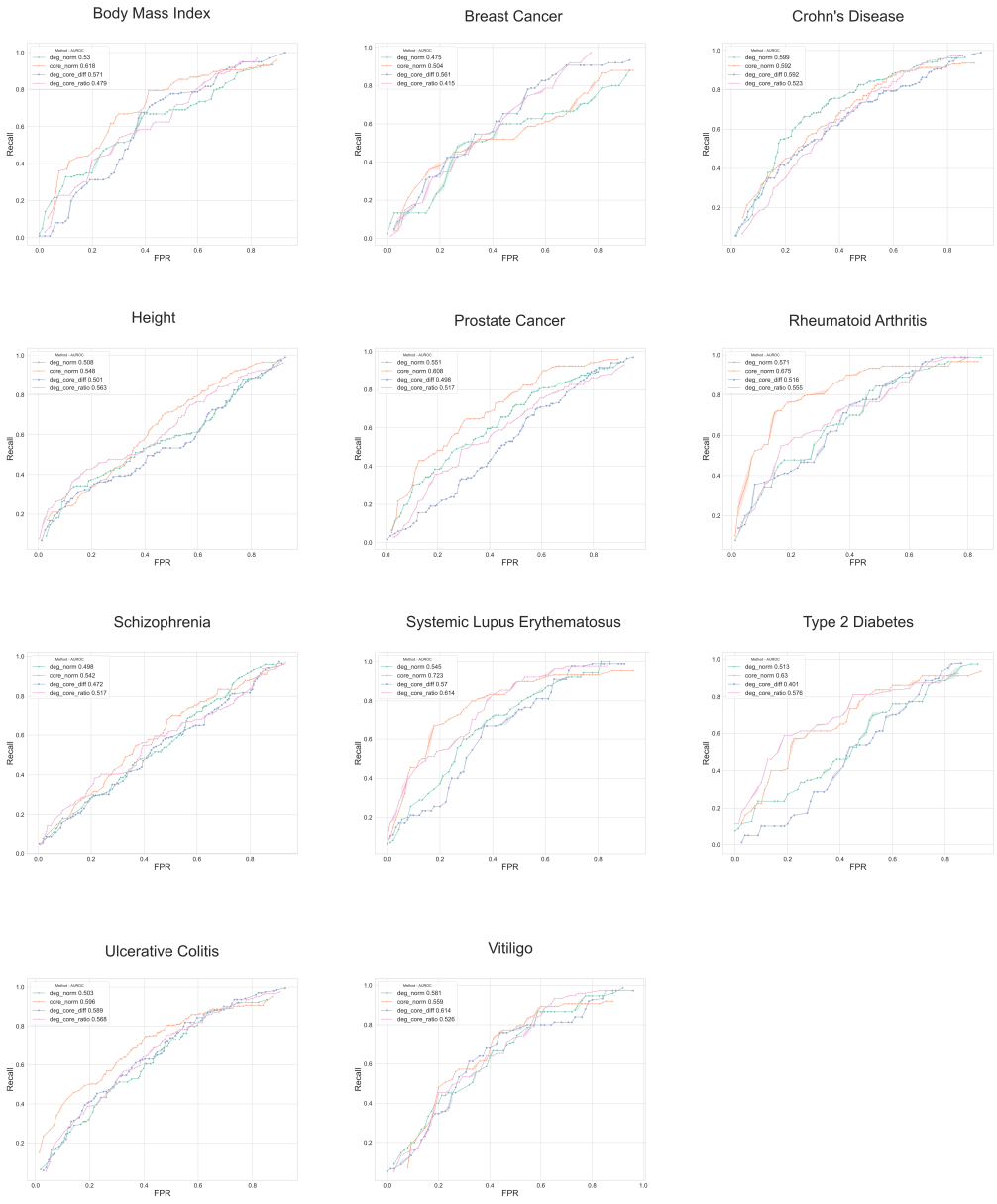


Figure A.1: ROC curves for all 11 GWAS gene sets. The different colors imply different normalization schemes. The lines depict the mean curve for the 5-cross validation results.

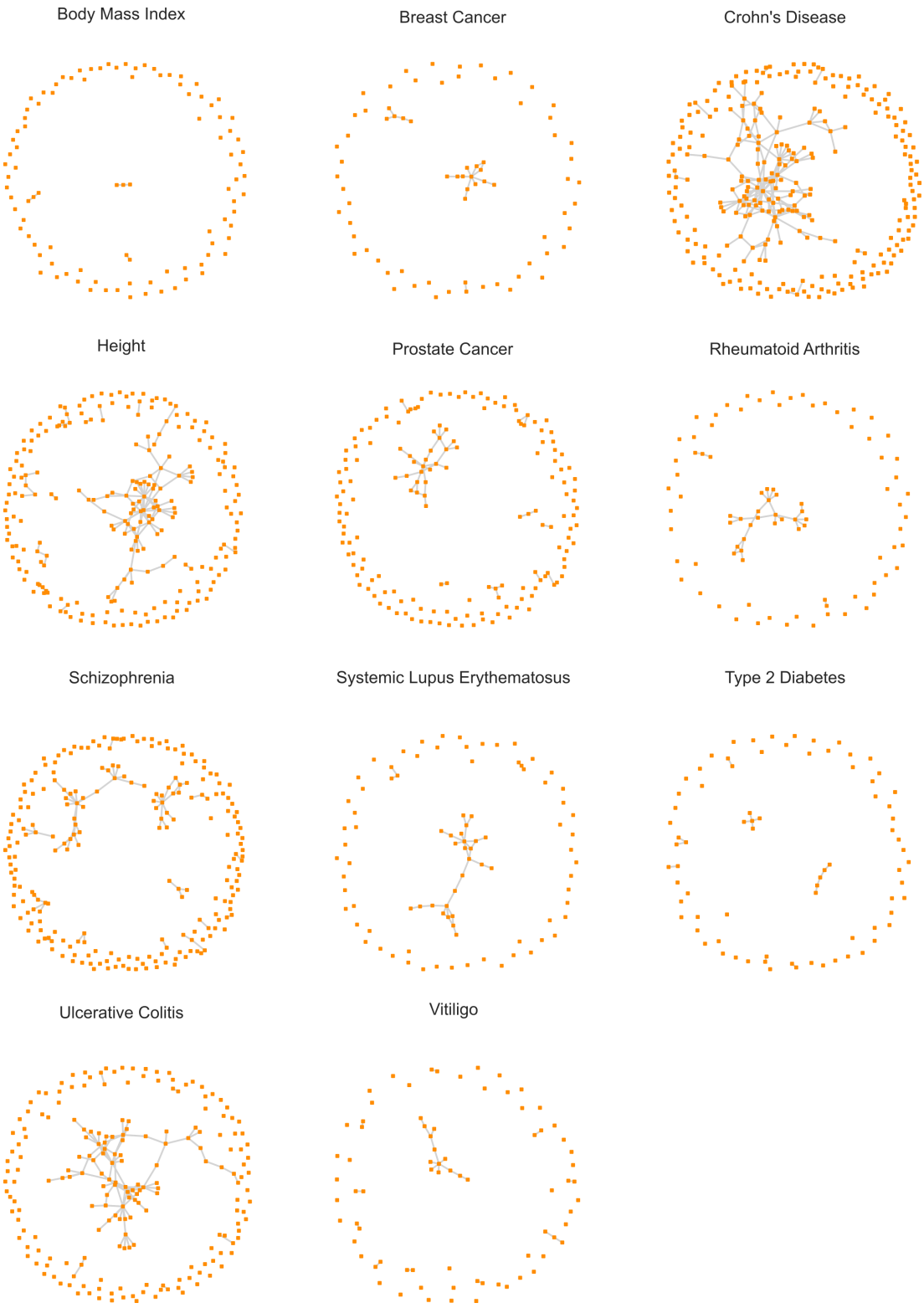


Figure A.2: Seed sub-networks for 11 GWAS gene sets. Each node is a gene from the gene set, and each edge is an interaction from the PPI network.

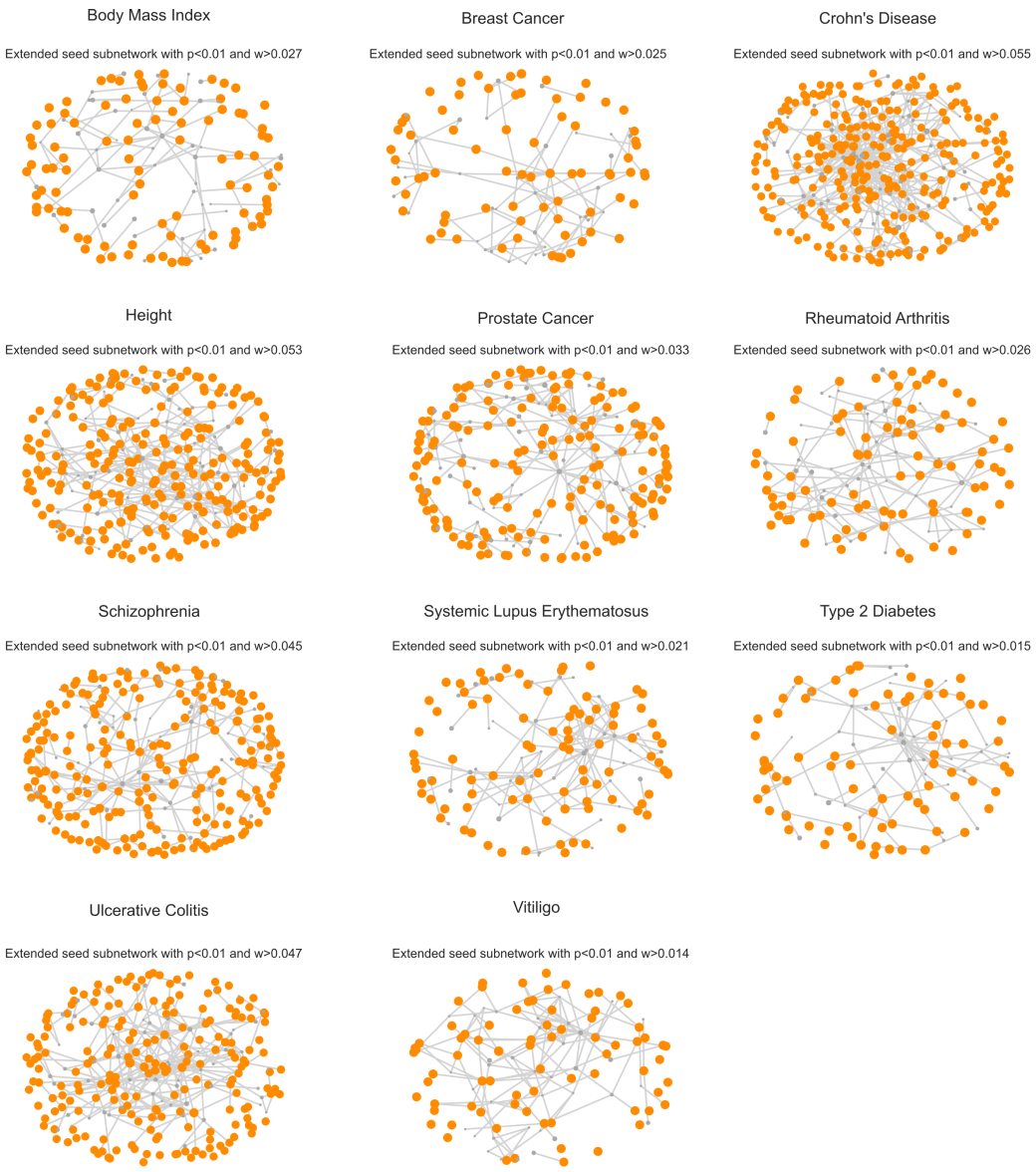


Figure A.3: Extended seed sub-networks for 11 GWAS gene sets from NetCore. The orange nodes are original seed nodes, the gray nodes were added to the seed sub-network after the propagation, according to their results (significant P-value of $p < 0.01$ and a minimum weight, which was calculated based on the weights distribution after the propagation). The sizes of the nodes reflect their weights after the propagation. The edges are originally from the PPI network.

Figure A.4: Largest modules for 11 GWAS sets from NetCore. The largest module is extracted from the extended seed sub-network (Figure A.3), where it is the largest connected component of the sub-network. Orange genes are in the original gene sets (seed nodes), and gray ones were added after the propagation. The edges are originally from the PPI network.

Figure A.5: Over-representation analysis (Section 6.4.1) was executed for each GWAS gene set twice: 1) for the genes in the largest component in the seed sub-network (Figure A.2) and 2) for the genes in the largest NetCore modules (Figure A.4). The enriched pathways are depicted for each GWAS gene set for which there was an overlap of at least 4 genes between the pathway set and the queried genes. The red color scale indicates the enrichment score (measured by $-\log_{10}$ of the enrichment Q-value), and the blue color scale indicates the number of overlapping genes within the pathway.

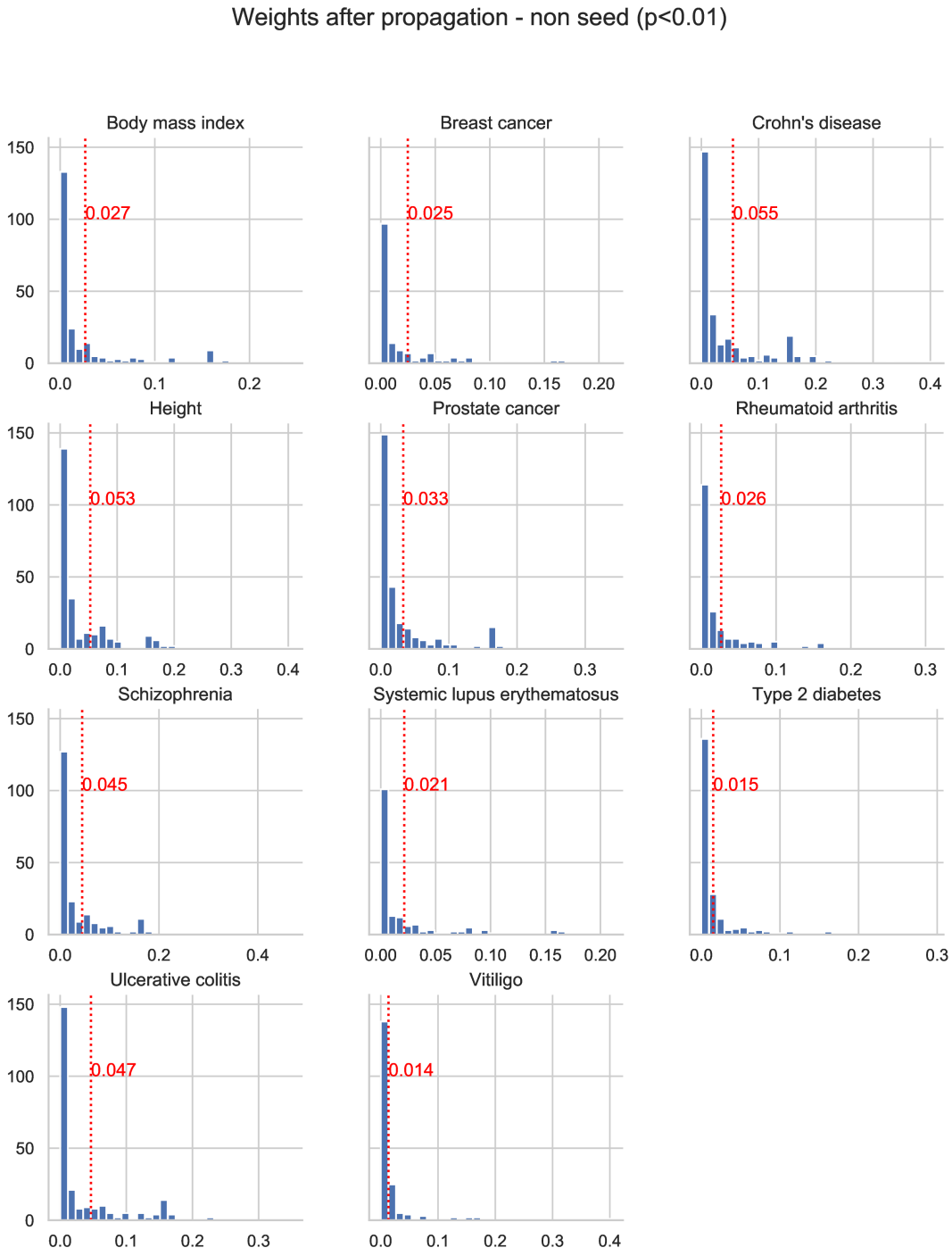


Figure A.6: Distribution of the weights after the propagation for genes in the network with a significant level ($p < 0.01$), which are not part of the input seed list. The propagation was applied using a binary scoring scheme, i.e. the initial weight for seed genes was 1 and for the rest of the genes in the network 0. The X-axis denotes the weights after the propagation, which vary in range, depending on the GWAS gene set. The vertical dashed red line denotes the 75th percentile of the distribution, which is the minimum weight criterion chosen for NetCore.

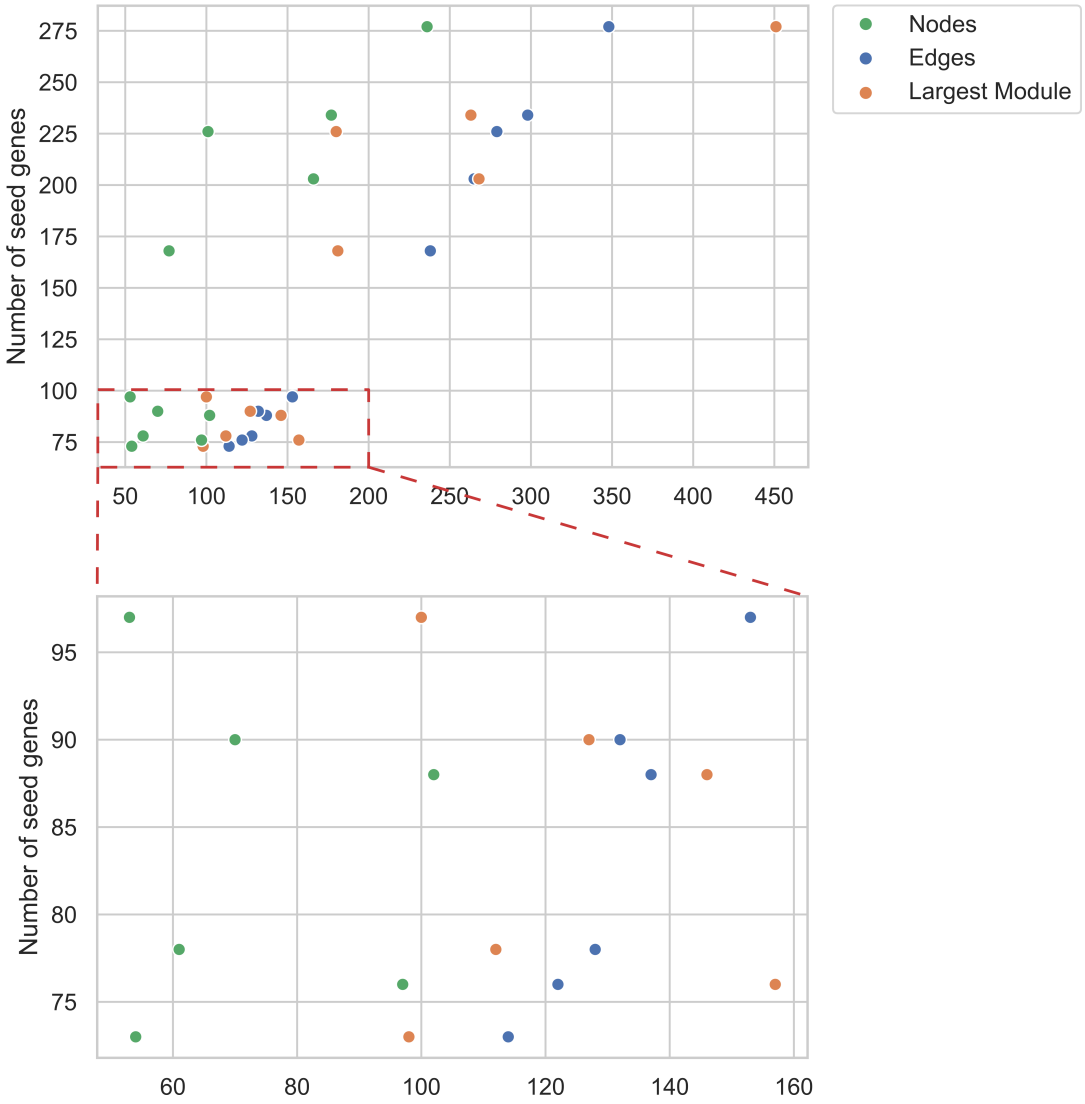


Figure A.7: For every GWAS gene set is denoted: on the Y-axis the size of the set (within the CPDB PPI network) and on the X-axis the sizes of the extended seed sub-network measured by: the number of nodes (green), number of edges (blue) and number of nodes in the largest module (orange).

Bibliography

[1] Michaela Adamcova, Veronika Skarkova, Jitka Seifertova, and Emil Rudolf. Cardiac troponins are among targets of doxorubicin-induced cardiotoxicity in hiPCS-CMs. *International journal of molecular sciences* 20.11 (2019), 2638.

[2] Euan A Adie, Richard R Adams, Kathryn L Evans, David J Porteous, and Ben S Pickard. Speeding disease gene discovery by sequence based candidate prioritization. *BMC bioinformatics* 6.1 (2005), 55.

[3] Ivan A Adzhubei, Steffen Schmidt, Leonid Peshkin, Vasily E Ramensky, Anna Gerasimova, Peer Bork, Alexey S Kondrashov, and Shamil R Sunyaev. A method and server for predicting damaging missense mutations. *Nature methods* 7.4 (2010), 248–249.

[4] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Frederik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, et al. Gene prioritization through genomic data fusion. *Nature biotechnology* 24.5 (2006), 537–544.

[5] Rafsan Ahmed, Ilyes Baali, Cesim Erten, Evis Hoxha, and Hilal Kazan. MEXCOwalk: mutual exclusion and coverage based random walk to identify cancer modules. *Bioinformatics* 36.3 (2019), 872–879.

[6] Uri David Akavia, Oren Litvin, Jessica Kim, Felix Sanchez-Garcia, Dylan Kotliar, Helen C Causton, Panisa Pochanard, Eyal Mozes, Levi A Garraway, and Dana Pe'er. An integrated approach to uncover drivers of cancer. *Cell* 143.6 (2010), 1005–1017.

[7] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74 (1 2002), 47–97.

[8] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Diameter of the world-wide web. *Nature* 401.6749 (1999), 130–131.

[9] Can Alkan, Bradley P Coe, and Evan E Eichler. Genome structural variation discovery and genotyping. *Nature Reviews Genetics* 12.5 (2011), 363–376.

[10] Benjamin L Allen and Dylan J Taatjes. The Mediator complex: a central integrator of transcription. *Nature reviews Molecular cell biology* 16.3 (2015), 155–166.

[11] Patrick Aloy and Robert B Russell. Potential artefacts in protein-interaction networks. *FEBS Letters* 530.1-3 (2002), 253–254.

[12] Joanna S Amberger, Carol A Bocchini, François Schiettecatte, Alan F Scott, and Ada Hamosh. OMIM. org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic acids research* 43.D1 (2015), D789–D798.

[13] Jordan Anaya. OncoLnc: linking TCGA survival data to mRNAs, miRNAs, and lncRNAs. *PeerJ Computer Science* 2 (2016), e67.

[14] Melvin E Andersen, Harvey J Clewell III, Edilberto Bermudez, Gabrielle A Willson, and Russell S Thomas. Genomic signatures and dose-dependent transitions in nasal epithelial responses to inhaled formaldehyde in the rat. *Toxicological sciences* 105.2 (2008), 368–383.

[15] Roy M Anderson, B Anderson, and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

[16] Mauren Isfer Anghebem-Oliveira, Bruna Rodrigues Martins, Dayane Alberton, Edneia Amancio de Souza Ramos, Geraldo Picheth, and Fabiane Gomes de Moraes Rego. Type 2 diabetes-associated genetic variants of FTO, LEPR, PPARg, and TCF7L2 in gestational diabetes in a Brazilian population. *Archives of endocrinology and metabolism* 61.3 (2017), 238–248.

[17] Silke Appel-Cresswell, Ilaria Guella, Anna Lehman, Dean Foti, and Matthew J Farrer. PSEN1 p. Met233Val in a complex neurodegenerative movement and neuropsychiatric disorder. *Journal of movement disorders* 11.1 (2018), 45.

[18] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics* 25.1 (2000), 25–29.

[19] Christina Backes, Alexander Rurainski, Gunnar W Klau, Oliver Müller, Daniel Stöckel, Andreas Gerasch, Jan Küntzer, Daniela Maisel, Nicole Ludwig, Matthias Hein, et al. An integer linear programming approach for finding deregulated subgraphs in regulatory networks. *Nucleic acids research* 40.6 (2012), e43–e43.

[20] Gary D Bader, Doron Betel, and Christopher WV Hogue. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Research* 31.1 (2003), 248–250.

[21] Gary D Bader, Michael P Cary, and Chris Sander. Pathguide: a Pathway Resource List. *Nucleic Acids Research* 34 (2006), D504–D506.

[22] Matthew H Bailey, Collin Tokheim, Eduard Porta-Pardo, Sohini Sengupta, Denis Bertrand, Amila Weerasinghe, Antonio Colaprico, Michael C Wendl, Jaegil Kim, Brendan Reardon, et al. Comprehensive characterization of cancer driver genes and mutations. *Cell* 173.2 (2018), 371–385.

[23] Michael J Bamshad, Sarah B Ng, Abigail W Bigham, Holly K Tabor, Mary J Emond, Deborah A Nickerson, and Jay Shendure. Exome sequencing as a tool for Mendelian disease gene discovery. *Nature Reviews Genetics* 12.11 (2011), 745–755.

[24] Anamika Banerjee, Hoau-Yan Wang, Karin E Borgmann-Winter, Mathew L MacDonald, Hagop Kaprielian, Andres Stucky, Jessica Kvasic, Chijioke Egbujo, Rabindranath Ray, Konrad Talbot, et al. Src kinase as a mediator of convergent molecular abnormalities leading to NMDAR hypoactivity in schizophrenia. *Molecular psychiatry* 20.9 (2015), 1091–1100.

[25] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science* 286.5439 (1999), 509–512.

[26] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature reviews genetics* 12.1 (2011), 56–68.

[27] Albert-László Barabási and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics* 5.2 (2004), 101–113.

[28] Rodrigo Barderas, Ingrid Babel, Ramón Díaz-Uriarte, Víctor Moreno, Adolfo Suárez, Felix Bonilla, Roi Villar-Vázquez, Gabriel Capellá, and J Ignacio Casal. An optimized predictor panel for colorectal cancer diagnosis based on the combination of tumor-associated antigens obtained from protein and phage microarrays. *Journal of proteomics* 75.15 (2012), 4647–4655.

[29] Gal Barel and Ralf Herwig. Network and pathway analysis of toxicogenomics data. *Frontiers in genetics* 9 (2018), 484.

[30] Gal Barel and Ralf Herwig. NetCore: a network propagation approach using node coreness. *Nucleic Acids Research* 48.17 (2020), e98–e98.

[31] Vladimir Batagelj and Matjaz Zaversnik. An O (m) algorithm for cores decomposition of networks. *arXiv* 0310049 (2003).

[32] Anna Bauer-Mehren, Markus Bundschus, Michael Rautschka, Miguel A. Mayer, Ferran Sanz, and Laura I. Furlong. Gene-Disease Network Analysis Reveals Functional Modules in Mendelian, Complex and Environmental Diseases. *PLoS one* 6.6 (2011), 1–13.

[33] Anna Bauer-Mehren, Laura I Furlong, and Ferran Sanz. Pathway databases and tools for their exploitation: benefits, current limitations and challenges. *Molecular Systems Biology* 5.1 (2009), 290.

[34] Daniela Beisser, Gunnar W. Klau, Thomas Dandekar, Tobias Müller, and Marcus T. Dittrich. BioNet: an R-Package for the functional analysis of biological networks. *Bioinformatics* 26.8 (2010), 1129–1130.

[35] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995), 289–300.

[36] Matteo Bersanelli, Ettore Mosca, Daniel Remondini, Gastone Castellani, and Luciano Milanesi. Network diffusion-based analysis of high-throughput data for the detection of differentially enriched modules. *Scientific reports* 6 (2016), 34841.

[37] Hadas Biran, Tovi Almozlino, Martin Kupiec, and Roded Sharan. WebPropagate: A Web Server for Network Propagation. *Journal of Molecular Biology* 430.15 (2018), 2231 –2236.

[38] Hadas Biran, Martin Kupiec, and Roded Sharan. Comparative analysis of normalization methods for network propagation. *Frontiers in genetics* 10 (2019), 4.

[39] Melanie A Blevins, Mingxia Huang, and Rui Zhao. The Role of CtBP1 in Oncogenic Processes and Its Potential as a Therapeutic Target. *Molecular Cancer Therapeutics* 16.6 (2017), 981–990.

[40] Walter Bodmer and Carolina Bonilla. Common and rare variants in multifactorial susceptibility to common diseases. *Nature genetics* 40.6 (2008), 695.

[41] Béla Bollobás. The evolution of sparse graphs, in Graph theory and combinatorics proceedings. *Cambridge Combinatorial Conference in Honour of Paul Erdos*. 1984, 335–357.

[42] Béla Bollobás. *Random graphs*. 73. Cambridge university press, 2001.

[43] David Botstein and Neil Risch. Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease. *Nature genetics* 33.3 (2003), 228–237.

[44] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE transactions on knowledge and data engineering* 20.2 (2007), 172–188.

[45] A Bravo, M Cases, N Queralt-Rosinach, F Sanz, and LI Furlong. A knowledge-driven approach to extract disease-related biomarkers from the literature. *BioMed research international* 2014 (2014).

[46] Àlex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I Furlong. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC bioinformatics* 16.1 (2015), 55.

[47] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine (1998).

[48] James H Bullard, Elizabeth Purdom, Kasper D Hansen, and Sandrine Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC bioinformatics* 11.1 (2010), 1–13.

[49] Annalisa Buniello, Jacqueline A L MacArthur, Maria Cerezo, Laura W Harris, James Hayhurst, Cinzia Malangone, Aoife McMahon, Joannella Morales, Edward Mountjoy, Elliot Sollis, et al. The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Research* 47.D1 (2018), D1005–D1012.

[50] William S Bush and Jason H Moore. Chapter 11: Genome-Wide Association Studies. *PLoS Computational Biology* 8.12 (2012), 1–11.

[51] Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, et al. The UK Biobank resource with deep phenotyping and genomic data. *Nature* 562.7726 (2018), 203–209.

[52] Tolga Can, Orhan Çamoundefinedlu, and Ambuj K. Singh. Analysis of Protein-Protein Interaction Networks Using Random Walks. *Proceedings of the 5th International Workshop on Bioinformatics*. BIOKDD '05. Association for Computing Machinery, 2005, 61–68.

[53] Daniel E Carlin, Samson H Fong, Yue Qin, Tongqiu Jia, Justin K Huang, Bokan Bao, Chao Zhang, and Trey Ideker. A Fast and Flexible Framework for Network-Assisted Genomic Association. *iScience* 16 (2019), 155–161.

[54] Michael P Cary, Gary D Bader, and Chris Sander. Pathway information for systems biology. *FEBS Letters* 579.8 (2005), 1815–1820.

[55] Ethan Cerami, Jianjiong Gao, Ugur Dogrusoz, Benjamin E. Gross, Selcuk Onur Sumer, Bülent Arman Aksoy, Anders Jacobsen, Caitlin J Byrne, Michael L Heuer, Erik Larsson, et al. The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data. *Cancer Discovery* 2.5 (2012), 401–404.

[56] Hui Chen, Zhiying Xu, Bin Yang, Xiaoli Zhou, and Hongfang Kong. RASGRF1 Hypermethylation, a Putative Biomarker of Colorectal Cancer. *Annals of Clinical & Laboratory Science* 48.1 (2018), 3–10.

[57] Feixiong Cheng, Junfei Zhao, and Zhongming Zhao. Advances in computational approaches for prioritizing driver mutations and significantly mutated genes in cancer genomes. *Briefings in Bioinformatics* 17.4 (2015), 642–656.

[58] Wei-Chung Cheng, I-Fang Chung, Chen-Yang Chen, Hsing-Jen Sun, Jun-Jeng Fen, Wei-Chun Tang, Ting-Yu Chang, Tai-Tong Wong, Hsei-Wei Wang, et al. DriverDB: an exome sequencing database for cancer driver gene identification. *Nucleic acids research* 42.D1 (2014), D1048–D1054.

[59] Iouri Chepelev, Gang Wei, Qingsong Tang, and Keji Zhao. Detection of single nucleotide variations in expressed exons of the human genome using RNA-Seq. *Nucleic Acids Research* 37.16 (2009), e106–e106.

[60] Ara Cho, Jung Eun Shim, Eiru Kim, Fran Supek, Ben Lehner, and Insuk Lee. MUFFINN: cancer gene discovery via network analysis of somatic mutation data. *Genome biology* 17.1 (2016), 129.

[61] Hyunghoon Cho, Bonnie Berger, and Jian Peng. Compact integration of multi-network topology for functional analysis of genes. *Cell systems* 3.6 (2016), 540–548.

[62] Sarvenaz Choobdar, Mehmet E Ahsen, Jake Crawford, Mattia Tomasoni, Tao Fang, David Lamparter, Junyuan Lin, Benjamin Hescott, Xiaozhe Hu, Johnathan Mercer, et al. Assessment of network module identification across complex diseases. *Nature Methods* 16.9 (2019), 843–852.

[63] Jan Christiansen, Astrid M Kolte, Finn C Nielsen, et al. IGF2 mRNA-binding protein 2: biological function and putative role in type 2 diabetes. *Journal of molecular endocrinology* 43.5 (2009), 187–195.

[64] Elizabeth T Cirulli and David B Goldstein. Uncovering the roles of rare variants in common disease through whole-genome sequencing. *Nature Reviews Genetics* 11.6 (2010), 415–425.

[65] Reuven Cohen and Shlomo Havlin. Scale-free networks are ultrasmall. *Physical review letters* 90.5 (2003), 058701.

[66] Ana Conesa, María José Nueda, Alberto Ferrer, and Manuel Talón. maSigPro: a method to identify significantly differential expression profiles in time-course microarray experiments. *Bioinformatics* 22.9 (2006), 1096–1102.

[67] 1000 Genomes Project Consortium et al. A map of human genome variation from population-scale sequencing. *Nature* 467.7319 (2010), 1061.

[68] 1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature* 526.7571 (2015), 68–74.

[69] International Cancer Genome Consortium et al. International network of cancer genome projects. *Nature* 464.7291 (2010), 993.

[70] Juliana Costa-Silva, Douglas Domingues, and Fabricio Martins Lopes. RNA-Seq differential expression analysis: An extended review and a software tool. *PLoS one* 12.12 (2017), e0190152.

[71] Lenore Cowen, Trey Ideker, Benjamin J Raphael, and Roded Sharan. Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics* 18.9 (2017), 551.

[72] Pau Creixell, Jüri Reimand, Syed Haider, Guanming Wu, Tatsuhiro Shibata, Miguel Vazquez, Ville Mustonen, Abel Gonzalez-Perez, John Pearson, Chris Sander, et al. Pathway and network analysis of cancer genomes. *Nature methods* 12.7 (2015), 615.

[73] Francis Crick. Central dogma of molecular biology. *Nature* 227.5258 (1970), 561–563.

[74] Michael E Cusick, Haiyuan Yu, Alex Smolyar, Kavitha Venkatesan, Anne-Ruxandra Carvunis, Nicolas Simonis, Jean-François Rual, Heather Borick, Pascal Braun, Matija Dreze, et al. Literature-curated protein interaction datasets. *Nature methods* 6.1 (2009), 39.

[75] Ivana Dedinská, Nadežda Mäčková, Daniela Kantárová, Lea Kováčiková, Karol Graňák, L'udovít Laca, Juraj Miklušica, Petra Skálová, Peter Galajda, and Marián Mokáň. Leptin–A new marker for development of post-transplant diabetes mellitus? *Journal of Diabetes and its Complications* 32.9 (2018), 863–869.

[76] Erwin L van Dijk, Yan Jaszczyszyn, Delphine Naquin, and Claude Thermes. The third revolution in sequencing technology. *Trends in Genetics* 34.9 (2018), 666–681.

[77] Li Ding, Michael C Wendl, Joshua F McMichael, and Benjamin J Raphael. Expanding the computational toolbox for mining cancer genomes. *Nature Reviews Genetics* 15.8 (2014), 556–570.

[78] Marcus T Dittrich, Gunnar W Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller. Identifying functional modules in protein–protein interaction networks: an integrated exact approach. *Bioinformatics* 24.13 (2008), i223–i231.

[79] David W Dodington, Harsh R Desai, and Minna Woo. JAK/STAT–emerging players in metabolism. *Trends in Endocrinology & Metabolism* 29.1 (2018), 55–65.

[80] Ian Donaldson, Joel Martin, Berry De Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D Bader, Katerina Michalickova, et al. PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC bioinformatics* 4.1 (2003), 11.

[81] Peter G Doyle and J Laurie Snell. *Random walks and electric networks*. Vol. 22. American Mathematical Soc., 1984.

[82] Marc A van Driel, Koen Cuelenaere, Patrick PCW Kemmeren, Jack AM Leunissen, and Han G Brunner. A new web-based data mining tool for the identification of candidate genes for human genetic disorders. *European Journal of Human Genetics* 11.1 (2003), 57–63.

[83] Evan E Eichler, Jonathan Flint, Greg Gibson, Augustine Kong, Suzanne M Leal, Jason H Moore, and Joseph H Nadeau. Missing heritability and strategies for finding the underlying causes of complex disease. *Nature Reviews Genetics* 11.6 (2010), 446–450.

[84] Kyle Ellrott, Matthew H Bailey, Gordon Saksena, Kyle R Covington, Cyriac Kandoth, Chip Stewart, Julian Hess, Singer Ma, Kami E Chiotti, Michael McLellan, et al. Scalable open science approach for mutation calling of tumor exomes using multiple genomic pipelines. *Cell systems* 6.3 (2018), 271–281.

[85] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), 17–60.

[86] Sinan Erten, Gurkan Bebek, Rob M Ewing, and Mehmet Koyutürk. D A D A: Degree-Aware Algorithms for Network-Based Disease Gene Prioritization. *BioData mining* 4.1 (2011), 19.

[87] Adam D Ewing, Kathleen E Houlahan, Yin Hu, Kyle Ellrott, Cristian Caloian, Takafumi N Yamaguchi, J Christopher Bare, Christine P'ng, Daryl Waggott, Veronica Y Sabelnykova, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nature methods* 12.7 (2015), 623–630.

[88] Stanley Fields and Ok-kyu Song. A novel genetic system to detect protein–protein interactions. *Nature* 340.6230 (1989), 245–246.

[89] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports* 659 (2016), 1–44.

[90] Lude Franke, Harm Van Bakel, Like Fokkens, Edwin D De Jong, Michael Egmont-Petersen, and Cisca Wijmenga. Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *The American Journal of Human Genetics* 78.6 (2006), 1011–1025.

[91] Kelly A Frazer, Sarah S Murray, Nicholas J Schork, and Eric J Topol. Human genetic variation and its contribution to complex traits. *Nature Reviews Genetics* 10.4 (2009), 241–251.

[92] Linton C Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry* 40.1 (1977), 35.

[93] Alex Frolkis, Craig Knox, Emilia Lim, Timothy Jewison, Vivian Law, David D Hau, Phillip Liu, Bijaya Gautam, Son Ly, An Chi Guo, et al. SMPDB: the small molecule pathway database. *Nucleic acids research* 38 (2010), D480–D487.

[94] Tetsuo Fujimaki, Kimihiko Kato, Mitsutoshi Oguri, Tetsuro Yohida, Hideki Horibe, Kiyoshi Yokoi, Sachiro Watanabe, Kei Satoh, Yukitoshi Aoyagi, Masashi Tanaka, et al. Association of a polymorphism of BTN2A1 with dyslipidemia in East Asian populations. *Experimental and therapeutic medicine* 2.4 (2011), 745–749.

[95] P Andrew Futreal, Lachlan Coin, Mhairi Marshall, Thomas Down, Timothy Hubbard, Richard Wooster, Nazneen Rahman, and Michael R Stratton. A census of human cancer genes. *Nature reviews cancer* 4.3 (2004), 177–183.

[96] Brigitte Ganter, Stuart Tugendreich, Cecelia I Pearson, Eser Ayanoglu, Susanne Baumhueter, Keith A Bostian, Lindsay Brady, Leslie J Browne, John T Calvin, Gwo-Jen Day, et al. Development of a large-scale chemogenomics database to improve drug candidate selection and to understand mechanisms of chemical toxicity and action. *Journal of Biotechnology* 119.3 (2005), 219–244.

[97] Jianjiong Gao, Bülent Arman Aksoy, Ugur Dogrusoz, Gideon Dresdner, Benjamin Gross, S Onur Sumer, Yichao Sun, Anders Jacobsen, Rileen Sinha, Erik Larsson, et al. Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal. *Science Signaling* 6.269 (2013), pl1–pl1.

[98] Levi A Garraway and Eric S Lander. Lessons from the cancer genome. *Cell* 153.1 (2013), 17–37.

[99] Kyle J. Gaulton, Karen L. Mohlke, and Todd J. Vision. A computational system to select candidate genes for complex human traits. *Bioinformatics* 23.9 (2007), 1132–1140.

[100] Carrie Anna Geisberg and Douglas B Sawyer. Mechanisms of anthracycline cardiotoxicity and strategies to decrease cardiac damage. *Current hypertension reports* 12.6 (2010), 404–410.

[101] Richard A George, Jason Y Liu, Lina L Feng, Robert J Bryson-Richardson, Diane Fatkin, and Merridee A. Wouters. Analysis of protein sequence and interaction data for candidate disease gene prediction. *Nucleic Acids Research* 34.19 (2006), e130–e130.

[102] Susan Dina Ghiassian, Jörg Menche, and Albert-László Barabási. A DIseAse MOdule Detection (DIAMOnD) Algorithm Derived from a Systematic Analysis of Connectivity Patterns of Disease Proteins in the Human Interactome. *PLoS Computational Biology* 11.4 (2015), 1–21.

[103] Andrew O Giacomelli, Xiaoping Yang, Robert E Lintner, James M McFarland, Marc Duby, Jaegil Kim, Thomas P Howard, David Y Takeda, Seav Huong Ly, Eejung Kim, et al. Mutational processes shape the landscape of TP53 mutations in human cancer. *Nature genetics* 50.10 (2018), 1381–1387.

[104] Abel Gonzalez-Perez, Ville Mustonen, Boris Reva, Graham RS Ritchie, Pau Creixell, Rachel Karchin, Miguel Vazquez, J Lynn Fink, Karin S Kassahn, John V Pearson, et al. Computational approaches to identify functional genetic variants in cancer genomes. *Nature methods* 10.8 (2013), 723.

[105] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics* 17.6 (2016), 333.

[106] Christopher Greenman, Philip Stephens, Raffaella Smith, Gillian L Dalgliesh, Christopher Hunter, Graham Bignell, Helen Davies, Jon Teague, Adam Butler, Claire Stevens, et al. Patterns of somatic mutation in human cancer genomes. *Nature* 446.7132 (2007), 153–158.

[107] Anja C Gumpinger, Kasper Lage, Heiko Horn, and Karsten Borgwardt. Prediction of cancer driver genes through network-based moment propagation of mutation scores. *Bioinformatics* 36 (2020), i508–i515.

[108] Aric A Hagberg, Daniel A Schult, and Pieter J Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. *Proceedings of the 7th Python in Science Conference*. 2008, 11 –15.

[109] Luke Hakes, David L Robertson, and Stephen G Oliver. Effect of dataset selection on the topological interpretation of protein interaction networks. *BMC genomics* 6.1 (2005), 131.

[110] Jing-Dong J Han, Nicolas Bertin, Tong Hao, Debra S Goldberg, Gabriel F Berriz, Lan V Zhang, Denis Dupuy, Albertha JM Walhout, Michael E Cusick, Frederick P Roth, et al. Evidence for dynamically organized modularity in the yeast protein–protein interaction network. *Nature* 430.6995 (2004), 88–93.

[111] Douglas Hanahan and Robert A Weinberg. The hallmarks of cancer. *cell* 100.1 (2000), 57–70.

[112] Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *cell* 144.5 (2011), 646–674.

[113] Christopher Hardt, Moritz Emanuel Beber, Axel Rasche, Atanas Kamburov, DG Hebels, JC Kleinjans, and Ralf Herwig. ToxDB: pathway-level interpretation of drug-treatment data. *Database* 2016 (2016).

[114] Thomas Hartung. Toxicology for the twenty-first century. *Nature* 460.7252 (2009), 208–212.

[115] Leland H Hartwell, John J Hopfield, Stanislas Leibler, and Andrew W Murray. From molecular to modular cell biology. *Nature* 402.6761 (1999), C47–C52.

[116] Hui He, Gang Wu, Haiyang Liu, Ying Cheng, Yanqiu Yu, Yawei Wang, and Yongfeng Liu. Low RIN1 expression in HCC is associated with tumor invasion and unfavorable prognosis. *American journal of clinical pathology* 140.1 (2013), 73–81.

[117] James M Heather and Benjamin Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics* 107.1 (2016), 1 –8.

[118] Ralf Herwig, Christopher Hardt, Matthias Lienhard, and Atanas Kamburov. Analyzing and interpreting genome data at the network level with ConsensusPathDB. *Nature Protocols* 11.10 (2016), 1889–1907.

[119] Abby Hill, Scott Gleim, Florian Kiefer, Frederic Sigoillot, Joseph Loureiro, Jeremy Jenkins, and Melody K. Morris. Benchmarking network algorithms for contextualizing genes of interest. *PLoS Computational Biology* 15.12 (2019), 1–14.

[120] Mizuho Hiramatsu, Mitsutoshi Oguri, Kimihiko Kato, Hideki Horibe, Tetsuo Fujimaki, Sachiro Watanabe, Kei Satoh, Yukitoshi Aoyagi, Masashi Tanaka, Dong-Jik Shin, et al. Synergistic effects of genetic variants of APOA5 and BTN2A1 on dyslipidemia or metabolic syndrome. *International journal of molecular medicine* 30.1 (2012), 185–192.

[121] Jorge E Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences* 102.46 (2005), 16569–16572.

[122] Joel N Hirschhorn and Mark J Daly. Genome-wide association studies for common diseases and complex traits. *Nature reviews genetics* 6.2 (2005), 95–108.

[123] Katherine A Hoadley, Christina Yau, Toshinori Hinoue, Denise M Wolf, Alexander J Lazar, Esther Drill, Ronglai Shen, Alison M Taylor, Andrew D Cherniack, Vésteinn Thorsson, et al. Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell* 173.2 (2018), 291–304.

[124] Katherine A Hoadley, Christina Yau, Denise M Wolf, Andrew D Cherniack, David Tamborero, Sam Ng, Max DM Leiserson, Beifang Niu, Michael D McLellan, Vladislav Uzunangelov, et al. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell* 158.4 (2014), 929–944.

[125] Emily Hodges, Zhenyu Xuan, Vivekanand Balija, Melissa Kramer, Michael N Molla, Steven W Smith, Christina M Middle, Matthew J Rodesch, Thomas J Albert, Gregory J Hannon, et al. Genome-wide in situ exon capture for selective resequencing. *Nature genetics* 39.12 (2007), 1522.

[126] Matan Hofree, John P Shen, Hannah Carter, Andrew Gross, and Trey Ideker. Network-based stratification of tumor mutations. *Nature methods* 10.11 (2013), 1108–1115.

[127] Gustav Holmgren, Jane Synnergren, Yalda Bogestål, Caroline Améen, Karolina Åkesson, Sandra Holmgren, Anders Lindahl, and Peter Sartipy. Identification of novel biomarkers for doxorubicin-induced toxicity in human cardiomyocytes derived from pluripotent stem cells. *Toxicology* 328 (2015), 102–111.

[128] Hideki Horibe, Chikara Ueyama, Tetsuo Fujimaki, Mitsutoshi Oguri, Kimihiko Kato, Sahoko Ichihara, and Yoshiji Yamada. Association of a polymorphism of BTN2A1 with dyslipidemia in community-dwelling individuals. *Molecular medicine reports* 9.3 (2014), 808–812.

[129] Heiko Horn, Michael S Lawrence, Candace R Chouinard, Yashaswi Shrestha, Jessica Xin Hu, Elizabeth Worstell, Emily Shea, Nina Ilic, Eejung Kim, Atanas Kamburov, et al. NetSig: network-based discovery from cancer genomes. *Nature methods* 15.1 (2018), 61.

[130] Borislav H Hristov, Bernard Chazelle, and Mona Singh. uKIN Combines New and Prior Information with Guided Network Propagation to Accurately Identify Disease Genes. *Cell Systems* 10.6 (2020), 470 –479.e3.

[131] Da Wei Huang, Brad T Sherman, Qina Tan, Joseph Kir, David Liu, David Bryant, Yongjian Guo, Robert Stephens, Michael W Baseler, H Clifford Lane, et al. DAVID Bioinformatics Resources: expanded annotation database and novel algorithms to better extract biology from large gene lists. *Nucleic acids research* 35 (2007), W169–W175.

[132] Hui Huang, Eitan E Winter, Huajun Wang, Keith G Weinstock, Heming Xing, Leo Goodstadt, Peter D Stenson, David N Cooper, Douglas Smith, M Mar Albà, et al. Evolutionary conservation and selection of human disease gene orthologs in the rat and mouse genomes. *Genome biology* 5.7 (2004), R47.

[133] Justin K Huang, Daniel E Carlin, Michael Ku Yu, Wei Zhang, Jason F Kreisberg, Pablo Tamayo, and Trey Ideker. Systematic evaluation of molecular networks for discovery of disease genes. *Cell systems* 6.4 (2018), 484–495.

[134] Kent W Hunter, Ruhul Amin, Sarah Deasy, Ngoc-Han Ha, and Lalage Wakefield. Genetic insights into the morass of metastatic heterogeneity. *Nature Reviews Cancer* 18.4 (2018), 211.

[135] Carolyn Hutter and Jean Claude Zenklusen. The cancer genome atlas: creating lasting value beyond its data. *Cell* 173.2 (2018), 283–285.

[136] Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* 18 (2002), S233–S240.

[137] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences* 98.8 (2001), 4569–4574.

[138] Takashi Ito, Kosuke Tashiro, Shigeru Muta, Ritsuko Ozawa, Tomoko Chiba, Mayumi Nishizawa, Kiyoshi Yamamoto, Satoru Kuhara, and Yoshiyuki Sakaki. Toward a protein–protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proceedings of the National Academy of Sciences* 97.3 (2000), 1143–1147.

[139] Bijay Jassal, Lisa Matthews, Guilherme Viteri, Chuqiao Gong, Pascual Lorente, Antonio Fabregat, Konstantinos Sidiropoulos, Justin Cook, Marc Gillespie, Robin Haw, et al. The reactome pathway knowledgebase. *Nucleic acids research* 48.D1 (2020), D498–D503.

[140] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature* 411.6833 (2001), 41–42.

[141] Woojeong Jin, Jinhong Jung, and U Kang. Supervised and extended restart in random walks for ranking and link prediction in networks. *PLoS one* 14.3 (2019), 1–23.

[142] Atanas Kamburov, Konstantin Pentchev, Hanna Galicka, Christoph Wierling, Hans Lehrach, and Ralf Herwig. ConsensusPathDB: toward a more complete picture of cell biology. *Nucleic acids research* 39 (2011), D712–D717.

[143] Atanas Kamburov, Ulrich Stelzl, and Ralf Herwig. IntScore: a web tool for confidence scoring of biological interactions. *Nucleic Acids Research* 40.W1 (2012), W140–W146.

[144] Atanas Kamburov, Ulrich Stelzl, Hans Lehrach, and Ralf Herwig. The ConsensusPathDB interaction database: 2013 update. *Nucleic acids research* 41.D1 (2013), D793–D800.

[145] Atanas Kamburov, Christoph Wierling, Hans Lehrach, and Ralf Herwig. ConsensusPathDB—a database for integrating human functional interaction networks. *Nucleic acids research* 37 (2009), D623–D628.

[146] Minoru Kanehisa, Susumu Goto, Yoko Sato, Miho Furumichi, and Mao Tanabe. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic acids research* 40.D1 (2011), D109–D114.

[147] Maricel G Kann. Advances in translational bioinformatics: computational approaches for the hunting of disease genes. *Briefings in Bioinformatics* 11.1 (2009), 96–110.

[148] Niki Katsiki, Dimitri P Mikhailidis, and Maciej Banach. Leptin, cardiovascular diseases and type 2 diabetes mellitus. *Acta Pharmacologica Sinica* 39.7 (2018), 1176–1188.

[149] Thomas Kelder, Martijn P Van Iersel, Kristina Hanspers, Martina Kutmon, Bruce R Conklin, Chris T Evelo, and Alexander R Pico. WikiPathways: building research communities on biological pathways. *Nucleic acids research* 40.D1 (2012), D1301–D1307.

[150] Su Yeon Kim and Terence P Speed. Comparing somatic mutation-callers: beyond Venn diagrams. *BMC bioinformatics* 14.1 (2013), 189.

[151] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature physics* 6.11 (2010), 888–893.

[152] Kakajan Komurov, Michael A White, and Prahlad T Ram. Use of data-biased random walks on graphs for the retrieval of context-specific networks from genomic data. *PLoS Computational Biology* 6.8 (2010), e1000889.

[153] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. *Proceedings of the 19th international conference on machine learning*. Vol. 2002. 2002, 315–22.

[154] Tamás Korcsmáros, Illés J Farkas, Máté S Szalay, Petra Rovó, Dávid Fazekas, Zoltán Spiró, Csaba Böde, Katalin Lenti, Tibor Vellai, and Péter Csermely. Uniformly curated signaling pathways reveal tissue-specific cross-talks and support drug target discovery. *Bioinformatics* 26.16 (2010), 2042–2050.

[155] A Korn, A Schubert, and Andras Telcs. Lobby index in networks. *Physica A: Statistical Mechanics and its Applications* 388.11 (2009), 2221–2226.

[156] Martin Krallinger, Alfonso Valencia, and Lynette Hirschman. Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome biology* 9.2 (2008), 1–14.

[157] Lars Kuepfer, Olivia Clayton, Christoph Thiel, Henrik Cordes, Ramona Nudischer, Lars M Blank, Vanessa Baier, Stephane Heymans, Florian Caiment, Adrian Roth, et al. A model-based assay design to reproduce in vivo patterns of acute drug-induced toxicity. *Archives of Toxicology* 92.1 (2018), 553–555.

[158] Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N. Robinson. Walking the Interactome for Prioritization of Candidate Disease Genes. *The American Journal of Human Genetics* 82.4 (2008), 949 –958.

[159] Kasper Lage, E Olof Karlberg, Zenia M Størling, Páll I Olason, Anders G Pedersen, Olga Rigina, Anders M Hinsby, Zeynep Tümer, Flemming Pociot, Niels Tommerup, et al. A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature biotechnology* 25.3 (2007), 309–316.

[160] Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, et al. Initial sequencing and analysis of the human genome (2001).

[161] Michael S Lawrence, Petar Stojanov, Craig H Mermel, James T Robinson, Levi A Garraway, Todd R Golub, Matthew Meyerson, Stacey B Gabriel, Eric S Lander, and Gad Getz. Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature* 505.7484 (2014), 495–501.

[162] Michael S Lawrence, Petar Stojanov, Paz Polak, Gregory V Kryukov, Kristian Cibulskis, Andrey Sivachenko, Scott L Carter, Chip Stewart, Craig H Mermel, Steven A Roberts, et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature* 499.7457 (2013), 214–218.

[163] Insuk Lee, U Martin Blom, Peggy I Wang, Jung Eun Shim, and Edward M Marcotte. Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome research* 21.7 (2011), 1109–1121.

[164] Sangseon Lee, Sangsoo Lim, Taeheon Lee, Inyoung Sung, and Sun Kim. Cancer subtype classification and modeling by pathway attention and propagation. *Bioinformatics* (2020).

[165] Ben Lehner and Andrew G Fraser. A first-draft human protein-interaction map. *Genome biology* 5.9 (2004), R63.

[166] Mark D M Leiserson, Fabio Vandin, Hsin-Ta Wu, Jason R Dobson, Jonathan V Eldridge, Jacob L Thomas, Alexandra Papoutsaki, Younhun Kim, Beifang Niu, Michael McLellan, et al. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nature Genetics* 47.2 (2015), 106–114.

[167] Carrie G Lenneman and Douglas B Sawyer. Cardio-oncology: an update on cardiotoxicity of cancer-related treatment. *Circulation research* 118.6 (2016), 1008–1020.

[168] Yongjin Li and Jagdish C. Patra. Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics* 26.9 (2010), 1219–1224.

[169] Jens G Lohr, Petar Stojanov, Michael S Lawrence, Daniel Auclair, Bjoern Chapuy, Carrie Sougnez, Peter Cruz-Gordillo, Birgit Knoechel, Yan W Asmann, Susan L Slager, et al. Discovery and prioritization of somatic mutations in diffuse large B-cell lymphoma (DLBCL) by whole-exome sequencing. *Proceedings of the National Academy of Sciences* 109.10 (2012), 3879–3884.

[170] László Lovász et al. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty* 2.1 (1993), 1–46.

[171] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology* 15.12 (2014), 550.

[172] Linyuan Lü, Tao Zhou, Qian-Ming Zhang, and H Eugene Stanley. The H-index of a network node and its relation to degree and coreness. *Nature communications* 7 (2016), 10168.

[173] Weijun Luo, Gaurav Pant, Yeshvant K Bhavnasi, Steven G Blanchard Jr, and Cory Brouwer. Pathview Web: user friendly pathway visualization and data integration. *Nucleic acids research* 45.W1 (2017), W501–W508.

[174] Núria López-Bigas and Christos A. Ouzounis. Genome-wide identification of genes likely to be involved in human genetic disease. *Nucleic Acids Research* 32.10 (2004), 3108–3114.

[175] Hongwu Ma, Anatoly Sorokin, Alexander Mazein, Alex Selkov, Evgeni Selkov, Oleg Demin, and Igor Goryanin. The Edinburgh human metabolic network reconstruction and its functional analysis. *Molecular systems biology* 3.1 (2007), 135.

[176] Jacqueline MacArthur, Emily Bowler, Maria Cerezo, Laurent Gil, Peggy Hall, Emma Hastings, Heather Junkins, Aoife McMahon, Annalisa Milano, Joannella Morales, et al. The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). *Nucleic acids research* 45.D1 (2017), D896–D901.

[177] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. *ACM SIGCOMM Computer Communication Review* 36.4 (2006), 135–146.

[178] Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports* 533.4 (2013), 95 – 142.

[179] Matthias Mann, Ronald C Hendrickson, and Akhilesh Pandey. Analysis of Proteins and Proteomes by Mass Spectrometry. *Annual Review of Biochemistry* 70.1 (2001), 437–473.

[180] Teri A Manolio, Francis S Collins, Nancy J Cox, David B Goldstein, Lucia A Hindorff, David J Hunter, Mark I McCarthy, Erin M Ramos, Lon R Cardon, Aravinda Chakravarti, et al. Finding the missing heritability of complex diseases. *Nature* 461.7265 (2009), 747–753.

[181] Edward M Marcotte, Ioannis Xenarios, and David Eisenberg. Mining literature for protein–protein interactions. *Bioinformatics* 17.4 (2001), 359–363.

[182] Marcel Margulies, Michael Egholm, William E Altman, Said Attiya, Joel S Bader, Lisa A Bemben, Jan Berka, Michael S Braverman, Yi-Ju Chen, Zhoutao Chen, et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437.7057 (2005), 376–380.

[183] John C Marioni, Christopher E Mason, Shrikant M Mane, Matthew Stephens, and Yoav Gilad. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research* 18.9 (2008), 1509–1517.

[184] Joseph A. Marsh and Sarah A. Teichmann. Structure, Dynamics, Assembly, and Evolution of Protein Complexes. *Annual Review of Biochemistry* 84.1 (2015), 551–575.

[185] Marco Masseroli, Dario Martucci, and Francesco Pinciroli. GFINDer: Genome Function INtegrated Discoverer through dynamic annotation, statistical analysis, and mining. *Nucleic acids research* 32 (2004), W293–W300.

[186] Lisa R Matthews, Philippe Vaglio, Jérôme Reboul, Hui Ge, Brian P Davis, James Garrels, Sylvie Vincent, and Marc Vidal. Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or "interologs". *Genome research* 11.12 (2001), 2120–2126.

[187] Arnon Mazza, Konrad Klockmeier, Erich Wanker, and Roded Sharan. An integer programming framework for inferring disease complexes from network data. *Bioinformatics* 32.12 (2016), i271–i277.

[188] Patrick McGillivray, Declan Clarke, William Meyerson, Jing Zhang, Donghoon Lee, Mengting Gu, Sushant Kumar, Holly Zhou, and Mark Gerstein. Network Analysis as a Grand Unifier in Biomedical Data Science. *Annual Review of Biomedical Data Science* 1.1 (2018), 153–180.

[189] John V McGowan, Robin Chung, Angshuman Maulik, Izabela Piotrowska, J Malcolm Walker, and Derek M Yellon. Anthracycline chemotherapy and cardiotoxicity. *Cardiovascular drugs and therapy* 31.1 (2017), 63–75.

[190] Nan Mei, James C Fuscoe, Edward K Lobenhofer, and Lei Guo. Application of microarray-based analysis of gene expression in the field of toxicogenomics. *Rat Genomics*. Springer, 2010, 227–241.

[191] Jörg Menche, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. Uncovering disease-disease relationships through the incomplete interactome. *Science* 347.6224 (2015), 1257601.

[192] Matthew Meyerson, Stacey Gabriel, and Gad Getz. Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews Genetics* 11.10 (2010), 685–696.

[193] Ron Milo, Nadav Kashtan, Shalev Itzkovitz, Mark EJ Newman, and Uri Alon. On the uniform generation of random graphs with prescribed degree sequences. *arXiv* 0312028 (2003).

[194] Koyel Mitra, Anne-Ruxandra Carvunis, Sanath Kumar Ramesh, and Trey Ideker. Integrative approaches for finding modular structure in biological networks. *Nature Reviews Genetics* 14.10 (2013), 719–732.

[195] Shayan Moazeni, Martin Cadeiras, Eric H Yang, Mario C Deng, and Kim-Lien Nguyen. Anthracycline induced cardiotoxicity: biomarkers and "Omics" technology in the era of patient specific care. *Clinical and translational medicine* 6.1 (2017), 17.

[196] Yves Moreau and Léon-Charles Tranchevent. Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Reviews Genetics* 13.8 (2012), 523–536.

[197] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods* 5.7 (2008), 621–628.

[198] Ralf Mrowka, Andreas Patzak, and Hanspeter Herzel. Is there a bias in proteome research? *Genome research* 11.12 (2001), 1971–1973.

[199] Elena Nabieva, Kam Jim, Amit Agarwal, Bernard Chazelle, and Mona Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics* 21 (2005), i302–i310.

[200] H Nakagawa, CP Wardell, M Furuta, H Taniguchi, and A Fujimoto. Cancer whole-genome sequencing: present and future. *Oncogene* 34.49 (2015), 5943–5950.

[201] Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics* 26.8 (2010), 1057–1063.

[202] Mark EJ Newman. *Networks: an introduction*. Oxford; New York: Oxford University Press, 2010.

[203] Mark EJ Newman. Communities, modules and large-scale structure in networks. *Nature physics* 8.1 (2012), 25–31.

[204] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E* 69.2 (2004), 026113.

[205] Pauline C Ng and Steven Henikoff. SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research* 31.13 (2003), 3812–3814.

[206] Sarah B Ng, Kati J Buckingham, Choli Lee, Abigail W Bigham, Holly K Tabor, Karin M Dent, Chad D Huff, Paul T Shannon, Ethylin Wang Jabs, Deborah A Nickerson, et al. Exome sequencing identifies the cause of a mendelian disorder. *Nature genetics* 42.1 (2010), 30.

[207] Rasmus Nielsen, Joshua S Paul, Anders Albrechtsen, and Yun S Song. Genotype and SNP calling from next-generation sequencing data. *Nature Reviews Genetics* 12.6 (2011), 443–451.

[208] William S Noble, Rui Kuang, Christina Leslie, and Jason Weston. Identifying remote protein homologs by network propagation. *The FEBS Journal* 272.20 (2005), 5119–5128.

[209] Jae Dong Noh. Exact scaling properties of a hierarchical network model. *Phys. Rev. E* 67 (4 2003), 045103.

[210] David T Okou, Karyn Meltz Steinberg, Christina Middle, David J Cutler, Thomas J Albert, and Michael E Zwick. Microarray-based genomic selection for high-throughput resequencing. *Nature methods* 4.11 (2007), 907–909.

[211] Stephen Oliver. Guilt-by-association goes global. *Nature* 403.6770 (2000), 601–602.

[212] Chiara Orsini, Marija M Dankulov, Pol Colomer-de Simón, Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Kevin E Bassler, Zoltán Toroczkai, Marián Boguñá, Guido Caldarelli, et al. Quantifying randomness in real networks. *Nature communications* 6.1 (2015), 1–10.

[213] Ali Oskooei, Jannis Born, Matteo Manica, Vigneshwari Subramanian, Julio Sáez-Rodríguez, and María Rodríguez Martínez. PaccMann: Prediction of anticancer compound sensitivity with multi-modal attention-based neural networks. *arXiv* 1811.06802 (2018).

[214] Martin Oti and Han G Brunner. The modular nature of genetic diseases. *Clinical genetics* 71.1 (2007), 1–11.

[215] Martin Oti, Jeroen van Reeuwijk, Martijn A Huynen, and Han G Brunner. Conserved co-expression for candidate disease gene prioritization. *BMC bioinformatics* 9.1 (2008), 208.

[216] Martin Oti, Berend Snel, Martijn A Huynen, and Han G Brunner. Predicting disease genes using protein–protein interactions. *Journal of medical genetics* 43.8 (2006), 691–698.

[217] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab, 1999.

[218] Philipp Pagel, Stefan Kovac, Matthias Oesterheld, Barbara Brauner, Irmtraud Dunger-Kaltenbach, Goar Frishman, Corinna Montrone, Pekka Mark, Volker Stümpflen, Hans-Werner Mewes, et al. The MIPS mammalian protein-protein interaction database. *Bioinformatics* 21.6 (2005), 832–834.

[219] Antonio F Pardiñas, Peter Holmans, Andrew J Pocklington, Valentina Escott-Price, Stephan Ripke, Noa Carrera, Sophie E Legge, Sophie Bishop, Darren Cameron, Marian L Hamshere, et al. Common schizophrenia alleles are enriched in mutation-intolerant genes and in regions under strong background selection. *Nature genetics* 50.3 (2018), 381–389.

[220] Peter J Park. ChIP–seq: advantages and challenges of a maturing technology. *Nature reviews genetics* 10.10 (2009), 669–680.

[221] Jodi R Parrish, Keith D Gulyas, and Russell L Finley. Yeast two-hybrid contributions to interactome mapping. *Current Opinion in Biotechnology* 17.4 (2006), 387 –393.

[222] Richard Paules. Phenotypic anchoring: linking cause and effect. *Environmental Health Perspectives* 111.6 (2003), A338–A339.

[223] Evan O Paull, Daniel E Carlin, Mario Niepel, Peter K Sorger, David Haussler, and Joshua M Stuart. Discovering causal pathways linking genomic events to transcriptional states using Tied Diffusion Through Interacting Events (TieDIE). *Bioinformatics* 29.21 (2013), 2757–2764.

[224] Ana B Pavel, Dmitriy Sonkin, and Anupama Reddy. Integrative modeling of multi-omics data to identify cancer drivers and infer patient-specific gene activity. *BMC systems biology* 10.1 (2016), 16.

[225] A Caviani Pease, Dennis Solas, Edward J Sullivan, Maureen T Cronin, Christopher P Holmes, and SP Fodor. Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proceedings of the National Academy of Sciences* 91.11 (1994), 5022–5026.

[226] Wei Peng, Min Li, Lu Chen, and Lusheng Wang. Predicting protein functions by using unbalanced random walk algorithm on three biological networks. *IEEE/ACM transactions on computational biology and bioinformatics* 14.2 (2015), 360–369.

[227] Xiaoqing Peng, Jianxin Wang, Wei Peng, Fang-Xiang Wu, and Yi Pan. Protein–protein interactions: detection, reliability assessment and applications. *Briefings in Bioinformatics* 18.5 (2016), 798–819.

[228] A Petitjean, MIW Achatz, AL Borresen-Dale, P Hainaut, and M Olivier. TP53 mutations in human cancers: functional selection and impact on cancer prognosis and outcomes. *Oncogene* 26.15 (2007), 2157–2165.

[229] Sergio Picart-Armada, Wesley K Thompson, Alfonso Buil, and Alexandre Perera-Lluna. The effect of statistical normalisation on network propagation scores. *BioRxiv* (2020).

[230] Janet Piñero, Núria Queralt-Rosinach, Àlex Bravo, Jordi Deu-Pons, Anna Bauer-Mehren, Martin Baron, Ferran Sanz, and Laura I. Furlong. DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database* 2015 (2015).

[231] Janet Piñero, Juan Manuel Ramírez-Anguita, Josep Saüch-Pitarch, Francesco Ronzano, Emilio Centeno, Ferran Sanz, and Laura I Furlong. The DisGeNET knowledge platform for disease genomics: 2019 update. *Nucleic acids research* 48.D1 (2020), D845–D855.

[232] Rosario M Piro and Ferdinando Di Cunto. Computational approaches to disease-gene prediction: rationale, classification and successes. *The FEBS Journal* 279.5 (2012), 678–696.

[233] Graham M Pitcher, Lorraine V Kalia, David Ng, Nathalie M Goodfellow, Kathleen T Yee, Evelyn K Lambe, and Michael W Salter. Schizophrenia susceptibility pathway neuregulin 1–ErbB4 suppresses Src upregulation of NMDA receptors. *Nature medicine* 17.4 (2011), 470.

[234] Dexter Pratt, Jing Chen, David Welker, Ricardo Rivas, Rudolf Pillich, Vladimir Rynkov, Keiichiro Ono, Carol Miello, Lyndon Hicks, Sandor Szalma, et al. NDEx, the network data exchange. *Cell systems* 1.4 (2015), 302–305.

[235] N. Pržulj, D.A. Wigle, and I. Jurisica. Functional topology in a network of protein interactions. *Bioinformatics* 20.3 (2004), 340–348.

[236] Yu-Qing Qiu, Shihua Zhang, Xiang-Sun Zhang, and Luonan Chen. Detecting disease associated modules and prioritizing active genes based on high throughput data. *BMC bioinformatics* 11.1 (2010), 26.

[237] Franck Rapaport, Raya Khanin, Yupu Liang, Mono Pirun, Azra Krek, Paul Zumbo, Christopher E Mason, Nicholas D Socci, and Doron Betel. Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome biology* 14.9 (2013), 1–13.

[238] Benjamin J Raphael, Jason R Dobson, Layla Oesper, and Fabio Vandin. Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine. *Genome medicine* 6.1 (2014), 5.

[239] Emanuel Raschi, Valentina Vasina, Maria Grazia Ursino, Giuseppe Boriani, Andrea Martoni, and Fabrizio De Ponti. Anticancer drugs and cardiotoxicity: insights and perspectives in the era of targeted therapy. *Pharmacology & therapeutics* 125.2 (2010), 196–218.

[240] Dimitra Repana, Joel Nulsen, Lisa Dressler, Michele Bortolomeazzi, Santhilata Kuppili Venkata, Aikaterini Tourna, Anna Yakovleva, Tommaso Palmieri, and Francesca D Ciccarelli. The Network of Cancer Genes (NCG): a comprehensive catalogue of known and candidate cancer genes from cancer sequencing screens. *Genome biology* 20.1 (2019), 1.

[241] Matthew A Reyna, Uthsav Chitra, Rebecca Elyanow, and Benjamin J Raphael. NetMix: A network-structured mixture model for reduced-bias estimation of altered subnetworks. *bioRxiv* (2020).

[242] Matthew A Reyna, Mark D M Leiserson, and Benjamin J Raphael. Hierarchical HotNet: identifying hierarchies of altered subnetworks. *Bioinformatics* 34.17 (2018), i972–i980.

[243] Stephan Ripke, Alan R Sanders, Kenneth S Kendler, Douglas F Levinson, Pamela Sklar, Peter A Holmans, Dan-Yu Lin, Jubao Duan, Roel A Ophoff, Ole A Andreassen, et al. Genome-wide association study identifies five new schizophrenia loci. *Nature genetics* 43.10 (2011), 969.

[244] Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids research* 43.7 (2015), e47–e47.

[245] Alexander W Rives and Timothy Galitski. Modular organization of cellular networks. *Proceedings of the national Academy of sciences* 100.3 (2003), 1128–1133.

[246] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26.1 (2010), 139–140.

[247] Igor Rodchenkov, Ozgun Babur, Augustin Luna, Bulent Arman Aksoy, Jeffrey V Wong, Dylan Fong, Max Franz, Metin Can Siper, Manfred Cheung, Michael Wrana, et al. Pathway Commons 2019 Update: integration, analysis and exploration of pathway data. *Nucleic Acids Research* 48.D1 (2019), D489–D497.

[248] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, et al. A Proteome-Scale Map of the Human Interactome Network. *Cell* 159.5 (2014), 1212 –1226.

[249] Pedro Romero, Jonathan Wagg, Michelle L Green, Dale Kaiser, Markus Krummenacker, and Peter D Karp. Computational prediction of human metabolic pathways from the complete human genome. *Genome biology* 6.1 (2005), R2.

[250] Jean-François Rual, Kavitha Venkatesan, Tong Hao, Tomoko Hirozane-Kishikawa, Amélie Dricot, Ning Li, Gabriel F Berriz, Francis D Gibbons, Matija Dreze, Nono Ayivi-Guedehoussou, et al. Towards a proteome-scale map of the human protein–protein interaction network. *Nature* 437.7062 (2005), 1173–1178.

[251] Matthew Ruffalo, Mehmet Koyutürk, and Roded Sharan. Network-based integration of disparate omic data to identify "silent players" in cancer. *PLoS Computational Biology* 11.12 (2015).

[252] Michael W Salter and Graham M Pitcher. Dysregulated Src upregulation of NMDA receptor activity: a common link in chronic pain and schizophrenia. *The FEBS journal* 279.1 (2012), 2–11.

[253] Lukasz Salwinski, Christopher S Miller, Adam J Smith, Frank K Pettit, James U Bowie, and David Eisenberg. The Database of Interacting Proteins: 2004 update. *Nucleic Acids Research* 32 (2004), D449–D451.

[254] Christopher M Sanderson. The Cartographers toolbox: building bigger and better human protein interaction networks. *Briefings in Functional Genomics* 8.1 (2009), 1–11.

[255] Frederick Sanger, Steven Nicklen, and Alan R Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences* 74.12 (1977), 5463–5467.

[256] Eric E Schadt, Steve Turner, and Andrew Kasarskis. A window into third- generation sequencing. *Human molecular genetics* 19.R2 (2010), R227–R240.

[257] Martin H Schaefer, Luis Serrano, and Miguel A Andrade-Navarro. Correcting for the study bias associated with protein–protein interaction measurements reveals differences between protein degree distributions from different cancer types. *Frontiers in Genetics* 6 (2015), 260.

[258] Michael C Schatz, Arthur L Delcher, and Steven L Salzberg. Assembly of large genomes using second-generation sequencing. *Genome research* 20.9 (2010), 1165–1173.

[259] Harald Jörn Schneider, Nele Friedrich, Jens Klotsche, Sabine Schipf, Matthias Nauck, Henry Völzke, Caroline Sievers, Lars Pieper, Winfried März, Hans-Ulrich Wittchen, et al. Prediction of incident diabetes mellitus by baseline IGF1 levels. *European Journal of Endocrinology* 164.2 (2011), 223 –229.

[260] Benno Schwikowski, Peter Uetz, and Stanley Fields. A network of protein–protein interactions in yeast. *Nature biotechnology* 18.12 (2000), 1257–1261.

[261] Fritz J Sedlazeck, Hayan Lee, Charlotte A Darby, and Michael C Schatz. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nature Reviews Genetics* 19.6 (2018), 329–346.

[262] Dominik Seelow, Jana Marie Schwarz, and Markus Schuelke. GeneDistiller — distilling candidate genes from linkage intervals. *PLoS one* 3.12 (2008).

[263] Nathalie Selevsek, Florian Caiment, Ramona Nudischer, Hans Gmuender, Irina Agarkova, Francis L Atkinson, Ivo Bachmann, Vanessa Baier, Gal Barel, Chris Bauer, et al. Network integration and modelling of dynamic drug responses at multi-omics levels. *Communications biology* 3.1 (2020), 1–15.

[264] Katsunori Senda, Takanori Goi, Yasuo Hirono, Kanji Katayama, and Akio Yamaguchi. Analysis of RIN1 gene expression in colorectal cancer. *Oncology reports* 17.5 (2007), 1171–1175.

[265] Ku Chee Seng and Chia Kee Seng. The success of the genome-wide association approach: a brief story of a long struggle. *European Journal of Human Genetics* 16.5 (2008), 554–564.

[266] Dari Shalon, Stephen J Smith, and Patrick O Brown. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome research* 6.7 (1996), 639–645.

[267] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research* 13.11 (2003), 2498–2504.

[268] R Sherrington, EI Rogaev, Y al Liang, EA Rogaeva, G Levesque, M Ikeda, H Chi, C Lin, G Li, K Holman, et al. Cloning of a gene bearing missense mutations in early-onset familial Alzheimer's disease. *Nature* 375.6534 (1995), 754–760.

[269] U Martin Singh-Blom, Nagarajan Natarajan, Ambuj Tewari, John O Woods, Inderjit S Dhillon, and Edward M Marcotte. Prediction and validation of gene-disease associations using methods inspired by social network analyses. *PLoS one* 8.5 (2013).

[270] Damian Smedley, Sebastian Köhler, Johanna Christina Czeschik, Joanna Amberger, Carol Bocchini, Ada Hamosh, Julian Veldboer, Tomasz Zemojtel, and Peter N. Robinson. Walking the interactome for candidate prioritization in exome sequencing studies of Mendelian diseases. *Bioinformatics* 30.22 (2014), 3215–3222.

[271] Nick GC Smith and Adam Eyre-Walker. Human disease genes: patterns and predictions. *Gene* 318 (2003), 169–175.

[272] Donny Soh, Difeng Dong, Yike Guo, and Limsoon Wong. Consistency, comprehensiveness, and compatibility of pathway databases. *BMC bioinformatics* 11.1 (2010), 449.

[273] Zbyslaw Sondka, Sally Bamford, Charlotte G Cole, Sari A Ward, Ian Dunham, and Simon A Forbes. The COSMIC Cancer Gene Census: describing genetic dysfunction across all human cancers. *Nature Reviews Cancer* 18.11 (2018), 696–705.

[274] David A Stead, Norman W Paton, Paolo Missier, Suzanne M Embury, Cornelia Hedeler, Binling Jin, Alistair JP Brown, and Alun Preece. Information quality in proteomics. *Briefings in Bioinformatics* 9.2 (2008), 174–188.

[275] Laurel J Steinherz, Peter G Steinherz, Charlotte TC Tan, Glenn Heller, and M Lois Murphy. Cardiac toxicity 4 to 20 years after completing anthracycline therapy. *Jama* 266.12 (1991), 1672–1677.

[276] Gil Stelzer, Naomi Rosen, Inbar Plaschkes, Shahar Zimmerman, Michal Twik, Simon Fishilevich, Tsippi Iny Stein, Ron Nudel, Iris Lieder, Yaron Mazor, et al. The GeneCards suite: from gene data mining to disease genome sequence analyses. *Current protocols in bioinformatics* 54.1 (2016), 1–30.

[277] Ulrich Stelzl, Uwe Worm, Maciej Lalowski, Christian Haenig, Felix H Brembeck, Heike Goehler, Martin Stroedicke, Martina Zenkner, Anke Schoenherr, Susanne Koeppen, et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell* 122.6 (2005), 957–968.

[278] Karin B. Stibius and Kim Sneppen. Modeling the Two-Hybrid Detector: Experimental Bias on Protein Interaction Networks. *Biophysical Journal* 93.7 (2007), 2562 –2566.

[279] Michael R Stratton, Peter J Campbell, and P Andrew Futreal. The cancer genome. *Nature* 458.7239 (2009), 719–724.

[280] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* 102.43 (2005), 15545–15550.

[281] Chen Suo, Olga Hrydziuszko, Donghwan Lee, Setia Pramana, Dhany Saputra, Himanshu Joshi, Stefano Calza, and Yudi Pawitan. Integration of somatic mutation, expression and functional data reveals potential driver genes predictive of breast cancer survival. *Bioinformatics* 31.16 (2015), 2607–2613.

[282] Máté Szalay-Bekő, Robin Palotai, Balázs Szappanos, István A Kovács, Balázs Papp, and Péter Csermely. ModuLand plug-in for Cytoscape: determination of hierarchical layers of overlapping network modules and community centrality. *Bioinformatics* 28.16 (2012), 2202–2204.

[283] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research* 47.D1 (2019), D607–D613.

[284] Sarah A Gagliano Taliun, Peter VandeHaar, Andrew P Boughton, Ryan P Welch, Daniel Taliun, Ellen M Schmidt, Wei Zhou, Jonas B Nielsen, Cristen J Willer, Seunggeun Lee, et al. Exploring and visualizing large-scale genetic associations by using PheWeb. *Nature Genetics* 52.6 (2020), 550–552.

[285] Vivian Tam, Nikunj Patel, Michelle Turcotte, Yohan Bossé, Guillaume Paré, and David Meyre. Benefits and limitations of genome-wide association studies. *Nature Reviews Genetics* 20.8 (2019), 467–484.

[286] ICGC The, TCGA Pan-Cancer Analysis of Whole, Genomes Consortium, et al. Pan-cancer analysis of whole genomes. *Nature* 578.7793 (2020), 82.

[287] Rui Tian, Malay K Basu, and Emidio Capriotti. Computational methods and resources for the interpretation of genomic variants in cancer. *BMC genomics* 16.S8 (2015), S7.

[288] Collin J Tokheim, Nickolas Papadopoulos, Kenneth W Kinzler, Bert Vogelstein, and Rachel Karchin. Evaluating the evaluation of cancer driver genes. *Proceedings of the National Academy of Sciences* 113.50 (2016), 14330–14335.

[289] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Random walk with restart: fast solutions and applications. *Knowledge and Information Systems* 14.3 (2008), 327–346.

[290] Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols* 7.3 (2012), 562–578.

[291] Judy Truong, Andrew T Yan, Gemma Cramarossa, and Kelvin KW Chan. Chemotherapy-induced cardiotoxicity: detection, prevention, and management. *Canadian Journal of Cardiology* 30.8 (2014), 869–878.

[292] E Tzolos, PD Adamson, PS Hall, IR Macpherson, O Oikonomidou, M MacLean, SC Lewis, H McVicars, DE Newby, NL Mills, et al. Dynamic changes in high-sensitivity cardiac troponin I in response to anthracycline-based chemotherapy. *Clinical Oncology* 32.5 (2020), 292–297.

[293] Peter Uetz, Loic Giot, Gerard Cagney, Traci A Mansfield, Richard S Judson, James R Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, et al. A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae. *Nature* 403.6770 (2000), 623–627.

[294] Steven Van Vooren, Bernard Thienpont, Björn Menten, Frank Speleman, Bart De Moor, Joris Vermeesch, and Yves Moreau. Mapping biomedical concepts onto the human genome by mining literature on chromosomal aberrations. *Nucleic acids research* 35.8 (2007), 2533–2543.

[295] Fabio Vandin, Patrick Clay, Eli Upfal, and Benjamin J Raphael. Discovery of mutated subnetworks associated with clinical data in cancer. *Biocomputing 2012*. World Scientific, 2012, 55–66.

[296] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Computational Biology* 6.1 (2010).

[297] Oron Vanunu and Roded Sharan. A Propagation-based Algorithm for Inferring Gene-Disease Associations. *German Conference on Bioinformatics*. Gesellschaft für Informatik e. V., 2008, 54–63.

[298] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science* 291.5507 (2001), 1304–1351.

[299] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* 17.3 (2020), 261–272.

[300] Bert Vogelstein and Kenneth W Kinzler. Cancer genes and the pathways they control. *Nature medicine* 10.8 (2004), 789–799.

[301] Christian Von Mering, Lars J Jensen, Berend Snel, Sean D Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A Huynen, and Peer Bork. STRING: known and predicted protein–protein associations, integrated and transferred across organisms. *Nucleic acids research* 33 (2005), D433–D437.

[302] Christian Von Mering, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* 417.6887 (2002), 399–403.

[303] Albertha JM Walhout, Simon J Boulton, and Marc Vidal. Yeast two-hybrid systems and protein interaction mapping projects for yeast and worm. *International Journal of Genomics* 1.2 (2000), 88–94.

[304] Albertha JM Walhout, Raffaella Sordella, Xiaowei Lu, James L Hartley, Gary F Temple, Michael A Brasch, Nicolas Thierry-Mieg, and Marc Vidal. Protein interaction mapping in C. elegans using proteins involved in vulval development. *Science* 287.5450 (2000), 116–122.

[305] Albertha JM Walhout and Marc Vidal. High-throughput yeast two-hybrid assays for large-scale protein interaction mapping. *Methods* 24.3 (2001), 297–306.

[306] Bingxuan Wang, P Charukeshi Chandrasekera, and John J Pippin. Leptin-and leptin receptor-deficient rodent models: relevance for human type 2 diabetes. *Current diabetes reviews* 10.2 (2014), 131–145.

[307] Qingguo Wang, Peilin Jia, Fei Li, Haiquan Chen, Hongbin Ji, Donald Hucks, Kimberly Brown Dahlman, William Pao, and Zhongming Zhao. Detecting somatic point mutations in cancer genome sequencing data: a comparison of mutation callers. *Genome medicine* 5.10 (2013), 91.

[308] Qun Wang, Ying Gao, Yufu Tang, Lie Ma, Mingjing Zhao, and Xiaoge Wang. Prognostic significance of RIN1 gene expression in human non-small cell lung cancer. *Acta histochemica* 114.5 (2012), 463–468.

[309] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics* 10.1 (2009), 57–63.

[310] Katelyn R Ward, Robert E Featherstone, Melissa J Naschek, Olga Melnychenko, Anamika Banerjee, Janice Yi, Raymond L Gifford, Karin E Borgmann-Winter, Michael W Salter, Chang-Gyu Hahn, et al. Src deficient mice demonstrate behavioral and electrophysiological alterations relevant to psychiatric and developmental disease. *Progress in Neuro-Psychopharmacology and Biological Psychiatry* 93 (2019), 84–92.

[311] Stanley Wasserman, Katherine Faust, et al. *Social network analysis: Methods and applications*. Vol. 8. Cambridge university press, 1994.

[312] James D Watson and Francis HC Crick. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature* 171.4356 (1953), 737–738.

[313] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature* 393.6684 (1998), 440.

[314] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics* 45.10 (2013), 1113.

[315] Gezhi Weng, Upinder S. Bhalla, and Ravi Iyengar. Complexity in Biological Signaling Systems. *Science* 284.5411 (1999), 92–96.

[316] David A Wheeler, Maithreyan Srinivasan, Michael Egholm, Yufeng Shen, Lei Chen, Amy McGuire, Wen He, Yi-Ju Chen, Vinod Makhijani, G Thomas Roth, et al. The complete genome of an individual by massively parallel DNA sequencing. *Nature* 452.7189 (2008), 872–876.

[317] David A Wheeler and Linghua Wang. From human genome to cancer genome: the first decade. *Genome research* 23.7 (2013), 1054–1062.

[318] Michelle Whirl-Carrillo, Ellen M McDonagh, JM Hebert, Li Gong, K Sangkuhl, CF Thorn, Russ B Altman, and Teri E Klein. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics* 92.4 (2012), 414–417.

[319] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research* 46.D1 (2018), D1074–D1082.

[320] Xuebing Wu, Rui Jiang, Michael Q Zhang, and Shao Li. Network-based global inference of human disease genes. *Molecular Systems Biology* 4.1 (2008), 189.

[321] Zikai Wu, Xingming Zhao, and Luonan Chen. Identifying responsive functional modules from protein-protein interaction network. *Molecules and cells* 27.3 (2009), 271–277.

[322] Ioannis Xenarios, Danny W Rice, Lukasz Salwinski, Marisa K Baron, Edward M Marcotte, and David Eisenberg. DIP: the database of interacting proteins. *Nucleic acids research* 28.1 (2000), 289–291.

[323] Jianzhen Xu and Yongjin Li. Discovering disease-genes by topological features in human protein–protein interaction network. *Bioinformatics* 22.22 (2006), 2800–2805.

[324] Yaping Yang, Donna M Muzny, Jeffrey G Reid, Matthew N Bainbridge, Alecia Willis, Patricia A Ward, Alicia Braxton, Joke Beuten, Fan Xia, Zhiyv Niu, et al. Clinical Whole-Exome Sequencing for the Diagnosis of Mendelian Disorders. *New England Journal of Medicine* 369.16 (2013), 1502–1511.

[325] Soon-Hyung Yook, Zoltán N Oltvai, and Albert-László Barabási. Functional and topological characterization of protein interaction networks. *Proteomics* 4.4 (2004), 928–942.

[326] Matthew D Young, Davis J McCarthy, Matthew J Wakefield, Gordon K Smyth, Alicia Oshlack, and Mark D Robinson. Differential expression for RNA sequencing (RNA-Seq) data: mapping, summarization, statistical analysis, and experimental design. *Bioinformatics for high throughput sequencing*. Springer, 2012, 169–190.

[327] Haiyuan Yu, Pascal Braun, Muhammed A Yıldırım, Irma Lemmens, Kavitha Venkatesan, Julie Sahalie, Tomoko Hirozane-Kishikawa, Fana Gebreab, Na Li, Nicolas Simonis, et al. High-quality binary protein interaction map of the yeast interactome network. *Science* 322.5898 (2008), 104–110.

[328] Wei Yu, Anja Wulf, Tiebin Liu, Muin J Khoury, and Marta Gwinn. Gene Prospector: an evidence gateway for evaluating potential susceptibility genes and interacting risk factors for human diseases. *BMC bioinformatics* 9.1 (2008), 528.

[329] Travis I Zack, Steven E Schumacher, Scott L Carter, Andrew D Cherniack, Gordon Saksena, Barbara Tabak, Michael S Lawrence, Cheng-Zhong Zhang, Jeremiah Wala, Craig H Mermel, et al. Pan-cancer patterns of somatic copy number alteration. *Nature genetics* 45.10 (2013), 1134–1140.

[330] Andreas Zanzoni, Luisa Montecchi-Palazzi, Michele Quondam, Gabriele Ausiello, Manuela Helmer-Citterich, and Gianni Cesareni. MINT: a Molecular INTeraction database. *FEBS Letters* 513.1 (2002), 135–140.

[331] Lulu Zhang, Yingfen Qin, Danyan Liang, Li Li, Yaojie Liang, Lulin Chen, Lei Tong, Jia Zhou, Hong Li, and Haiying Zhang. Association of polymorphisms in LEPR with type 2 diabetes and related metabolic traits in a Chinese population. *Lipids in health and disease* 17.1 (2018), 2.

[332] Wei Zhang, Jianzhu Ma, and Trey Ideker. Classifying tumors by supervised network propagation. *Bioinformatics* 34.13 (2018), i484–i493.

Summary

In recent years, the framework of network propagation has been adopted multiple times for the purpose of generating novel genotype-phenotype associations. However, existing methods usually rely on the standard degree-based formulation, which skews the results due to degree bias within protein-protein interaction networks. Furthermore, the network modules, which are identified post propagation by some of these methods, are rather dispersed and their genes are not well connected.

In this thesis we present NetCore, a novel network propagation framework based on node core, for genotype-phenotype associations and module identification. NetCore explicitly addresses the node degree bias by incorporating node core in the random walk with restart formulation of network propagation. Additionally, NetCore applies a semi-supervised module identification procedure that allows us to connect between well characterized genes and novel candidate genes, which are significantly scored at the end of the propagation.

We evaluate the performance of NetCore using gene sets from 11 different traits, which are based on previously established genome-wide associations. We show, using a cross-validation scheme, that our core-based approach improves the performance in comparison to the standard degree-based approach. Furthermore, we determine that our semi-supervised module identification procedure allows us to enhance the connectivity between the known phenotype-associated genes by introducing connections to novel candidate genes. The performance is assessed with respect to the choice of the different parameters in NetCore, along with assorted versions of the protein-protein interaction network, which was extracted from ConsensusPathDB.

We demonstrate the application of NetCore to identify disease genes and modules for schizophrenia genome wide mutation data as well as for pan-cancer mutation data. We compare the results with existing network propagation methods and highlight the benefits of using NetCore in comparison to those. To illustrate the versatility of NetCore, we also apply it to gene expression levels measured upon anthracycline drug treatments, in order to elucidate the mechanisms of drug-induced toxicity.

Altogether, this thesis provides a novel framework, with an easy-to-use implementation, which can be applied to various types of genomics data in order to obtain a re-ranking of genes and functionally relevant network modules. Our contributions improve the re-ranking after propagation, augment the experimental evidence towards candidate genes, and produce modules which connect well-characterized genes with novel predictions.

Zusammenfassung

Network Propagation, also die Analyse der Informationsausbreitung in Netzwerken, hat sich in den letzten Jahren als nützliches Konzept für die medizinisch-biologische Forschung erwiesen, insbesondere bei der Analyse von Genotyp-Phänotyp Assoziationen (GPA). Existierende Methoden basieren dabei auf dem Knotengrad bei der Berechnung der Lösungen der mathematischen Prozesse (random walk with restart (RWR)). Der Knotengrad in biologischen Netzwerken neigt allerdings zu Verzerrungen. Außerdem stellt sich die Frage, wie aus dem errechneten Gleichgewichtszustand Teilnetzwerke bestimmt werden können (Netzwerkmodule), die Knoten mit hohem Gewicht miteinander verbinden und biologische Funktionen abbilden. Dies ist bei bisherigen Verfahren nicht optimal gelöst.

In dieser Arbeit wird ein neues Verfahren zur Network Propagation entwickelt (NetCore) zur Analyse von GPA und zur Identifizierung von Netzwerkmodulen. NetCore basiert dabei im Gegensatz zu existierenden Methoden nicht auf dem Knotengrad als Parameter für die Berechnung des Gleichgewichtszustandes, sondern führt dazu den Core des Knotens (node core) ein. Dieses Maß erweist sich als robust gegenüber technischen oder Annotations-bedingten Verzerrungen in den Interaktionsnetzwerken und ist damit dem Knotengrad überlegen. Das neue Maß wird in den RWR eingebaut, so dass die Konvergenzbedingungen erfüllt sind. Nach dem Erreichen des Gleichgewichtszustandes realisiert NetCore im zweiten Schritt eine semi-überwachte Prozedur zur Identifizierung von Netzwerkmodulen, indem bereits bekannte Gene (Knoten) für den untersuchten Phänotyp als Initialisierung verwendet und mit den signifikant bewerteten Knoten verknüpft werden.

NetCore wurde anhand von 11 verschiedenen Genotyp-Phänotyp Analysen aus genomweiten Assoziationsstudien validiert. Mithilfe von Kreuzvalidierung wird gezeigt, dass der Core-basierende Ansatz (NetCore) zu einer Verbesserung im Vergleich zu Knotengrad-basierenden Ansätzen führt. In der Arbeit wird gezeigt, dass NetCore sehr gut geeignet ist, um krankheitsrelevante Gene und Netzwerkmodule aus verschiedenen Typen von experimentellen Ausgangsdaten zu berechnen. Zum einen wird das Verfahren auf Mutationsdatensätze zu Schizophrenie und Krebs angewendet. Zum anderen wird das Verfahren auf Genexpressionsdaten in einem konkreten Anwendungsfall zur Medikamententoxizität getestet. Hierbei wurden 3D Mikrogewebe menschlicher Herzmuskelzellen mit Anthrazyklinen behandelt, und die Effekte dieser Behandlung mit RNA-seq gemessen. Es wird gezeigt, wie solche Genexpressionsmessungen auf das Netzwerk übertragen werden können, und wie NetCore daraus biologisch-funktionell sinnvolle Netzwerkmodule identifizieren kann.

Die Dissertation trägt zur Robustifizierung und Verbesserung von RWR Verfahren bei und ist ein Werkzeug zur Identifizierung von GPA sowie von Netzwerkmodulen zur funktionellen Beschreibung der zugrundeliegenden biologischen Prozesse.

Curriculum Vitae

For reasons of data protection, the curriculum vitae is not published in the online version.

Declaration

Declaration of authorship

Name: Barel

First name: Gal

I declare to the Freie Universität Berlin that I have completed the submitted dissertation independently and without the use of sources and aids other than those indicated. The present thesis is free of plagiarism. I have marked as such all statements that are taken literally or in content from other writings. This dissertation has not been submitted in the same or similar form in any previous doctoral procedure.

I agree to have my thesis examined by a plagiarism examination software.

Berlin, September 25 2020

Gal Barel